Modeling Users' Curiosity in Recommender Systems

ZHE FU, University of North Carolina at Charlotte, USA
XI NIU, University of North Carolina at Charlotte, USA

Today's recommender systems are criticized for recommending items that are too obvious to arouse users' interests. Therefore the research community has advocated some "beyond accuracy" evaluation metrics such as novelty, diversity, and serendipity with the hope of promoting information discovery and sustaining users' interests over a long period of time. While bringing in new perspectives, most of these evaluation metrics have not considered individual users' differences in their capacity to experience those "beyond accuracy" items. Open-minded users may embrace a wider range of recommendations than conservative users. In this paper, we proposed to use curiosity traits to capture such individual users' differences. We developed a model to approximate an individual's curiosity distribution over different stimulus levels. We used an item's surprise level to estimate the stimulus level and whether such a level is in the range of the user's appetite for stimulus, called *Comfort Zone*. We then proposed a recommender system framework that considers both user preference and their *Comfort Zone* where the curiosity is maximally aroused. Our framework differs from a typical recommender system in that it leverages human's *Comfort Zone* for stimuli to promote engagement with the system. A series of evaluation experiments have been conducted to show that our framework is able to rank higher the items with not only high ratings but also high curiosity stimulation. The recommendation list generated by our algorithm has higher potential of inspiring user curiosity compared to the state-of-the-art deep learning approaches. The personalization factor for assessing the surprise stimulus levels further helps the recommender model achieve smaller (better) inter-user similarity.

CCS Concepts: • **Information systems** → **Recommender systems**.

Additional Key Words and Phrases: Recommender Systems, Curiosity, Surprise, Deep Learning

1 INTRODUCTION

Today's recommender systems have been criticized for having the problem of "information filter bubble" [49] or "echo chamber" [2] by offering people close matches with what they have seen already, but not exposing them to a broader range of information. To burst the information bubble and break the echo chamber, the research community has called for some "beyond accuracy" objectives such as novelty, diversity, unexpectedness, and serendipity. Among these "beyond accuracy" objectives, we propose to add one: curiosity, meaning a response to a stimulus [4]. Curiosity represents a strong desire to know or learn something, and is central in human information seeking [31]. We believe it should be an important objective for recommender systems to promote users' engagement to continue using the system. However, curiosity receives little attention in recommender systems research.

In this paper, we regard each recommended item as a stimulus to a user. We define curiosity as a user's response likelihood to a recommended item. Therefore, the curiosity distribution for that user is his/her response probability distribution estimated over his/her historical items. We built a curiosity distribution curve for each

Authors' addresses: Zhe Fu, University of North Carolina at Charlotte, 9201 University City Blvd., Charlotte, North Carolina, USA, 28223-0001, zfu2@uncc.edu; Xi Niu, University of North Carolina at Charlotte, Charlotte, USA, xniu2@uncc.edu.

1556-4681/2023/8-ART $15.00
https://doi.org/10.1145/3617598

user. The curve was then incorporated into a state-of-the-art deep learning recommendation model to re-rank or re-prioritize items that were highly likely to stimulate the user's curiosity.

The curiosity distribution curve was inspired by the early German psychologist Wilhelm Wundt, who proposed the Wundt curve in the 1800s [66] that describes the relationship between the amount of stimulus and the pleasant feeling. According to the Wundt curve, as in Figure 1, too little stimulus will not be exciting whereas too much will cause anxiety. This creates a stimulus "sweet spot" where the pleasant feeling is near its peak. This "sweet spot" is highly personal. In this study, we proposed an approach to quantify it as a *Comfort Zone*.

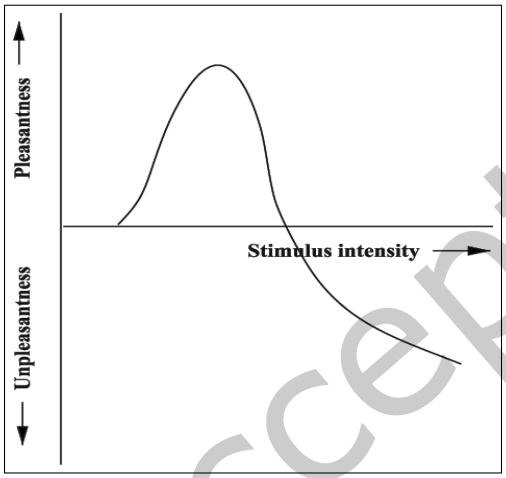

Fig. 1. Illustration of the Wundt curve [66]

Specifically, we developed a computational approach to quantify the stimulus level of each recommendation to a user, and the desired stimulus range as the *Comfort Zone*. Then we used an item's stimulus level distance to the *Comfort Zone* as a criterion to re-rank the items predicted by a deep learning recommender model. The re-ranking algorithm promotes the items that have sufficient stimulus amount to be exciting but not too much to be intimidating. The evaluation experiments have demonstrated that our recommender framework has a balanced consideration between recommendation accuracy and user response likelihood.

This study's contributions are four-fold: 1) we are the first to quantify the region of the *Comfort Zone* to reflect the original idea of the Wundt curve in 1874 [66]; 2) we proposed to use *surprise* as the stimulus, and developed an approach to quantify a personalized surprise level of an item to a user; 3) we innovatively incorporated the curiosity component into five deep learning recommender models using a re-ranking approach; and 4) we proposed a new evaluation metric: *Discounted Cumulative Curiousness (DCC)*, to better evaluate our new recommender framework's ranking quality in terms of its potential to inspire curiosity.

We conducted extensive experiments on three widely-used real world datasets for recommender systems: the Amazon Books [36], the Yelp Restaurants [20], and the Million Song Dataset [5]. All these datasets are information-rich because of their large number of items and abundant users' rating history. Also, book reading behavior, restaurant selections, and music preferences of users are highly driven by personal curiosity and personal taste.

2 RELATED WORK

This research brings together three lines of relevant research: beyond-accuracy recommender systems, modeling curiosity in intelligent systems, and modeling surprise in artificial intelligence (AI).

2.1 Beyond-Accuracy Recommender Systems

With the wide application of deep learning methods, tremendous success has been achieved in recommender systems to recommend "accurate" matches to users' preferences (e.g., [13, 25, 26]). However, these recommender systems overly focus on users' past preferences and excessively pursue the recommendation accuracy. They tend to ignore the users' ever-changing, novel, diverse, and serendipitous needs. Therefore, in recent years, researchers in recommender systems have advocated "beyond-accuracy" recommendations. Some recent representative works will be introduced.

Novelty is usually defined in two ways in recommender systems. One is newly listed items and the other is the existing unpopular items that people tend to miss. For newly listed items, Mohamed et al. [41] recommended new music products to users with the intent of making the recommendation more attractive. Similarly, Deldjoo et al. [8] recommended new movies which could increase the novelty of the recommendations and help reach the goal of business-centric recommendations. Mazumdar et al. [35] recommended newly added Points of Interests (POIs) to users in travel recommendations. For the existing unpopular items, Zhang et al. [67] believed there was an embedded tendency in humans to explore novelty which is extremely outstanding in dining behavior. They recommended some rarely visited restaurants that were likely to be missed.

Diversity as an objective has been introduced in recommender systems to promote different and diverse kinds of items for users. Some researchers regarded diversity as the dissimilarity between recommended items and users' historical records. For example, Cui et al. [7] optimized the recommendation diversity by maximizing the sum of the pairwise correlations and the sum of pairwise dissimilarity between recommended items and the users' historical items. Speaking of dissimilarity, there are many well-established metrics such as cosine similarity, Euclidean distance, etc. There are also some new works on similarity measurement, such as reachable distance function for KNN classification [70] and KNN classification with one-step computation [69]. Other than dissimilarity, Li et al. [29] defined diversity as both user side and item side elasticity. User elasticity is the ability of a user to accept items different from their past behaviors. Item elasticity is the item's characteristics that could be liked by many people. They considered diversity using both a user specific and item specific way. Sun et al. [58] regarded diversity as the added uncertainty on top of a user's regular visit. They intentionally added noises to the user-item interaction graph and generated diversified item representations.

Serendipity is gaining much attention these years as one of the "beyond accuracy" objectives. Two recent comprehensive survey papers on serendipity recommendations are [1, 11]. Although currently there is no consensus on the definition of serendipity, most of the researchers believe the core element of serendipity is unexpectedness. Li et al. [27] defined unexpectedness as distance to a user's typical visit. They incorporated unexpectedness into their deep learning recommendation model to provide serendipitous recommendations to users. Zhang et al. [68] calculated unexpectedness from the aspects of both category difference and the latent representation difference between candidate items and users' profiles. Some other researchers calculated the unexpectedness through co-occurrence. For example, Niu et al. [43, 45] proposed a method to model the users' expectation on news as the expected likelihood of a particular bag of co-occurring topics. A lower likelihood compared to the expected likelihood is regarded as unexpectedness. It is noteworthy that some researchers describe unexpectedness using the term 'surprise'. We believe the two terms are equivalent and interchangeable. In addition to unexpectedness, serendipity may also have overlaps with novelty and diversity. To provide serendipitous items, extensive research has been conducted by integrating these objectives into various recommendation algorithms,

especially deep learning models in recent years. We refer the readers to two recent survey papers on serendipity recommendations: Ziarani et al. [73] and Kotkov et al. [24].

Although these "beyond accuracy" objectives may have overlaps in definitions and computations, they are widely believed to contribute to providing a richer set of information resources for the users. For these "beyond accuracy" objectives, we believe one is missing: curiosity, which is believed to play an essential role in the information seeking process. However, curiosity has been under-studied compared to other "beyond accuracy" goals.

2.2 Modeling Curiosity in Intelligent Systems

In the classic study by Berlyne in Psychology in 1966[4], curiosity has been described as both a trait and a state. The trait of curiosity refers to a personality possessed by different individuals as different desire levels to learn new things, while the state of curiosity means a status that the person is in that drives him/her to respond to a stimulus. The latter definition, the state definition, has been modeled in many studies in computational systems and artificial intelligence such as studies [38, 42, 46, 51]. In this study, however, we modeled both of them in our probabilistic curiosity curve where the entire curve represents a trait and a particular point in the curve represents a state.

In the field of Artificial Intelligence (AI) and Robotics, various computational models have been developed to simulate and stimulate curiosity. According to Wu et al. [65], most of these computational approaches model the curiosity arousal process as a two-step process: identify one or several stimulus variables and appraise the stimulus level; then based on the stimulus level, evaluate the curiosity level. In the *the first step*, some models used a single variable to determine the stimulation value. For example, Saunders and Gero [51] developed a computational model of curiosity for intelligent design agents, focusing on the appraisal of novelty, as in the PCM study [71]. Novelty is common for evaluating the curiosity arousal. Other models combined several stimulus variables to determine the stimulus level. For example, in Wu et al. [65], they used curiosity in a virtual companion to detect potentially interesting learning objects for users and help them avoid the feeling of being lost. They considered four stimulus variables: novelty, uncertainty, conflict, and complexity, and proposed a measure for each of them. For example, they measured uncertainty by counting the number of uncertain elements; they defined conflict as the incompatibility degree between the human learner's understanding and the expert knowledge embedded in the virtual world; and they interpreted complexity as how difficult a topic is to a student. Other studies like Macedo and Cardoso [32, 34] proposed a model of curiosity for intelligent agents to simulate human-like exploratory behavior. They introduced novelty, surprise, and uncertainty into their computational model of curiosity.

As *the second step* in the two-step process of modeling curiosity, the level of curiosity is evaluated through a mapping from the stimulus value to the curiosity value. Some models assumed a linear relationship between the stimulus level and curiosity such as [65]. Other models simply used the stimulus value as the curiosity value such as [33, 48, 53]. Still other models followed the principle of "sweet spot" by explicitly simulating the Wundt curve, which represents a nonlinear mapping from stimuli to curiosity such as models in [37, 51]. These models avoided too small and too big stimuli in their stimulus selection approaches.

These previous studies have marked milestones for developing curiosity models in intelligent systems. They have inspired our motivation to find different stimulus factors other than novelty in recommender systems; they also inspired us to use the non-linear mapping from stimuli to curiosity to better approximate the Wundt curve.

2.3 Computational Models of Surprise in Artificial Intelligence

Surprise, as a potential stimulus variable, has received substantial attention in AI research these years. Studies of computational creativity find that unexpected discovery leads to reflective thinking of the current problem,

which in turn leads to further unexpected discoveries [59]. According to Grace et al. [14], this reflective behavior suggests that surprise is one possible trigger for curiosity.

There are three interpretations for surprise in the literature of computational curiosity. The first one interprets surprise as the difference between an expectation and the real outcome. The well-known Bayesian surprise model proposed by Itti and Baldi [21] belongs to this type. Prediction error also matches well with this type and has been utilized in many curiosity models to measure the level of surprise, such as the studies in [3, 52, 54, 60]. The second interpretation describes surprise as the change of knowledge. Storck et al. [56] modeled this type of surprise using the information gain before and after an observation. The third one is using improbability of existence of an item or an event, as proposed by Macedo and Cardoso [33]. Using improbability as surprise, a series of studies by Grace and Maher [15–17] have developed a personalized curiosity engine called PQE that recommends surprising and interesting recipes to users to encourage their curiosity and help diversify their diet. Their surprise model was based on how unlikely the ingredients co-exist in a recipe. A series of studies led by Niu [9, 12, 44, 45, 47] adopted several Information Theory metrics such as entropy and mutual information to calculate how surprising a news article is to its reader.

These previous studies informed this study of the basic idea of using low likelihood or rare occurrence to measure surprise. Built on but different from these existing studies, this study further takes into consideration a person's previous experience into surprise calculation because the same item is believed to carry different amounts of surprise to different users.

3 THE FRAMEWORK ARCHITECTURE

Our proposed recommender framework consists of three main components, as shown in Figure 2. The Preference Model, the Curiosity Model, and the Recommendation Generator. The Preference Model, like any state-of-the-art recommendation models, captures the users' interests to recommend preferred items. The Curiosity Model constructs the curiosity distribution curve for each user. The Recommendation Generator uses the knowledge from both the Preference Model and the Curiosity Model re-ranks items based on a balance between preference and curiosity, and recommends to the user. We will introduce each component in the following subsections. The important mathematical notations used in our proposed framework are listed in Table 1.

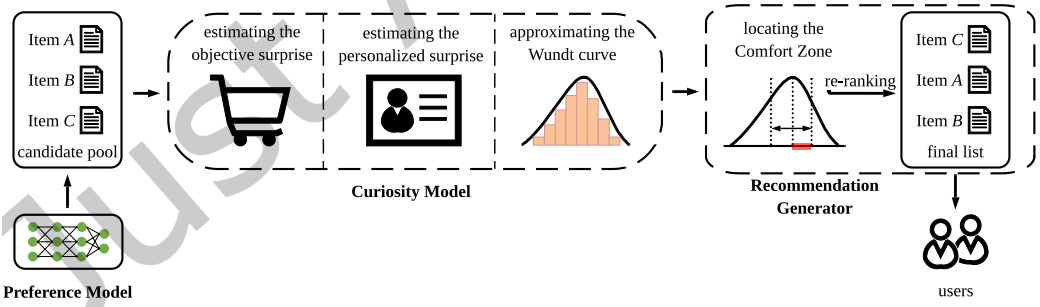

Fig. 2. The framework of the proposed recommender system

3.1 Preference Model

The Preference Model makes use of a user's previous ratings as the user profile and then adopts the state-of-the-art deep learning recommender techniques to identify a set of items that are most preferable to the user. Deep learning techniques typically have higher recommendation accuracy compared to the other techniques due to their strong ability in representation learning and matching function learning from data. Having this Preference

Table 1. Mathematical Notations

Symbol	Description
i	an item in the recommendation list
S	an item-level surprise score
$P^t_{u,i}$	a personalized surprise score of the item i for the user u at the moment t
$F^t_{u,i}$	the frequency (count of times) of the user u has accessed the items related to the item i before the moment t
$SI^t_{u,i}$	the stimulus intensity of the item i for the user u at the moment t
LB_u	the lower bound of the user u's *Comfort Zone*
UB_u	the upper bound of the user u's *Comfort Zone*
λ	a forgetting rate

Model as a separate component enables us to experiment with different off-the-shelf deep learning algorithms without affecting other components of the framework. This facilitates the later experimentation, evaluation, and rapid deployment.

3.2 Curiosity Model

The Curiosity Model, a core component of the proposed recommender framework, uses a user's access history to construct the user's curiosity distribution curve.

3.2.1 Preliminaries. A model that serves as the preliminaries to this study is the Probabilistic Curiosity Model (PCM) developed by Zhao et al. [71]. In PCM, each recommended item presents a stimulus to the user. PCM adopts a probabilistic approach to quantify the Wundt curve. It views a user's selected or responded stimulus level or stimulus intensity (SI) as a random variable, and curiosity as the probability distribution of the random variable. In this way, a user's stimulus selection (response) process can be interpreted as drawing a sample (stimulus) from her/his curiosity distribution. Figure 3 illustrates a user's selection of stimulus under the guidance of the user's curiosity distribution. For this user, SIs around the values near 0.6 is the level at which the curiosity will be maximally aroused and therefore will be selected with a maximal probability. The user may also select other SIs, but the chance is smaller. Here PCM takes both an internal and external view of curiosity. The internal view considers curiosity as an internal factor of the person guiding the selection of items, whereas the external view is the distribution of response probability that could be observed.

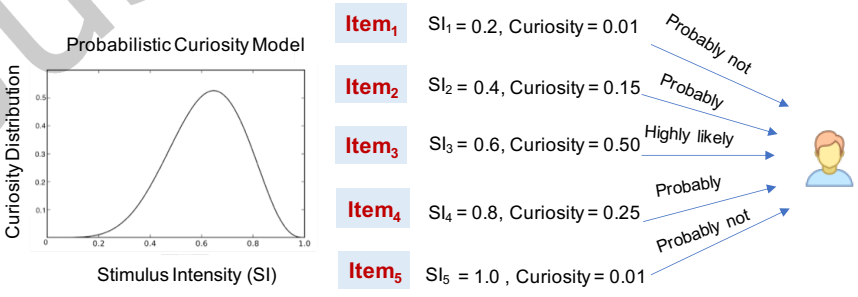

Fig. 3. Illustration of the Probabilistic Curiosity Model (PCM) by [71]

Our study fundamentally advances the Probabilistic Curiosity Model (PCM) developed by Zhao et al. [71], due to our deeper understanding the Wundt curve [66], which leads to a non-linear range of desired item stimulus, and a different optimization function in a recommender system. The PCM study [71] calls the stimulus level where the maximum curiosity is aroused as Anxiety Turning Point (ATP), as in the left figure of Figure 4. It then defines a restriction function to make sure the recommended items' stimulus levels do not exceed this Anxiety Turning Point. We believe that is an oversimplified understanding of the relationship between stimuli and curiosity. We instead propose there is a *Comfort Zone* in stimulus intensity where the user curiosity levels are all relatively high. We believe this *Comfort Zone* better reflects the original idea of "sweet spot" in the Wundt curve [66]. We desire any stimulus level that not only falls within this *Comfort Zone* but also in the upper range, as demonstrated in the right figure in Figure 4. Developmental psychology has long supported the notion of the acquisition of new knowledge being dependent on past exposure to be sufficiently similar knowledge. Vygotsky conceived of the "Zone of Proximal Development": the region of adjacent knowledge beyond but approachable by the learner given their current knowledge [62]. We believe the seeking of stimuli and experiences in this upper range of *Comfort Zone* is known as curiosity, and as curiosity leads to new knowledge, the *Comfort Zone* expands ever outward, leading to renewed curiosity about newly adjacent knowledge. This iterative development cycle is the grounding of our hypothesis for desired stimulus range rather than an Anxiety Turning Point in PCM developed by Zhao et al. [71].

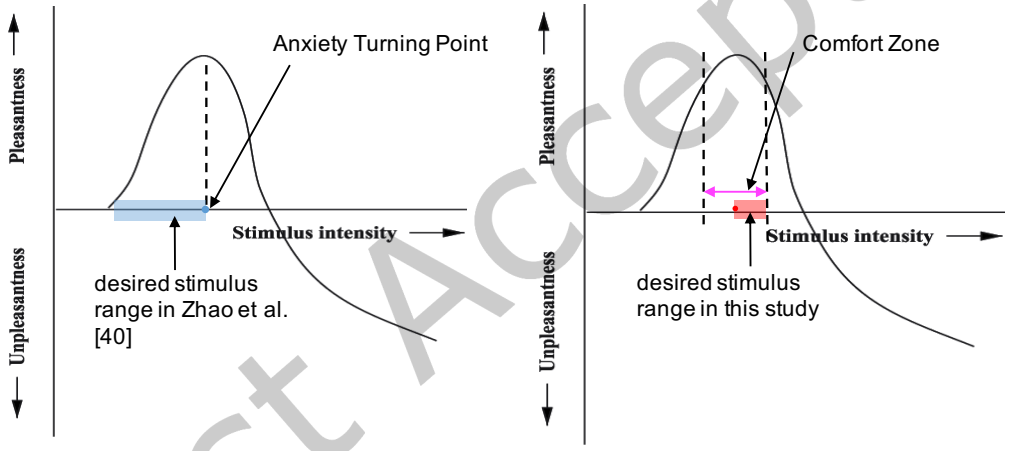

Fig. 4. Comparison: desired stimulus range in Zhao et al. [71] (left figure) and this study (right figure)

In addition, the PCM study [71] used novelty as stimulus for curiosity. We instead propose to use surprise, which, compared to novelty, is more complex and richer in meaning. Surprise is believed to have elements of novelty [50]. In addition, surprise also represents how strongly a stimulus violates an expectation [45]. Therefore, it is more widely believed to be a trigger of curiosity [21]. Since PCM study [71] was published six years ago, it used a traditional collaborative filtering approach as a base. With the recent huge success of deep learning techniques, we would like to experiment our ideas on curiosity models on top of the state-of-the-art deep learning models to investigate the trade-offs between the model accuracy and the curiosity-inspiring potential.

3.2.2 Quantifying the stimulus intensity (SI): computational measure of personalized surprise. The stimulus intensity (SI) could be defined by a number of factors that are extracted from some measurable properties of a stimulus. We propose to use the level of surprise as the curiosity stimulus, since surprise captures all the elements of stimulus factors identified by Berlyne [4], such as novelty [50], conflict with expectation [16], hard to

explain [10], etc. Surprise has been widely believed as a stimulus to the desire to know or learn, which is defined as curiosity [21].

To quantify surprise, this study proposes a two-step calculation: 1) built an objective surprise measure based on the society's collective knowledge; and 2) a personalization factor was incorporated to discount the objective surprise to reflect the personalized level of surprise. The first step was adopted from our previous study [45] with the proven validity. The second step is the new contribution in this paper due to our renewed understanding that surprise is subjective and related to the expectations of the observer. The same observation may carry different amount of surprise for different observers, or even for the same observer at different times. Therefore, we will add a personalization factor on top of the result of the first step.

The first step's approach in our previous study [45] will be briefly introduced here. This approach followed the definition of surprise as violation of expectation [40]. A low likelihood of the expectation would be a surprise to the user. Each item was represented as a "bag" of its elements, represented as a set $E = \{e_1, e_2, ...e_l\}, l \in L$, where L is the size of the "bag". For example, the book "The Prophet" could be represented as a bag of its topics: humanities, religion, and love poems. It then measured objective surprise as how unlikely these topics co-occur in one book. The topic religion tends to co-occur with humanities with a high likelihood, but not as much co-occurring with love poems. Expectations of co-occurrence likelihood have been implicitly formed by the society's collective knowledge, and were computationally constructed using a large collection of such items or some external knowledge base. A surprise in that sense is: "Seeing the topic religion is surprising given seeing the topic love poems." Pointwise Mutual Information (PMI) [6] was used to calculate how much more likely than expected it is that an element e_i occurs given the occurrence of another element e_j. We call this pairwise surprise score s, as in Equation 1:

$$s(e_i, e_j) = -PMI(e_i, e_j) = -log_2 \frac{p(e_i, e_j)}{p(e_i)p(e_j)} \tag{1}$$

where $p(e_i)$ and $p(e_j)$ represent the individual occurrence probabilities of the elements e_i and e_j, and $p(e_i, e_j)$ represents the joint occurrence probability of the two. In this equation, the denominator of the log fraction represents the expectation of these two elements in the collection, and the numerator represents the actual or observed likelihood for this particular combination. The ratio between the two is the divergence between the two, and therefore reflects the amount of surprise.

Since many items have more than two elements, the pairwise surprise s will be calculated for all possible pairwise combinations, and the highest of those values becomes the overall surprise score, S. This is shown in Equation 2, where E is the set of all possible pairwise combinations belonging to the item. This approach adopted the highest instead of the average, based on the idea of Grace et al. [15] that the peak element-level surprise dominates the item-level surprise.

$$S = max_E s(e_i, e_j) \tag{2}$$

The second step is our contribution in this study by incorporating a personalization factor to the surprise calculation. Guided again by the study of Berlyne [4] where the stimulus intensity is believed to be influenced by how often the stimulus has been experienced by a user. The idea is that the more frequently the user has accessed the item or similar items, the less surprising the item will be to the user. To mimic the impact of past access frequency on the current feeling, we used an exponential decay function $e^{-\lambda t}$, commonly used to describe a natural decreasing process at a rate proportional to its current value and with an exponential forgetting rate [28]. Therefore, the personalization factor is represented as in Equation 3:

$$P^t_{u,i} = e^{-\lambda F^t_{u,i}} \tag{3}$$

where λ is the forgetting rate and $F^i_{u,t}$ is the frequency that the user u has experienced the items related to the item i before time t. Note that $F^t_{u,i}$ is a variable that is user-dependent, item-dependent, and also time-dependent. Therefore $SI^t_{u,i}$, the stimulus intensity of the item i for user u at the moment t, is the multiplication of the personalization factor and the objective surprise of the item, represented as Equation 4:

$$SI^t_{u,i} = P^t_{u,i} S \tag{4}$$

Although a simplified personalization model that may not capture all the factors impacting the personal feeling of surprise, this approach reasonably makes use of a user's past access frequency to approximate a person's familiarity level with an area, the most important element in forming an expectation [15]. Surprise just reflects how strongly an encounter violates such an expectation.

3.2.3 Approximating the Wundt curve. Since we view a stimulus selection process as drawing samples (stimuli) from a person's curiosity distribution, it is natural to expect that the person's curiosity distribution curve follows the probability density function (PDF) of the $SI^t_{u,i}$ values. Specifically in this study, the empirical (observational) PDF of $SI^t_{u,i}$ is the distribution of a series of $SI^t_{u,i}$ values where $\{i \in I_u, t \in T_u\}$ in a user's past access history, as shown in the histogram for a hypothetical user in Figure 5. I_u is the set of items visited by the user u and T_u is the set of timestamps of those visited items.

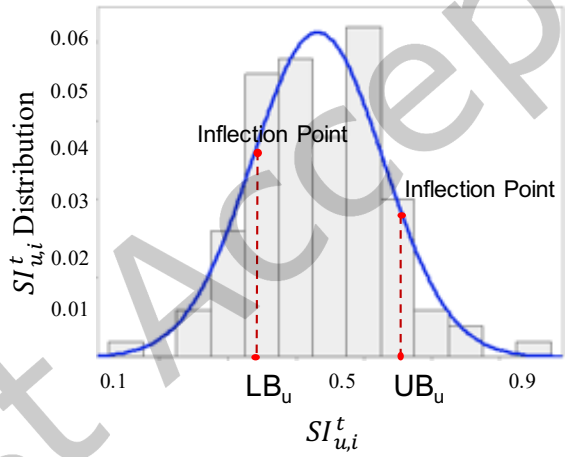

Fig. 5. Empirical PDF (the histogram) and fitted PDF (the curve)

In order to get a continuous PDF from the observational PDF histogram, we used the β distribution to fit a curve for the empirical PDF. β distribution has been applied to modeling random variables of human behavior limited to intervals of finite length in a wide variety of disciplines. It is a family of curves controlled by the parameters α and β to approximate any probability distribution. The fitted curve, as shown in the curve in Figure 5, serves as our Curiosity Model as in Figure 2, and also the approximation of the Wundt Curve because: first, the distribution generally follows the "inverted-U" shape, suggesting that probability density captures the degree of pleasantness implied by the Wundt curve. More importantly, second, we are able to find the two inflection points on the curve and their corresponding stimulus levels, which can be used as the bounds of the *Comfort Zone*, as illustrated in Figure 5. The lower bound and the upper bound are called LB and UB respectively. Third, the fitted distribution quantifies the Wundt Curve using a probabilistic view, which reflects the natural process that humans tend to select the pleasant stimuli more frequently. A person may also respond to a less pleasant

stimulus, but the chance is smaller. Overall speaking, a curious user's *Comfort Zone* is more rightward compared to that of a conservative user.

3.3 Recommendation Generator

We model the recommendation problem as a top-K item ranking problem which selects the top-K items to recommend considering both user preference and curiosity arousal potential. Specifically, the Recommendation Generator obtains top N ($N > K$) items from the Preference Model as a candidate pool. It then re-ranks the N items and gets the top-K items. The re-ranking favors items in the desired stimulus range as in the right figure of Figure 4. Within that range, the closer the item is to the upper bound (UB), the higher it gets ranked. This re-ranking approach reflects the idea that a stimulus that is in the user's *Comfort Zone* but a little beyond his/her typical reach will have a higher likelihood of stimulating curiosity. This way, the Recommender Generator considers both recommendation accuracy (represented by the Preference Model) and the potential to arouse the user's curiosity (represented by the Curiosity Model).

Figure 6 shows the curiosity distributions of two users A and B to represent how the same item has different surprise amounts to these two users, and also different potentials in inspiring curiosity for the two users. User B is more curious than User A according to their curiosity distributions and $UB_{u_B} > UB_{u_A}$. The same item i has different amounts of surprise for the two users, as it appears in different locations in their curiosity distribution curves. Let us suppose both users would give a high rating on this item i. However, based on its distances to UB_{u_A} and UB_{u_B}, this item i may make the final recommendation list for User B due to its proximity to UB_{u_B} but probably not for User A.

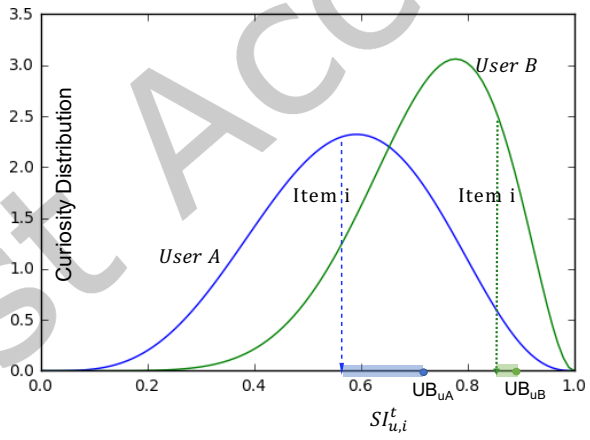

Fig. 6. Illustration of two users' curiosity distributions

4 IMPLEMENTATION OF THE RECOMMENDER FRAMEWORK

In this section, we first described the datasets and then some implementation details for the three components of our proposed recommender framework.

4.1 Datasets

We used 3 widely-used datasets in the recommender research community: Amazon Books [1], Yelp Restaurants [2], and Million Song Dataset [3] to implement our recommender framework. Among them, the Amazon Books dataset is a subset of the original Amazon books dataset [36]. To supplement the original dataset with the book topic labels, we utilized Amazon Product Advertising API to crawl the main topic labels for each book from the Amazon website. The Yelp Restaurants dataset is also a subset of the Yelp business dataset [20] with only restaurant-related records including the ratings and labels. The Million Song Dataset [5] is the original version that is publicly available. All these 3 datasets were pre-processed. In addition, in order to avoid the data sparsity problem, we have removed users with fewer than 10 ratings. The basic statistics of the three datasets used in this study is summarized in Table 2.

Table 2. Statistics of the three datasets used in this study

	Amazon Books	Yelp Restaurants	Million Song Dataset
No. of users	127,627	435,543	1,019,318
No. of items	494,108	39,964	384,546
No. of ratings	3,668,757	1,745,605	48,373,568

Each record of the datasets was split into a training set M (80%) and a test set T (20%). The training set is used to train the deep learning based recommendation algorithms and more importantly calculate the personalized surprise factor for each item. The test set is used to evaluate the performance of our proposed curiosity-based re-ranking approach.

4.2 Preference Calculation: User Rating Prediction

As mentioned in the Preference Model, we used the state-of-the-art deep learning techniques to identify items that are preferred by the user. These state-of-the-art algorithms will also serve as baseline algorithms in our evaluation studies in Section 5. Specifically, we used five recommender algorithms: 1) Variational Autoencoder with Multinomial Likelihood (Multi-VAE) [30], which is a deep learning model that extends the variational autoencoders; 2) Adversarial Personalized Ranking using Matrix Factorization (APR-MF) [18], an algorithm that is developed to maximize the likelihood that the user prefers one item over others; 3) a Personalized Transformer (SSE-PT) model [64], which is inspired by the Transformers [61] in natural languages processing and introduces personalization into self-attentive neural network architectures; 4) a Sequential Recommendation with Bidirectional Encoder Representations from Transformers (BERT4Rec) [57], which is the next-item sequential recommendation method based on Transformers [61] and a Cloze objective; and 5) a Diffusion Recommendation Model (DiffRec) [63], which is the most recent generative recommendation model. It learns the representation of the user interactions in a denoising manner based on Diffusion Models [19, 55].

Each base recommendation algorithm (Multi-VAE, APR-MF, SSE-PT, BERT4Rec, and DiffRec) was implemented to identify a set of N items as a candidate pool for further re-ranking. N has been set to be 100 in this study, a reasonably large pool of candidate items to search for curiosity-inspiring items without sacrificing recommendation accuracy too much.

[1] https://cseweb.ucsd.edu// jmcauley/
[2] https://www.yelp.com/dataset
[3] http://millionsongdataset.com/

4.3 Surprise and Curiosity Calculation

We calculated personalized surprise for each item for each user in the training set via Equation 1 through Equation 4, where e_i is a label for an item. A label for a book is its topic, such as historic, politics, biographies, etc. A label for a restaurant is a tag describing the restaurant's type and flavor, such as Mexican, pizza, bakeries, Chinese, etc. A label for a piece of music is its genre, such as classic, pop, rock, etc. In order to measure the personalization factor $P^t_{u,i}$ in Equation 3 for each item and each user at each time point t, we need to calculate $F^t_{u,i}$, the frequency that the user has accessed the items related to the item i before the moment t. The related items in this study were defined as the items that shared a label with the item i and the shared label must be one label of the label pair that has the highest s for the item i. Therefore, $F^t_{u,i}$ was calculated this way:

$$F^t_{u,i} = \frac{F^t_{u,e_1} + F^t_{u,e_2}}{2} \tag{5}$$

where e_1 and e_2 are the label pair with the highest s of the item i as in Equation 2. F^t_{u,e_1} and F^t_{u,e_2} are the number of times that the user u has accessed e_1 and e_2 respectively before time t. Time t is defined as the access moment of the item i, which means for each accessed item i, we have only considered the access history before this item through the timestamps information of the dataset.

As mentioned, the distribution of $SI^t_{u,i}$ values served as the empirical (observational) curiosity distribution. To further turn this empirical distribution to a continuous PDF distribution, we fit the β distribution using Python's *stats.beta* library in the *SciPy* package. The library took observational frequency distribution as the input, and output the beta distribution parameters α, β, and the curve's lower and upper limits. These values were used later to plot the curiosity distribution curve for each user using Python's *matplotlib* plotting package. The LB_u, and UB_u points were also calculated through the parameters α and β.

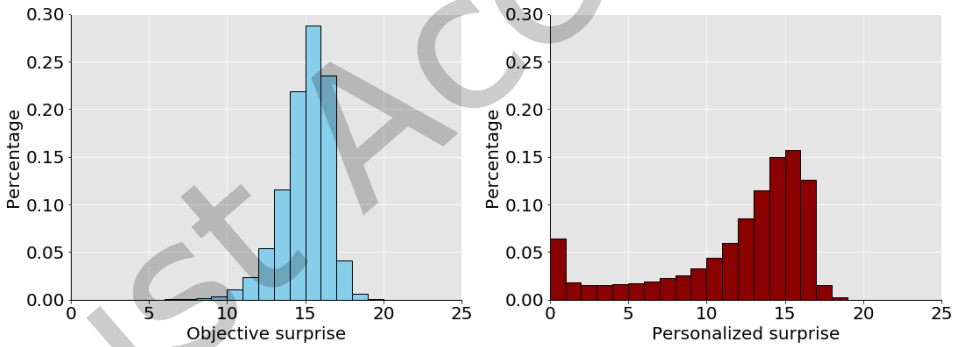

Fig. 7. The distribution of the objective surprise levels (left figure) and personalized surprise levels (right figure)

We will display the result from the Amazon Books as an example. As the result of surprise calculation, the distribution of the objective surprise S as in Equation 2 for all the books in the training dataset is presented in the left figure in Figure 7. The distribution generally follows a normal distribution with the average amount of objective surprise around 14 or 15. The right figure shows the distribution of the personalized surprise $SI^t_{u,i}$, which has lower bar height and spreads out to the lower end. This distribution suggests the personalization effect due to the exponential decay function proposed in Equation 3. More interestingly, this $SI^t_{u,i}$ distribution could serve as an aggregate empirical curiosity distribution for all the users in the training set. Its lower bar height and spreading toward the lower end mean that users' tastes were very different, justifying the benefits of personalization.

As a follow-up illustration, Table 3 shows three examples of surprising books with high amounts of objective surprise but different amounts of personalized surprise, $SI^t_{u,i}$, for two different users. Apparently, User 2 is more difficult to surprise because she has more past access experience with similar books.

Table 3. Examples of surprising books that have different personalized surprise amounts for two users

	Title	Topic Labels	Objective Surprise	Personalized Surprise	
				User 1	User 2
	Uganda Be Kidding Me	Uganda; Humor	19.04	17.22	15.59
	Generation Rx: How Prescription Drugs Are Altering American Lives, Minds, and Bodies	Pharmacology; Politics	17.22	16.38	14.10
	Cooking with a Serial Killer Recipes	Serial Killer; Culinary	16.30	15.50	14.03

To illustrate what empirical curiosity distribution and the fitted curve look like for different users, Figure 8 shows the histograms of the SIs (after normalization) and the fitted curiosity distribution (the blue curves) for five users in our training dataset. This Figure 8 shows that the users tend to respond to different average levels of stimulus. User 1 is relatively more conservative compared with User 4 and User 5. In addition, the variances of the users' distributions are different. User 1 and User 2 have relatively small variances while User 3 to User 5 have relatively large variances. A small variance means that the curiosity level is stable, suggesting that users' curiosity tends not to change much with different levels of stimuli, while a large variance shows that the user's curiosity may vary greatly. Generally, for each curiosity distribution, there is a *Comfort Zone* within which the stimulus has the largest chance to be responded to. The *Comfort Zone* is different for these five users.

4.4 Recommendation Generation

As mentioned in Section 3.3, Recommender Generator obtains a top N items from a base recommender algorithm as a candidate pool, and then re-ranks the N items and gets the top-K items according to the proximity between the item's amount of surprise ($SI^t_{u,i}$) to that user's UB_u, representing as $-dist(SI^t_{u,i}, UB_u)$. The items that make the final list of top-K are also subject to two conditions: $SI^t_{u,i} > \frac{1}{2}(LB_u + UB_u)$ and $SI^t_{u,i} < UB_u$. The experiment algorithms are represented as MultiVAE+Cur, APR-MF+Cur, SSE-PT+Cur, BERT4Rec+Cur, and DiffRec+Cur, meaning a base counterpart plus the curiosity re-ranking.

5 EVALUATION STUDIES

In this section, we proposed and applied three performance metrics to evaluate our recommender framework. We then presented the evaluation results in terms of the three metrics.

5.1 Evaluation Metrics

We proposed three proposed metrics to evaluate our recommender framework:

5.1.1 Recall. This work adapted the *one plus random* evaluation method [23] with some modification. It randomly splits each user's rated items into a training set M and test set T. An additional probe set P is constructed by selecting up to 10 highly rated items (e.g., those having a four- or five-star rating on a 1 to 5 scale) from the user's test set T. Then, for each user u, predictions will be computed to select the top N ($N = 100$ in this study) unrated

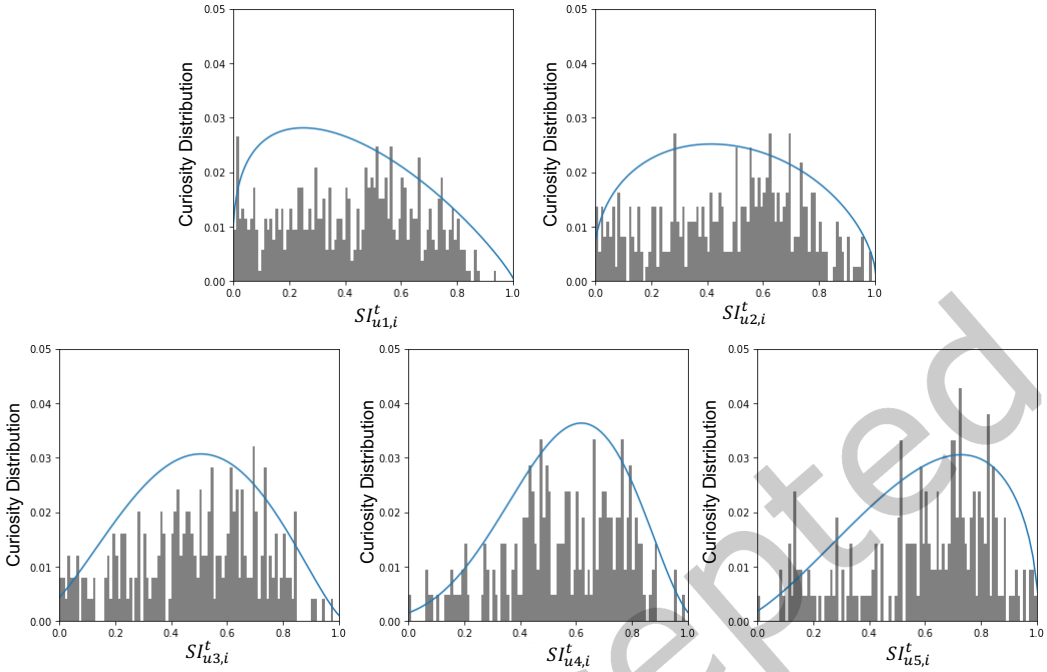

Fig. 8. Examples of five users' curiosity distributions

items as the candidate pool plus all the p items in P. The set of $100 + p$ items is ranked according to a baseline algorithm (MultiVAE, APR-MF,SSE-PT, BERT4Rec, or DiffRec), or an experimental algorithm (MultiVAE+Cur, APR-MF+Cur, SSE-PT+Cur, BERT4Rec+Cur, or DiffRec+Cur). We examined whether the experimental algorithms were able to rank the p items higher among the $100 + p$ items than the baseline algorithms. The underlying rationale is since all the items in P represent both high ratings (preferences) and high response likelihood (curiosity), they should be ranked higher compared to most items in the candidate set N.

Specifically, for each user u, whether the items in P are ranked higher is calculated by Recall@K, which is defined as:

$$Recall_u@K = \frac{\text{number of items in P ranked in top K}}{\text{the total number of items in P}} \tag{6}$$

The overall value of $Recall@K$ is the average of $Recall_u@K$ for all the users. $Recall@K$ is an important metric to evaluate whether a recommender algorithm is able to rank higher curiosity inspiring items with higher response likelihood, while not sacrificing preferences much.

5.1.2 Discounted Cumulative Curiousness (DCC). Inspired by the measure of Discounted Cumulative Gain (DCG) [22] that considers both relevance and ranking position to measure the ranking quality, we propose a measure, called Discounted Cumulative Curiousness (DCC), to measure the ranking quality in terms of curiosity-inspiring potential, represented as:

$$DCC_u@K = \sum_{i=1}^{K} \frac{\text{curiousness score}}{log_2(i+1)} \tag{7}$$

where $DCC_u@K$ is the recommendation result list's DCC for user u at each position i from the first position up to the position K. How to measure curiousness is the key problem for applying this measure. Since the ranking is

generated (predicted) by ordering the candidate items by the *horizontal* distance between $SI^t_{u,i}$ and UB_u, we will evaluate curiousness using the "ground truth" data: the observed frequency offered by the *"vertical"* height of a histogram bar in a user's curiosity distribution, representing the actual response likelihood of an item with that stimulus level. We may use this observed frequency as the curiousness score in Equation 7.

The overall value of DCC@K is the average of $DCC_u@K$ for all the users. The higher the value of $DCC@K$, the more potential the recommender has to arouse users' curiosity.

5.1.3 Inter-User Similarity (IUS). Since our recommender framework quantifies a stimulus in a personalized way, we expect that its recommendations are different for different users. To test this expectation, we use inter-user similarity (IUS) proposed in [72]. The $IUS_{i,j}$ between the user i and j is the proportion of overlap between two recommendation lists L_i and L_j for the user i and j.

$$IUS_{i,j} = \frac{\left|L_i \bigcap L_j\right|}{|K|} \tag{8}$$

The overall value for IUS for all the users is the average of $IUS_{i,j}$ between all pairs of users. A large value of IUS means a high similarity between users and therefore less effect of personalization.

5.2 Evaluation Results

We have conducted two sets of evaluation studies for our recommender framework. The purpose of the first set is to evaluate the effectiveness of incorporating curiosity into the recommender system whereas the second set is to test the effectiveness of personalization of the surprise levels.

5.2.1 Evaluating the curiosity model. In this set of evaluation, we investigated the effect of different values of K on the three metrics we proposed: Recall, DCC, and IUS. We compared two sets of algorithms: MultiVAE, APR-MF, SSE-PT, BERT4Rec, and DiffRec without considering the Curiosity Model, as five baseline algorithms; and MultiVAE+Cur, APR-MF+Cur, SSE-PT+Cur, BERT4Rec+Cur, and DiffRec+Cur as our experimental algorithms.

Table 4, 5, and 6 show the Recall, DCC, and IUS levels at different K values for each algorithm. In terms of Recall, all five experimental algorithms outperformed their baseline counterparts at varying K values for all the three datasets. This confirms our hypothesis that re-ranking the candidate items by the proximity to the upper range of the user's *Comfort Zone* will result in a higher chance of hitting an item with high response likelihood as well as a high rating. The performance is as expected since as K increases the chance of hitting a good item is larger. Among the five experimental algorithms, SSE-PT+Cur and BERT4Rec+Cur have the best performances in the Amazon Books and Yelp Restaurants datasets while Multi-VAE+Cur and DiffRec+Cur are the best in the Million Song Dataset. In terms of DCC, all the five experimental algorithms have higher DCC values than their baseline counterpart algorithms, backing up our hypothesis again that re-ranking with the consideration of a person's Comfort Zone will generate a list of recommendations with higher potential of arousing curiosity.

Furthermore, the IUS values in Table 6 represent the similarity of inter-user recommendations for the ten recommender algorithms. A small value of IUS indicates a large effect of the personalization factor, which is therefore desired. Overall speaking, SSE-PT+Cur, BERT4Rec+Cur, and DiffRec+Cur have outperformed their baseline counterpart for all of the three datasets. APR-MF+Cur occasionally has slightly larger (worse) IUS in the Amazon Books dataset, while Multi-VAE+Cur occasionally has higher (worse) IUS in the Yelp Restaurants and Million Song Dataset. The reason is probably because both APR-MF and Multi-VAE are the state-of-the-art personalized deep learning approaches. Accommodating the curiosity consideration may hurt their strong personalization capability. In addition, in the current Amazon Books, Yelp Restaurants, and Million Songs, there are small sets of popular items which have been accessed by many users. In order to increase the response likelihood, the experimental algorithms tend to recommend some items from this popular set, which slightly

Table 4. Model performance on Recall with varying K

Method		Recall@K (the larger the better)				
		Recall@10	Recall@20	Recall@30	Recall@40	Recall@50
Amazon Books	Multi-VAE	9.33%	16.06%	20.24%	23.80%	26.86%
	Multi-VAE+Cur	**14.70%**	**30.92%**	**43.80%**	**49.81%**	**54.89%**
	APR-MF	5.56%	11.32%	18.53%	25.49%	33.95%
	APR-MF+Cur	**23.16%**	**35.04%**	**45.90%**	**56.54%**	**59.29%**
	SSE-PT	11.88%	30.00%	39.66%	45.71%	46.69%
	SSE-PT+Cur	**33.27%**	**44.02%**	**56.50%**	**57.59%**	**59.70%**
	BERT4Rec	12.26%	38.87%	47.85%	51.99%	53.23%
	BERT4Rec+Cur	**31.09%**	**44.32%**	**54.59%**	**58.01%**	**59.29%**
	DiffRec	10.90%	16.54%	22.57%	27.07%	28.57%
	DiffRec+Cur	**16.09%**	**32.33%**	**45.86%**	**50.75%**	**54.13%**
Yelp Restaurants	Multi-VAE	4.66%	8.14%	11.01%	13.71%	16.21%
	Multi-VAE+Cur	**15.77%**	**28.57%**	**37.73%**	**42.10%**	**44.53%**
	APR-MF	5.00%	33.92%	41.05%	47.01%	48.60%
	APR-MF+Cur	**20.17%**	**40.52%**	**47.72%**	**50.09%**	**50.13%**
	SSE-PT	19.28%	39.03%	48.71%	50.84%	50.84%
	SSE-PT+Cur	**40.85%**	**51.60%**	**51.64%**	**51.65%**	**51.65%**
	BERT4Rec	18.54%	39.28%	47.67%	50.00%	50.06%
	BERT4Rec+Cur	**46.43%**	**50.07%**	**50.16%**	**50.18%**	**50.18%**
	DiffRec	5.30%	8.97%	15.62%	17.83%	22.09%
	DiffRec+Cur	**18.09%**	**28.33%**	**36.99%**	**43.06%**	**45.23%**
Million Song Dataset	Multi-VAE	7.71%	10.90%	13.34%	15.20%	16.70%
	Multi-VAE+Cur	**34.58%**	**41.29%**	**44.43%**	**46.48%**	**46.87%**
	APR-MF	11.33%	20.50%	30.27%	36.10%	36.11%
	APR-MF+Cur	**27.33%**	**30.43%**	**30.46%**	**40.46%**	**46.28%**
	SSE-PT	20.25%	29.38%	30.37%	35.65%	35.83%
	SSE-PT+Cur	**30.72%**	**41.38%**	**44.38%**	**48.86%**	**48.94%**
	BERT4Rec	14.16%	28.37%	36.41%	37.94%	38.08%
	BERT4Rec+Cur	**31.90%**	**39.02%**	**46.38%**	**47.65%**	**47.84%**
	DiffRec	8.67%	17.18%	21.72%	23.67%	23.98%
	DiffRec+Cur	**37.88%**	**41.92%**	**45.51%**	**48.18%**	**48.83%**

hurts IUS. The result reflects the well-known phenomenon of "the rich get richer" [39] in the dataset we used in this study.

Regarding the five experimental algorithms, APR-MF+Cur, SSE-PT+Cur, and BERT4Rec+Cur outperformed MultiVAE+Cur and DiffRec+Cur in terms of IUS on the Amazon Books and Yelp Restaurants datasets, while MultiVAE+Cur, SSE-PT+Cur, and DiffRec+Cur performed better than APR-MP+Cur and BERT4Rec+Cur in the Million Song Dataset.

5.2.2 Evaluating personalized surprise vs. objective surprise. To follow up with the phenomenon of "the rich get richer" in the 3 datasets we used, we want to conduct analysis on the effect of personalization and whether personalization helped mitigate such a problem. In this study, we have calculated the personalized surprise for each user based on Equation 4 with the expectation that the same item may contain different amounts of surprise to different individuals. This second set of evaluation studies is to evaluate whether using personalized surprise to

Table 5. Model performance on DCC with varying K

Method		DCC@K (the larger the better)				
		DCC@10	DCC@20	DCC@30	DCC@40	DCC@50
Amazon Books	Multi-VAE	1.34	2.17	2.72	3.30	3.84
	Multi-VAE+Cur	**4.54**	**6.05**	**6.47**	**6.84**	**7.15**
	APR-MF	1.54	2.93	3.75	4.21	4.39
	APR-MF+Cur	**4.54**	**6.74**	**8.11**	**8.17**	**8.72**
	SSE-PT	2.03	3.30	4.05	4.42	4.53
	SSE-PT+Cur	**3.81**	**5.29**	**5.97**	**6.61**	**6.61**
	BERT4Rec	2.57	4.34	4.87	5.09	5.81
	BERT4Rec+Cur	**3.65**	**5.44**	**6.84**	**7.96**	**8.50**
	DiffRec	1.99	2.77	3.35	3.81	4.38
	DiffRec+Cur	**5.33**	**6.74**	**7.08**	**7.47**	**7.74**
Yelp Restaurants	Multi-VAE	1.13	1.76	2.30	2.80	3.30
	Multi-VAE+Cur	**3.81**	**4.53**	**4.76**	**4.84**	**4.85**
	APR-MF	4.36	4.53	4.55	4.56	4.56
	APR-MF+Cur	**4.54**	**6.63**	**7.83**	**8.40**	**8.60**
	SSE-PT	4.45	6.47	7.64	8.25	8.45
	SSE-PT+Cur	**4.53**	**6.92**	**7.83**	**8.40**	**8.72**
	BERT4Rec	3.54	5.85	6.43	6.67	6.75
	BERT4Rec+Cur	**5.35**	**7.46**	**8.02**	**8.18**	**8.20**
	DiffRec	1.72	2.45	2.93	3.45	3.94
	DiffRec+Cur	**4.50**	**5.32**	**5.62**	**5.80**	**5.92**
Million Song Dataset	Multi-VAE	1.23	1.63	1.96	2.33	2.63
	Multi-VAE+Cur	**3.82**	**4.00**	**4.09**	**4.16**	**4.20**
	APR-MF	4.06	5.12	5.54	5.70	5.75
	APR-MF+Cur	**4.14**	**5.19**	**5.80**	**6.10**	**6.23**
	SSE-PT	4.36	5.11	5.44	5.55	5.60
	SSE-PT+Cur	**4.53**	**5.81**	**5.94**	**6.34**	**6.60**
	BERT4Rec	3.88	4.26	4.51	4.68	4.76
	BERT4Rec+Cur	**4.59**	**5.15**	**5.53**	**5.68**	**5.75**
	DiffRec	1.92	2.37	2.56	2.92	3.23
	DiffRec+Cur	**4.36**	**5.12**	**5.48**	**5.62**	**5.70**

estimate stimulus level brings value in finding curiosity-inspiring items as well as inter-user similarity, compared to if we just use the objective surprise as stimulus level: the same item carries the same amount of stimulus for everyone.

We selected one algorithm, SSE-PT+Cur from the previous round of evaluation because of its relatively better and stable performance in all of the three datasets compared to the other two experimental algorithms. We applied this algorithm into two settings: using objective surprise as $SI'^{t}_{u,i}$ or using personalized surprise as $SI^{t}_{u,i}$, and compared their performance in these two settings. The results are shown in Table 7, 8, and 9. SSE-PT+Obj denotes the SSE-PT algorithm using the objective surprise calculation. In terms of Recall, the personalized approach outperforms the objective approach. This confirmed our hypothesis that personalized surprise better reflects the stimulus intensity specific to a user and therefore results in a higher chance of hitting a curiosity-inspiring and relevant items. While in terms of DCC, the personalized approach has lower DCC values, probably because it diversifies the items, deviating from the popular items by adding a personalization factor. For Table 9, in terms

Table 6. Model performance on IUS with varying K

Method		IUS@10	IUS@20	IUS@30	IUS@40	IUS@50
		(the smaller the better)				
Amazon Books	Multi-VAE	14.44%	17.23%	22.64%	27.44%	32.67%
	Multi-VAE+Cur	**13.08%**	**16.44%**	**19.45%**	**20.40%**	**22.61%**
	APR-MF	**4.61%**	**5.73%**	9.85%	12.20%	13.24%
	APR-MF+Cur	5.80%	6.98%	**8.63%**	**10.09%**	**11.44%**
	SSE-PT	4.94%	6.60%	10.05%	12.72%	12.99%
	SSE-PT+Cur	**3.59%**	**5.62%**	**7.22%**	**8.63%**	**8.77%**
	BERT4Rec	5.03%	6.75%	9.04%	11.70%	13.37%
	BERT4Rec+Cur	**3.69%**	**5.67%**	**6.49%**	**6.89%**	**7.40%**
	DiffRec	8.76%	11.27%	15.46%	19.24%	25.07%
	DiffRec+Cur	**6.95%**	**8.74%**	**10.49%**	**11.89%**	**12.61%**
Yelp Restaurants	Multi-VAE	5.43%	9.04%	**10.61%**	**12.38%**	18.84%
	Multi-VAE+Cur	**5.27%**	**8.65%**	11.64%	13.85%	**15.14%**
	APR-MF	2.47%	14.27%	22.76%	28.18%	33.12%
	APR-MF+Cur	**1.33%**	**9.19%**	**12.67%**	**16.22%**	**24.17%**
	SSE-PT	3.12%	5.89%	11.12%	12.72%	13.94%
	SSE-PT+Cur	**1.93%**	**4.72%**	**7.02%**	**8.56%**	**9.75%**
	BERT4Rec	3.11%	4.30%	6.36%	9.40%	11.61%
	BERT4Rec+Cur	**1.92%**	**2.80%**	**4.96%**	**6.41%**	**7.73%**
	DiffRec	5.67%	9.18%	12.38%	15.68%	17.68%
	DiffRec+Cur	**4.67%**	**7.58%**	**9.44%**	**10.74%**	**11.78%**
Million Song Dataset	Multi-VAE	**2.67%**	**2.99%**	5.70%	7.55%	9.25%
	Multi-VAE+Cur	3.29%	3.49%	**5.51%**	**6.26%**	**9.18%**
	APR-MF	5.80%	7.26%	9.31%	11.79%	17.05%
	APR-MF+Cur	**4.61%**	**5.75%**	**7.34%**	**9.26%**	**16.01%**
	SSE-PT	3.62%	4.71%	5.89%	7.53%	9.52%
	SSE-PT+Cur	**1.61%**	**2.91%**	**3.31%**	**4.44%**	**4.45%**
	BERT4Rec	5.95%	7.52%	8.89%	9.37%	10.98%
	BERT4Rec+Cur	**3.32%**	**4.88%**	**5.92%**	**6.23%**	**7.19%**
	DiffRec	3.43%	4.32%	6.26%	8.34%	9.99%
	DiffRec+Cur	**3.33%**	**3.88%**	**5.92%**	**6.21%**	**8.89%**

of IUS, the personalized approach has constantly achieved a smaller IUS for different values of K, suggesting the effectiveness of personalization. This observation supports our belief that using personalized surprise has alleviated the problem of convergence to some popular items.

6 CONCLUSION AND FUTURE WORK

This paper presents a recommender framework that considers both user preference and curiosity arousal potential. The Curiosity Model is constructed for each individual user to model their unique appetite for stimuli. To quantify a stimulus, we proposed to use surprise as the stimulus factor and developed a measure for estimating personalized amount of surprise an item contains. Moreover, we have quantified the *Comfort Zone* concept in the Wundt Curve by finding its lower and upper bounds from the fitted curve. We measured the distance between an item's stimulus level to a user's upper bound of the *Comfort Zone* with the aim of providing a little beyond but approachable recommendations to inspire user curiosity. Three popular datasets about books, restaurants, and music, representing a wide range of personal tastes, have been adopted to illustrate our idea. In the evaluation studies, we have shown that our algorithms are able to rank higher those items with not only high ratings but also

Table 7. Evaluating personalization on Recall with varying K

Method		Recall@K (the larger the better)				
		Recall@10	Recall@20	Recall@30	Recall@40	Recall@50
Amazon	SSE-PT+Obj	30.15%	44.02%	49.94%	50.64%	50.65%
Books	SSE-PT+Cur	**33.27%**	**44.39%**	**56.50%**	**57.59%**	**59.70%**
Yelp	SSE-PT+Obj	11.15%	39.07%	48.95%	49.82%	49.98%
Restaurants	SSE-PT+Cur	**40.85%**	**51.60%**	**51.64%**	**51.65%**	**51.65%**
Million Song	SSE-PT+Obj	22.76%	32.24%	38.55%	39.74%	39.74%
Dataset	SSE-PT+Cur	**30.72%**	**41.38%**	**44.38%**	**48.86%**	**48.94%**

Table 8. Evaluating personalization on DCC with varying K

Method		DCC@K (the larger the better)				
		DCC@10	DCC@20	DCC@30	DCC@40	DCC@50
Amazon	SSE-PT+Obj	**4.54**	**6.71**	**8.03**	**8.18**	**8.79**
Books	SSE-PT+Cur	3.81	5.29	5.97	6.61	6.61
Yelp	SSE-PT+Obj	**4.53**	**7.01**	**9.06**	**9.57**	**9.59**
Restaurants	SSE-PT+Cur	4.53	6.92	7.83	8.40	8.72
Million Song	SSE-PT+Obj	**4.54**	**7.04**	**9.16**	**11.09**	**12.90**
Dataset	SSE-PT+Cur	4.53	5.81	5.94	6.34	6.60

Table 9. Evaluating personalization on IUS with varying K

Method		IUS@10	IUS@20	IUS@30	IUS@40	IUS@50
		(the smaller the better)				
Amazon	SSE-PT+Obj	7.76%	13.66%	16.84%	19.21%	20.43%
Books	SSE-PT+Cur	**03.59%**	**5.62%**	**7.22%**	**8.63%**	**8.77%**
Yelp	SSE-PT+Obj	1.38%	23.73%	32.60%	37.07%	39.75%
Restaurants	SSE-PT+Cur	**0.72%**	**10.82%**	**20.43%**	**26.74%**	**31.73%**
Million Song	SSE-PT+Obj	4.80%	5.98%	7.46%	9.33%	15.51%
Dataset	SSE-PT+Cur	**1.61%**	**2.91%**	**3.31%**	**4.44%**	**4.45%**

high response likelihood to invite consumption. The personalization factor for assessing the stimulus (surprise) amount helps the recommender achieve smaller inter-user similarity.

For the near future, we plan to extend the framework to generate a *sequence* of recommendations that are able to transport users from the borders of their current *Comfort Zone* to "as-yet-too-alien" items that the system might persuade them to appreciate. Finally, since our idea relies on the availability of the user access and rating history with a recommender system, how to apply the framework in a "cold-start" mode without relying much on user history is also our future research question.

ACKNOWLEDGMENTS

This research is supported by National Science Foundation (NSF) (Award #1910696). We are grateful to NSF to make this research possible.

REFERENCES

[1] Fakhri Abbas and Xi Niu. 2019. Computational Serendipitous Recommender System Frameworks: A Literature Survey. In *2019 IEEE/ACS 16th International Conference on Computer Systems and Applications (AICCSA)*. IEEE, 1–8.

[2] Eytan Bakshy, Solomon Messing, and Lada A Adamic. 2015. Exposure to ideologically diverse news and opinion on Facebook. *Science* 348, 6239 (2015), 1130–1132.

[3] Andrew G Barto, Satinder Singh, and Nuttapong Chentanez. 2004. Intrinsically motivated learning of hierarchical collections of skills. In *Proceedings of the 3rd International Conference on Development and Learning*. 112–19.

[4] Daniel E Berlyne. 1966. Curiosity and exploration. *Science* 153, 3731 (1966), 25–33.

[5] Thierry Bertin-Mahieux, Daniel P.W. Ellis, Brian Whitman, and Paul Lamere. 2011. The Million Song Dataset. In *Proceedings of the 12th International Conference on Music Information Retrieval (ISMIR 2011)*.

[6] Gerlof Bouma. 2009. Normalized (pointwise) mutual information in collocation extraction. *Proceedings of GSCL* (2009), 31–40.

[7] Zhihua Cui, Xianghua Xu, XUE Fei, Xingjuan Cai, Yang Cao, Wensheng Zhang, and Jinjun Chen. 2020. Personalized recommendation system based on collaborative filtering for IoT scenarios. *IEEE Transactions on Services Computing* 13, 4 (2020), 685–695.

[8] Yashar Deldjoo, Maurizio Ferrari Dacrema, Mihai Gabriel Constantin, Hamid Eghbal-Zadeh, Stefano Cereda, Markus Schedl, Bogdan Ionescu, and Paolo Cremonesi. 2019. Movie genome: alleviating new item cold start in movie recommendation. *User Modeling and User-Adapted Interaction* 29, 2 (2019), 291–343.

[9] Xiangyu Fan and Xi Niu. 2018. Implementing and evaluating serendipity in delivering personalized health information. *ACM Transactions on Management Information Systems (TMIS)* 9, 2 (2018), 1–19.

[10] Meadhbh Foster and Mark T Keane. 2013. Surprise: Youve got some explaining to do. *arXiv preprint arXiv:1308.2236* (2013).

[11] Zhe Fu, Xi Niu, and Mary Lou Maher. 2023. Deep Learning Models for Serendipity Recommendations: A Survey and New Perspectives. *Comput. Surveys* (2023).

[12] Zhe Fu, Xi Niu, and Li Yu. 2023. Wisdom of Crowds and Fine-Grained Learning for Serendipity Recommendations. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 739–748.

[13] Zhe Fu, Li Yu, and Xi Niu. 2022. TRACE: Travel reinforcement recommendation based on location-aware context extraction. *ACM Transactions on Knowledge Discovery from Data (TKDD)* 16, 4 (2022), 1–22.

[14] Kazjon Grace and Mary Lou Maher. 2015. Surprise and reformulation as meta-cognitive processes in creative design. In *Proceedings of the Third Annual Conference on Advances in Cognitive Systems ACS*. 8.

[15] Kazjon Grace, Mary Lou Maher, David Wilson, and Nadia Najjar. 2017. Personalised specific curiosity for computational design systems. In *Design Computing and Cognition'16*. Springer, 593–610.

[16] Kazjon Grace, Mary Lou Maher, David C Wilson, and Nadia A Najjar. 2016. Combining CBR and deep learning to generate surprising recipe designs. In *International Conference on Case-Based Reasoning*. Springer, 154–169.

[17] Kazjon Grace, Mary Lou Maher2 Maryam Mohseni, and Rafael Pérez y Pérez. 2017. Encouraging p-creative behaviour with computational curiosity. In *Proceedings of the 8th International Conference on Computational Creativity. Association for Computational Creativity*.

[18] Xiangnan He, Zhankui He, Xiaoyu Du, and Tat-Seng Chua. 2018. Adversarial personalized ranking for recommendation. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*. 355–364.

[19] Jonathan Ho, Ajay Jain, and Pieter Abbeel. 2020. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems* 33 (2020), 6840–6851.

[20] Yelp Inc. 2020. Yelp open dataset. Available at https://www.yelp.com/dataset. (2020). Accessed: 2021-12-13.

[21] Laurent Itti and Pierre F Baldi. 2006. Bayesian surprise attracts human attention. In *Advances in neural information processing systems*. 547–554.

[22] Kalervo Järvelin and Jaana Kekäläinen. 2002. Cumulated gain-based evaluation of IR techniques. *ACM Transactions on Information Systems (TOIS)* 20, 4 (2002), 422–446.

[23] Yehuda Koren. 2008. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 426–434.

[24] Denis Kotkov, Shuaiqiang Wang, and Jari Veijalainen. 2016. A survey of serendipity in recommender systems. *Knowledge-Based Systems* 111 (2016), 180–192.

[25] Youfang Leng, Li Yu, and Xi Niu. 2022. Dynamically aggregating individuals' social influence and interest evolution for group recommendations. *Information Sciences* 614 (2022), 223–239.

[26] Huiyuan Li, Li Yu, Xi Niu, Youfang Leng, and Qihan Du. 2023. Sequential and Graphical Cross-Domain Recommendations with a Multi-View Hierarchical Transfer Gate. *ACM Transactions on Knowledge Discovery from Data* (2023).

[27] Pan Li, Maofei Que, Zhichao Jiang, Yao Hu, and Alexander Tuzhilin. 2020. PURS: Personalized Unexpected Recommender System for Improving User Satisfaction. In *Fourteenth ACM Conference on Recommender Systems*. 279–288.

[28] Xiaoyan Li and W Bruce Croft. 2003. Time-based language models. In *Proceedings of the twelfth international conference on Information and knowledge management*. ACM, 469–475.

[29] Xueqi Li, Wenjun Jiang, Weiguang Chen, Jie Wu, and Guojun Wang. 2019. HAES: A new hybrid approach for movie recommendation with elastic serendipity. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*. 1503–1512.
[30] Dawen Liang, Rahul G Krishnan, Matthew D Hoffman, and Tony Jebara. 2018. Variational autoencoders for collaborative filtering. In *Proceedings of the 2018 world wide web conference*. 689–698.
[31] George Loewenstein. 1994. The psychology of curiosity: A review and reinterpretation. *Psychological bulletin* 116, 1 (1994), 75.
[32] Luís Macedo and Amílcar Cardoso. 1999. Towards artificial forms of surprise and curiosity. In *Proceedings of the European Conference on Cognitive Science, S. Bagnara, Ed.* Citeseer, 139–144.
[33] Luís Macedo and Amílcar Cardoso. 2001. Modeling forms of surprise in an artificial agent. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, Vol. 23.
[34] Luís Macedo and Amílcar Cardoso. 2005. The role of surprise, curiosity and hunger on exploration of unknown environments populated with entities. In *Proceedings of the Portuguese Conference on Artificial Intelligence*. 47–53.
[35] Pramit Mazumdar, Bidyut Kr Patra, and Korra Sathya Babu. 2020. Cold-start point-of-interest recommendation through crowdsourcing. *ACM Transactions on the Web (TWEB)* 14, 4 (2020), 1–36.
[36] Julian McAuley, Christopher Targett, Qinfeng Shi, and Anton van den Hengel. 2015. Image-based Recommendations on Styles and Substitutes. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 43–52.
[37] Kathryn Merrick and Rob Saunders Mary Lou Maher. 2008. Achieving adaptable behaviour in intelligent rooms using curious supervised learning agents. (2008).
[38] Kathryn E Merrick and Mary Lou Maher. 2009. *Motivated reinforcement learning: curious characters for multiuser games*. Springer Science & Business Media.
[39] Robert K Merton. 1968. The Matthew effect in science: The reward and communication systems of science are considered. *Science* 159, 3810 (1968), 56–63.
[40] Wulf-Uwe Meyer, Rainer Reisenzein, and Achim Schützwohl. 1997. Toward a process analysis of emotions: The case of surprise. *Motivation and Emotion* 21, 3 (1997), 251–274.
[41] Marwa Hussien Mohamed, Mohamed Helmy Khafagy, Heba Elbeh, and Ahmed Mohamed Abdalla. 2019. Sparsity and cold start recommendation system challenges solved by hybrid feedback. *International Journal of Engineering Research and Technology* 12, 12 (2019), 2735–2742.
[42] Xi Niu. 2018. An adaptive recommender system for computational serendipity. In *Proceedings of the 2018 ACM SIGIR International Conference on Theory of Information Retrieval*. 215–218.
[43] Xi Niu and Fakhri Abbas. 2017. A framework for computational serendipity. In *Adjunct Publication of the 25th Conference on User Modeling, Adaptation and Personalization*. 360–363.
[44] Xi Niu and Fakhri Abbas. 2019. Computational surprise, perceptual surprise, and personal background in text understanding. In *Proceedings of the 2019 Conference on Human Information Interaction and Retrieval*. 343–347.
[45] Xi Niu, Fakhri Abbas, Mary Lou Maher, and Kazjon Grace. 2018. Surprise Me If You Can: Serendipity in Health Information. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. ACM, 23.
[46] Xi Niu and Ahmad Al-Doulat. 2021. LuckyFind: Leveraging surprise to improve user satisfaction and inspire curiosity in a recommender system. In *Proceedings of the 2021 Conference on Human Information Interaction and Retrieval*. 163–172.
[47] Xi Niu, Wlodek Zadrozny, Kazjon Grace, and Weimao Ke. 2018. Computational surprise in information retrieval. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*. 1427–1429.
[48] Pierre-Yves Oudeyer. 2004. Intelligent adaptive curiosity: a source of self-development. (2004).
[49] Eli Pariser. 2011. *The filter bubble: How the new personalized web is changing what we read and how we think*. Penguin.
[50] Rajesh PN Rao and Dana H Ballard. 1999. Predictive coding in the visual cortex: a functional interpretation of some extra-classical receptive-field effects. *Nature neuroscience* 2, 1 (1999), 79.
[51] Rob Saunders and John S Gero. 2004. Curious agents and situated design evaluations. *AI EDAM* 18, 2 (2004), 153–161.
[52] Jürgen Schmidhuber. 1991. Adaptive confidence and adaptive curiosity. In *Institut fur Informatik, Technische Universitat Munchen, Arcisstr. 21, 800 Munchen 2*. Citeseer.
[53] Jürgen Schmidhuber. 1991. Curious model-building control systems. In *Neural Networks, 1991. 1991 IEEE International Joint Conference on*. IEEE, 1458–1463.
[54] Jürgen Schmidhuber. 1999. Artificial curiosity based on discovering novel algorithmic predictability through coevolution. In *Evolutionary Computation, 1999. CEC 99. Proceedings of the 1999 Congress on*, Vol. 3. IEEE, 1612–1618.
[55] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. 2015. Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning*. PMLR, 2256–2265.
[56] Jan Storck, Sepp Hochreiter, and Jürgen Schmidhuber. 1995. Reinforcement driven information acquisition in non-deterministic environments. In *Proceedings of the international conference on artificial neural networks, Paris*, Vol. 2. Citeseer, 159–164.
[57] Fei Sun, Jun Liu, Jian Wu, Changhua Pei, Xiao Lin, Wenwu Ou, and Peng Jiang. 2019. BERT4Rec: Sequential recommendation with bidirectional encoder representations from transformer. In *Proceedings of the 28th ACM international conference on information and*

knowledge management. 1441–1450.
[58] Jianing Sun, Wei Guo, Dengcheng Zhang, Yingxue Zhang, Florence Regol, Yaochen Hu, Huifeng Guo, Ruiming Tang, Han Yuan, Xiuqiang He, et al. 2020. A framework for recommending accurate and diverse items using bayesian graph convolutional neural networks. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2030–2039.
[59] Masaki Suwa, JS Gero, and Terry Purcell. 1999. Unexpected discoveries and s-inventions of design requirements: A key to creative designs. *Computational Models of Creative Design IV, Key Centre of Design Computing and Cognition, University of Sydney, Sydney, Australia* (1999), 297–320.
[60] Emre Ugur, Mehmet R Dogar, Maya Cakmak, and Erol Sahin. 2007. Curiosity-driven learning of traversability affordance on a mobile robot. In *Development and Learning, 2007. ICDL 2007. IEEE 6th International Conference on*. IEEE, 13–18.
[61] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems* 30 (2017).
[62] Lev Vygotsky. 1978. Interaction between learning and development. *Readings on the development of children* 23, 3 (1978), 34–41.
[63] Wenjie Wang, Yiyan Xu, Fuli Feng, Xinyu Lin, Xiangnan He, and Tat-Seng Chua. 2023. Diffusion Recommender Model. In *Proceedings of the 46th International ACM SIGIR conference on Research and Development in Information Retrieval*. ACM.
[64] Liwei Wu, Shuqing Li, Cho-Jui Hsieh, and James Sharpnack. 2020. SSE-PT: Sequential recommendation via personalized transformer. In *Fourteenth ACM Conference on Recommender Systems*. 328–337.
[65] Qiong Wu, Chunyan Miao, and Zhiqi Shen. 2012. A curious learning companion in virtual learning environment. In *Fuzzy Systems (FUZZ-IEEE), 2012 IEEE International Conference on*. IEEE, 1–8.
[66] Wilhelm Max Wundt. 1874. *Grundzüge de physiologischen Psychologie*. Vol. 1. W. Engelman.
[67] Fuzheng Zhang, Kai Zheng, Nicholas Jing Yuan, Xing Xie, Enhong Chen, and Xiaofang Zhou. 2015. A novelty-seeking based dining recommender system. In *Proceedings of the 24th International Conference on World Wide Web*. 1362–1372.
[68] Mingwei Zhang, Yang Yang, Rizwan Abbas, Ke Deng, Jianxin Li, and Bin Zhang. 2021. SNPR: A Serendipity-Oriented Next POI Recommendation Model. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*. 2568–2577.
[69] Shichao Zhang and Jiaye Li. 2021. Knn classification with one-step computation. *IEEE Transactions on Knowledge and Data Engineering* (2021).
[70] Shichao Zhang, Jiaye Li, and Yangding Li. 2022. Reachable distance function for KNN classification. *IEEE Transactions on Knowledge and Data Engineering* (2022).
[71] Pengfei Zhao and Dik Lun Lee. 2016. How much novelty is relevant?: It depends on your curiosity. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*. ACM, 315–324.
[72] Tao Zhou, Zoltán Kuscsik, Jian-Guo Liu, Matúš Medo, Joseph Rushton Wakeling, and Yi-Cheng Zhang. 2010. Solving the apparent diversity-accuracy dilemma of recommender systems. *Proceedings of the National Academy of Sciences* 107, 10 (2010), 4511–4515.
[73] Reza Jafari Ziarani and Reza Ravanmehr. 2021. Serendipity in Recommender Systems: A Systematic Literature Review. *Journal of Computer Science and Technology* 36, 2 (2021), 375–396.