Controllable Recommenders using Deep Generative Models and Disentanglement

Bhargav, S.; Kanoulas, E.

DOI
10.48550/arXiv.2110.05056

Publication date
2021

Document Version
Submitted manuscript

License
Unspecified

Citation for published version (APA):
Controllable Recommenders using Deep Generative Models and Disentanglement

Samarth Bhargav
s.bhargav@uva.nl
IRLab, University of Amsterdam
The Netherlands

Evangelos Kanoulas
e.kanoulas@uva.nl
IRLab, University of Amsterdam
The Netherlands

ABSTRACT
In this paper, we consider controllability as a means to satisfy dynamic preferences of users, enabling them to control recommendations such that their current preference is met. While deep models have shown improved performance for collaborative filtering, they are generally not amenable to fine grained control by a user, leading to the development of methods like deep language critiquing. We propose an alternate view, where instead of keyphrase based critiques, a user is provided ‘knobs’ in a disentangled latent space, with each knob corresponding to an item aspect. Disentanglement here refers to a latent space where generative factors (here, a preference towards an item category like genre) are captured independently in their respective dimensions, thereby enabling predictable manipulations, otherwise not possible in an entangled space. We propose using a (semi-)supervised disentanglement objective for this purpose, as well as multiple metrics to evaluate the controllability and the degree of personalization of controlled recommendations. We show that by updating the disentangled latent space based on user feedback, and by exploiting the generative nature of the recommender, controlled and personalized recommendations can be produced. Through experiments on two widely used collaborative filtering datasets, we demonstrate that a controllable recommender can be trained with a slight reduction in recommender performance, provided enough supervision is provided. The recommendations produced by these models appear to both conform to a user’s current preference and remain personalized.

KEYWORDS
Collaborative Filtering, Representation Learning, Recommendation, Disentangled Representation Learning, Controllable Recommendation, Deep Critiquing

1 INTRODUCTION
Auto-encoder based architectures have shown impressive performance in collaborative filtering with implicit feedback [19, 25, 30]. However, these methods fail to model short term or dynamic preferences of a user. Subsequently, methods to tackle this explicitly have been proposed, for example, through conversations via a conversational recommender system [13], or critiquing recommendations using language or keyphrases [18, 23, 35, 37]. In contrast to keyphrases, typically mined from reviews or descriptions (for instance), this work considers building controllable or critiquable recommenders using attribute data for items, which can be used to construct preference distributions at a user level. This is in turn used in the critiquing process or to express a short-term preference, allowing a user to exert control over a recommender in a meaningful manner. For instance, a user who typically watches Action movies, or (as a consequence) exclusively receives Action recommendations, can now explicitly request personalized movies of other genres, like Animation.

Controllability can be achieved by utilizing the generative nature of certain Deep Generative Recommenders [18, 23, 35, 37]. Such models can be trained with a slight reduction in recommender performance, this is achieved by utilizing a co-embedding space, and instead the latent space is directly manipulated. This means no addition optimization (like in [18, 22]) is required to incorporate (multi-step) critiques. Furthermore, this also allows for ‘positive’ critiques as well as ‘soft’ control (gradual, non-binary) instead of only binary critiques.

We propose using supervision to obtain a mapping of a particular factor/aspect to a latent dimension. Since supervision signals might not be available for all users, we experiment with settings where limited data is provided.

In addition, while there have been metrics previously proposed for evaluating controllability/critiquing [32, 35, 37], these metrics don’t explicitly account for the personalization of post-critique recommendations. We propose multiple metrics to evaluate the personalization of critiqued recommendations, measuring both binary critiques as well as ‘soft’, gradual controllability. To summarize, our contributions are as follows:

- We propose using (semi-)supervised disentanglement to learn disentangled representations for controllable, personalized recommendations. Through experiments on two large scale collaborative filtering datasets using 2 types of signals as ‘knobs’, we show the proposed models produce controllable recommendations, at the cost of a slight reduction in overall recommendation performance. In addition, we experiment

1https://github.com/samarthbhargav/disentangled-control-recsys
reliably were found to be very sensitive to hyperparameters, which are difficult to interpret, compared to content-based methods. The use of deep learning for recommendation: Ma et al. [25] assume that disentanglement is generated by user behaviours on ‘macro’ and ‘micro’ levels, and show that their models outperform non-disentangled baselines. Ma et al. [26] apply disentanglement to the sequential recommendation task, while Wang et al. [34] disentangle diverse user-intents using graph-based collaborative filtering; Cui et al. [9] propose DGCF, which have ‘implicit’ (unknown) and ‘explicit’ (known) signals that are disentangled using an RNN based model with a two-step method since some computations are non-differentiable. Wang et al. [32] propose using weakly-supervised disentanglement objective on pairs of items. This allows for attribute-based item retrieval, such that items differ more/less on the provided attribute. In contrast to prior work, we use semi-supervised disentanglement, while focusing on using disentanglement for controllability.

2.3 Critiquing / Controllable Recommenders

We focus on critiquing for deep recommenders (for a full overview of earlier work in critiquing, see Chapter 13 of [29]). The following paragraphs detail approaches that leverage language or keyphrases for critiquing, followed by other approaches like ours which does not utilize any language/keyphrase. We end this sub-section with a discussion about evaluation metrics.

Recent work has focused on using language or keyphrase based critiquing [18, 23, 35, 37]. Wu et al. [35] propose the Deep Language Based Critiquing (DLC) paradigm, where (subjective) keyphrases are treated both as explanations as well as a means for a user to critique recommendations. Here, critiquing is achieved by rejecting or ‘zero’-ing out keyphrases which are co-embedded with users. Luo et al. [22] adopt the CE-NCF for multi-step critiquing; [18] use a ranking optimization instead of pairwise re-scoring to re-rank items; [23] use a VAE-based model to achieve better training stability and lower computational complexity; [37] propose using Keyphrase Activation Vectors instead of using a second ‘head’ used in [35], and adapt it for ‘positive’ critiques. Antognini et al. [2] propose T-RECS, which infers keyphrases from the intersection of user profiles and an item. [1] show that a model under a weak-supervision scheme matches or beats recommendation/explanation and critiquing performance while being much faster. We use supervised disentanglement in the user-latent space instead of a separate embedding space, while not requiring any language data (at the cost of no explanations). Prior work focuses on binary signals ex. reject/accept, while our work considers ‘soft’ critiques, allowing for a gradual tuning. Note that while we experiment only with unit critiques, compound and/or multi-step critiques is also possible within our proposed models, which we leave for future work.

Wang et al. [32] propose an orthogonal task, where items are retrieved on a ‘gradient’ given an attribute, focusing on gradual change, similar to our work. Our work focuses on recommendation performance whereas [32] focus on gradient item retrieval. Cen et al. [7] propose ComiRec for sequential recommenders, where control is used to balance diversity and recommendation accuracy, from the perspective of a designer. The method presented in [25] is closest to us, where representations are first altered, followed by a nearest neighbours search over items using a beam search. In contrast, we utilize the generative nature of the model instead of...
performing nearest neighbour search, in addition to using semi-supervised disentanglement. The authors are aware of concurrent work [28] that is close to ours. Since the paper/code was unavailable at the time of experiments/publication, we leave the evaluation of [28] for future work.

**Metrics for critiquing/controllability.** Wu et al. [35] propose the Falling MAP for negative critiques. Given a user and a keyphrase (here, genre/tag) this metric measures the difference in MAP values computed on the set of items that have the critiqued keyphrase. If this value is positive, items of the critiqued keyphrase ‘falls’ down the ranking list, which is desirable. Similarly [37] propose a follow up metric, which accounts for positive critiques as well, comparing the normalized difference of average ratings of items before/after the critique. We note that these metrics don’t explicitly model the personalization of post-critique recommendations. In contrast, the proposed metrics explicitly consider this, by measuring performance on against items for which a user’s preference is known.

# 3 METHODOLOGY

We present disentanglement of the latent space using semi-supervision as a means to control in Section 3.1. Adapting disentanglement representation learning for this task is explored in detail in Section 3.2.

## 3.1 Control using (semi-)supervised disentanglement

We train a model using a (semi-)supervised disentanglement loss, producing disentangled user representations. During inference, representations for all users are inferred (possibly offline). In the foreground, the representations are manipulated on the fly by increasing/decreasing a ‘knob’, allowing for granular change. In the background, the mapping of a knob to a latent dimension is utilized to modify the user’s representation, which is fed to the decoder to produce controlled recommendations. The following paragraphs motivate the use of disentanglement and semi-supervision to build such a recommender.

As mentioned before, the entangled nature of a typical deep recommender renders meaningful manipulation impossible. This is because (a) there is no guarantee that manipulations produce changes in a single generative factor, and (b) it is not apparent which dimension to alter given a factor. Even if we assume (b), predictability is not guaranteed, e.g. manipulating the Thriller dimension may increase the number of Thriller movies, but it may inadvertently increase the number of Children movies, which is undesirable.

We propose using supervised disentanglement to tackle these issues. Disentanglement can isolate a generative factor to a single dimension, and altering this dimension (assuming perfect disentanglement) space should produce a change only in this generative factor. Using (semi-)supervised disentanglement addresses two issues: (a) unsupervised disentanglement has several shortcomings [20], which semi-supervision may alleviate; and (b) a particular dimension is now ‘tied’ to a generative factor, ensuring that the correct dimension is being altered.

These recommendations should ideally be personalized, since they were trained with a reconstruction loss. In essence, a recommender system can be tuned on the fly to express user preferences for the current session. This process is illustrated in Figure 1.

In this paper we opt for preference distributions i.e tag/genre distributions, as generative factors or ‘knobs’ (see Section 5.1.2), since they are widely available. However, preference distributions are not independent and can also be very noisy [4, 10], in contrast to independent and exact generative factors used in disentanglement literature e.g dSprites[27]. Consequently, they may be difficult to learn and/or disentangle. The following section discusses particulars about the models.

## 3.2 Unsupervised and Semi-supervised Disentanglement

In this section, two models, the β-VAE [12, 19] and the β-TCVAE [8], are adapted for the recommendation task using a Multinomial loss [19], resulting in recommenders that can produce disentangled representations. Furthermore, this objective is further supplemented with a semi-supervised loss.

### 3.2.1 Unsupervised Disentanglement

The VAE [17] optimizes the marginal (log-)likelihood of the observed data x in expectation over the whole distribution of latent factors z. It assumes that the data is supported on a low dimensional manifold in a high dimensional space. The assumption of a latent code allows us to express the marginal distribution as follows: $p_0(x) = \int p_0(x|z)p(z)$. Since this quantity is intractable, the Evidence Lower Bound is optimized instead [14]:

$$\log p_0(x) \geq L_{VAE} = \mathbb{E}_x[\mathbb{E}_{q_\phi(z|x)}[\log p_0(x|z)] - KL[q_\phi(z|x)\|p(z)]]$$

where $q_\phi(z|x)$ is a variational approximation of the posterior distribution, which is typically a Gaussian distribution. $p(z)$, the prior, is usually an isotropic Gaussian: $p(z) = \mathcal{N}(0, I)$. Several methods modify this objective to encourage disentanglement.

### 3.2.2 β-VAE

The β-VAE [6, 12] modifies the VAE loss i.e Equation 1 by adding a β-multiplier to the KL term:

$$L_{βVAE} = \mathbb{E}_x[\mathbb{E}_{q_\phi(z|x)}[\log p_0(x|z)] - β.KL[q_\phi(z|x)\|p(z)]]$$

Disentanglement is encouraged by setting $β > 1$, which further forces the posterior distribution $q_\phi(z|x)$ to be close to the isotropic prior for every $x$ (as opposed to only on average), imposing statistical independence. However, this involves a trade-off between reconstruction quality and disentanglement [6, 12]. While Liang et al. [19] use $β < 1$ for better recommender performance, we opt for $β >= 1$ to encourage disentanglement, and achieve similar/better recommender performance by removing the Dropout layer, following Shenbin et al. [30]. An annealing strategy is used for $β$, following [6, 19]. The resulting model, which uses a Multinomial loss (see Section 5.3.2), is termed Mult-β-VAE.

### 3.2.3 β-TCVAE

Chen et al. [8] show that by imposing a constraint on only a part of the KL divergence term, reconstruction and disentanglement increases compared to the β-VAE model. The KL term
is decomposed into 3 terms (a) an index-code Mutual Information (MI) term, (b) dimension-wise KL, and (c) a total correlation (TC) term. The index-code MI captures the information between the data and latent variables, based on the empirical data distribution; the dimension-wise KL prevents deviation of the posterior from the prior dimensions; and the TC term encourages statistical independence of the posterior distribution leading to disentanglement. The key idea is that the $\beta$-VAE constrains all 3 terms, which might harm performance, while only the TC term is penalized in this model. The loss function of the TC-VAE is as follows:

$$L_{\beta-\text{TCVAE}} = \mathbb{E}_x[\mathbb{E}_{q_\phi(z|x)}[\log p_\theta(x|z)] - \alpha \log(x; n)$$

$$- \gamma \sum_j KL[q_j(z)||p(z_j)] - \beta KL[q_\phi(z)||\Pi_j q_\phi(z_j)]$$ (3)

where $\alpha$, $\gamma$ and $\beta$ are multipliers for the index code MI, the dimension wise KL and the TC term respectively. Following Chen et al. [8], $\gamma = \alpha = 1$ is used, $\beta$ is a hyperparameter. The final model, which we term Multi-TCVAE, uses a Multinomial loss (see Section 5.3.2).

3.2.4 Semi-supervised Disentanglement. The distribution $q_\phi(z|x)$ can be constrained so that a given dimension of $q(z)$ is associated with a ground truth attribute. That is, given a representation (for instance, the mean vector from the posterior) for a data point, $\mu \in \mathbb{R}^D$, and the set of known attributes $a \in \mathbb{R}^A$ for that data point, with $A \leq D$, the $j$th dimension of $\mu$ is predictive of the $j$th dimension of $a$, $1 \leq j \leq A$. This is achieved by adding a semi-supervised loss $L_s$:

$$L_{ss} = L_{\text{unsup}} + \gamma_{ss} \mathbb{E}_x R_s(q_\phi(z|x), a)$$ (4)

$L_{\text{unsup}}$ is either Equation 2 or 3. The binary cross entropy loss is used for $R_s$ [21]:

$$R_s(\mu, a) = \sum_{i=1}^A a_i \log(\sigma(\mu_i)) + (1-a_i) \log(1-\sigma(\mu_i))$$ (5)

where $\sigma(\cdot)$ is the logistic function. Note that the remaining dimensions of $z$, those that are not supervised, are not constrained by the semi-supervision loss.

4 EVALUATION METHODOLOGY

In this section we propose an evaluation methodology and a set of metrics that investigates the extent of controllability and personalization of a recommender, described in Section 4.1. We also briefly describe existing metrics for evaluating disentanglement in Section 4.2.

4.1 Evaluating Controllability

For ease of discussion, this section assumes a single factor $g_j$, s.t. given a user $u$, and a factor $g_j$, the set of all items with factor $g_j$ is $I_{g_j}$, items encountered/rated by a user is $I_u$, and items with factor $g_j$ that user $u$ has encountered is $I_{(u,g_j)} = I_{g_j} \cap I_u$ (subscripts for $z$ are dropped for brevity in the following descriptions). Note that an item can have multiple factors e.g. a movie can be both Action and Adventure.

Given a user representation $z$, the dimension corresponding to $g_j$ is obtained by using a mapping function $k, s.t. k(g_j)$ is the dimension corresponding to $g_j$ in $z$, with $k(g_j) \in \{1, \ldots, K\}$, with $K \leq D$, where $D$ is the dimensionality of the latent space. In addition, we assume that manipulations of each dimension are in $[0, 1]$ ($0$ is minimum). A manipulation entails the replacement of the $k(g_j)$ dimension with a value indicating the position of a ‘knob’, producing $z$. In addition, the top-$n$ items of $z$ is denoted by $top_n(z)$, and the number of items of a factor $g_j$ in the top-$n$ list is denoted by $\text{Count}(top_n(z), g_j)$. Finally, the $i^{th}$ element of a vector $a$ is denoted by $a_i$, with $a_i = 0$ indicating that the $i^{th}$ dimension is set to 0.

4.1.1 Desiderata. For evaluating controllability, there are a list of desiderata:

1. Increasing the values of the $k(g_j)$ dimension should push items of $g_j$ higher. In other words, given $[x^m]_{k(g_j)} > [x^n]_{k(g_j)}$ and assuming $x^m$ produces $\bar{x}^m$ and $x^n$ produces $\bar{x}^n$ respectively, then:

$$\text{Count}(top_n(\bar{x}^m), g_j) > \text{Count}(top_n(\bar{x}^n), g_j)$$

for small values of $n$. A natural consequence of this is irrelevant items $I_g - j$ should be pushed down or be replaced with relevant items, i.e. Count(top$_n(\bar{x}^m), g_j$) should decrease. Put another way, $k(g_j)$ should only control items belonging to $g_j$, and not control items of other factors.

2. In addition, it is crucial to ensure that these recommendations are personalized i.e top$_n(\bar{x}^m)$ should have items of the requested genre, but only those items that the user might like, i.e. items in $I_{(u,g_j)}$.

We assume a ranking metric R-MET, computed against a held out set of items. R-MET can be any ranking metric, e.g. NDCG, assuming the relevance of all items except the ones in the holdout set to zero. We consider five metrics in total, with the two $\delta$ metrics, $\delta_{\text{ctrl}}$ and $\delta_{\text{rel}}$, measuring the ranking changes of items of interest versus other items; and three correlation metrics measuring the gradual change observed as a dimension is manipulated: Corr, and (Corr$_{\text{ctrl}}, \text{Corr}_{\text{rand}}$). The $\delta$ metrics measures the change induced when a user prioritizes items of only one factor, reflecting a critique or a short term preference i.e. ‘show me Animation movies only’.

The correlation metrics on the other hand can capture exploratory behaviours or soft preferences (‘maybe I will try at Horror movies’), and measures to what extent the recommender is interactive and predictable. These metrics are summarized in Table 1 along with

<table>
<thead>
<tr>
<th>Metric</th>
<th>Input</th>
<th>Holdout</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\delta_{\text{ctrl}}$</td>
<td>$I_u - I_{(u,g_j)}$</td>
<td>$I_{(u,g_j)}$</td>
</tr>
<tr>
<td>$\delta_{\text{rel}}$</td>
<td>$I_u - I_{(u,g_j)}$</td>
<td>$I_{g_j} - I_{(u,g_j)}$</td>
</tr>
<tr>
<td>Corr</td>
<td>$I_u - I_{(u,g_j)}$</td>
<td>$I_{(u,g_j)}$</td>
</tr>
<tr>
<td>Corr$_{\text{ctrl}}$</td>
<td>$I_u - I_{(u,g_j)}$</td>
<td>$I_{(u,g_j)}$</td>
</tr>
<tr>
<td>Corr$_{\text{rand}}$</td>
<td>$I_u - I_{(u,g_j)}$</td>
<td>$I_{(u,g_j)}$</td>
</tr>
</tbody>
</table>

Table 1: Metrics for Controllability, listed with the inputs and holdouts used in its computation. $g_j$ is the genre being manipulated, while $g_l$ is another genre.
their inputs and holdout sets. Note that all the metrics are averaged across genres and then users.

4.1.2 \( \delta \) metrics. Similar to F-MAP [35] and PostCritRatingDiff [37], \( \delta \) measures the change in the ranking as a knob is set to its maximum value. Note, however, that only items not belonging to \( g_j \) that the user likes are fed into the model, with the intent to observe if manipulating the \( k(g_j) \) dimension increases the positions of the held-out items of genre \( g_j \) that a user likes. As a consequence, the model has to infer the correct items of \( g_j \) to recommend, without knowledge of the users preference towards items of \( g_j \). Furthermore, in contrast to prior metrics, this evaluation is personalized i.e the metric is computed against \( I(u,g_j) \) and not \( I_{\bar{g}_j} \). The process to compute the metric is detailed below.

First, before the manipulation, items of other genres that the user has liked, \( I_u - I(u,g_j) \) is used to infer \( x \). R-MET default is computed on this, using \( I(u,g_j) \) as the holdout set, measuring the baseline ranking score of personalized items belonging to \( g_j \). Second, \( I_{\bar{g}_j} \) is set to max indicating a high preference, and decoded to produce \( x \). This is used to compute R-MET ctrl on the same holdout, expressing the ranking score of personalized items belonging to \( g_j \) after manipulation. The difference between the two metrics is \( \delta_{ctrl} \):

\[
\delta_{ctrl} = R\text{-MET}_{ctrl} - R\text{-MET}_{default} \tag{6}
\]

The higher the \( \delta_{ctrl} \), the better the ability to produce personalized and controlled recommendations.

The second metric, \( \delta_{irrel} \), is a variation of \( \delta_{ctrl} \). While recommending items of genre \( g_j \) before items of other genres is important, it is desirable to recommend items that the user might like, over irrelevant items of \( g_j \). \( \delta_{irrel} \) is similar to \( \delta_{ctrl} \), except the holdout set are items of \( g_j \) not rated by the user, \( I_{\bar{g}_j} = I(u,g_j) \). If \( \delta_{irrel} \) is high, items that the user might not like are also being promoted, which may be undesirable. While these two metrics reflect the extent of controllability, granular metrics, reflecting gradual change is presented in the following section.

4.1.3 Correlation Metrics. The following metrics compute the correlation between changes in ranking metric R-MET and the values of the manipulated dimension. The correlation measure used in the experiments is the Pearson’s correlation, we leave non-linear variants to future work.

\( Corr \) quantifies the correlation of gradual manipulations (as opposed to setting it to the max value as with \( \delta_{ctrl} \)) with R-MET using \( I(u,g_j) \) as the holdout set. \( Corr_{ctrl} \) and \( Corr_{rand} \) measure the same correlation with different holdout sets and identical inputs. \( Corr_{ctrl} \) uses \( I(u,g_j) \) as the holdout set, measuring the effect of manipulation against items of \( g_j \), whereas \( Corr_{rand} \) uses \( I_u \) which measures the effect of manipulation against items of a random control factor \( g_i \). The input set for the last 2 metrics has neither items of \( g_j \) nor \( g_i \). In the ideal case, \( Corr_{ctrl} \) should be positive i.e controlling \( k(g_j) \) positively influences items of \( g_j \), but \( Corr_{rand} \) should be zero (no change) or negative (decrease, either due to replacement or demotion), meaning items of another factor are not negatively influenced.

It is our hypothesis that keeping items of irrelevant genres low in the ranking is harder if the genre being controlled is highly correlated with another genre in the data (e.g. many Action movies are also tagged Adventure). To measure this, we use two \( g_i \): \( g_{easy} \) and \( g_{difficult} \), where \( g_{easy} \) is the least co-occurring factor with \( g_j \) and \( g_{difficult} \) conversely frequently co-occurs with \( g_j \) producing \( EasyCorr_{ctrl}, EasyCorr_{rand}, DiffCorr_{ctrl} \) and \( DiffCorr_{rand} \).²

4.2 Evaluating Disentanglement

In addition to controllability, disentanglement of the user representations can also be evaluated since the ground truth generative factors are available. Several metrics have been proposed to evaluate disentanglement, such as the Mutual Information Gap (MIG) [8], the \( \beta \)-VAE metric [12] or the FactorVAE metric [15]. We evaluate disentanglement with the Mutual Information Gap (MIG) [8], due to ease of implementation, wide applicability and the unbiased nature of the metric for all hyperparameter settings [20]. Given a generative factor, the empirical mutual information (MI) between the values of the ground truth generative factors and each dimension of encoded samples from the data is computed. The MIG of this generative factor is then the difference between the highest and second highest MI values. This quantity is averaged across all generative factors to obtain the MIG score. A MIG of 1 for a particular generative factor implies that one latent dimension has MI=1, with others having MI=0.

5 EXPERIMENTS

The datasets, along with the requisite processing steps and generative factors for supervision are described in Section 5.1. The research questions are described in Section 5.2, followed by specifics of models in Section 5.3.

5.1 Datasets

In our experiments we use the Million Songs Dataset (MSD)[4] and the Movielens-20M (ML-20M) [10], two widely used collaborative filtering datasets. The steps for preparing the dataset for training are outlined in Section 5.1.1. Preference distributions are constructed as signals for supervision and for evaluating disentanglement/controllability, which is described in Section 5.1.2.

5.1.1 Dataset processing. Each dataset is processed according to the steps outlined in [19]. The users are split into test, validation and train sets. The test/validation set size for ML-20M is 10,000 users and for MSD is 50,000 users, with 20% of items held-out. The models is trained with the entire train history, and evaluated with the 20% held out set. The data is binarized by keeping ratings of four or higher and only users who have interacted with at least 5 movies are kept.

5.1.2 Generative Factors. The generative factors considered for both datasets are preference distributions computed for each user using genres (MSD/ML-20M) or tags (MSD only). The number of users, items, generative factors are reported in Table 2. For a user \( u \), \( Genre_u \) is computed by calculating the proportion of movies that belong to a genre (or tag), divided by the total number of movies that the user has watched, capturing the (global) preferences of a user towards a genre/tag. Since there are 522362 tags in MSD with many repeats, only the most frequent are picked and grouped

²Co-occurrences are computed at user level and may differ across users
Table 2: Datasets used, along with signals being considered. The last column reports the MIG score, computed on the ground truth labels against itself using 10,000 users sampled from the test set

<table>
<thead>
<tr>
<th>Dataset (ID)</th>
<th># users</th>
<th># items</th>
<th>Generative Factors</th>
<th>GT MIG</th>
</tr>
</thead>
<tbody>
<tr>
<td>ML-20M</td>
<td>136,677</td>
<td>20108</td>
<td>genre distribution</td>
<td>0.2019</td>
</tr>
<tr>
<td>MSD (Genre)</td>
<td>571,355</td>
<td>41140</td>
<td>genre distribution</td>
<td>0.3238</td>
</tr>
<tr>
<td>MSD (Tag)</td>
<td>571,355</td>
<td>41140</td>
<td>tag distribution</td>
<td>0.2525</td>
</tr>
</tbody>
</table>

5.2 Research Questions

This section details the experimental setup employed in the paper. The evaluation metric for recommender performance throughout this paper is NDCG@100, following [19, 30] i.e R-MET is NDCG for all experiments. Controllability for models with 0 supervision cannot be evaluated since \( k(.) \) is unavailable. In addition, the manipulations (which replaces \( z_{k(g,i)} \)) are produced by using the inverse CDF of the probability distribution used during training [17].

5.2.1 RQ1 How much supervision is needed for achieving controllability? How does controllability vary across datasets and models?

To investigate this, we train Multi-\(\beta\)-VAE and Multi-\(\beta\)-VAE with varying levels of supervision: \( \{0\%, 1\%, 10\%, 50\%, 100\%\} \). For each combination of model, signal, level of supervision, we compute the five metrics outlined earlier. To summarize, \( \delta_{ctrl} \) evaluates the performance gain when a user sets a knob to its maximum setting. Corr measures the gradual change via a correlation of manipulations and R-MET; the other metrics contrast the controllability of a genre against two control genres. The holdout sets used to compute the metrics above are constructed for 100 users per genre (metrics remain constant beyond 100), while ensuring that there are at least 10/3 items in the input/holdout sets described in Table 1. The number of steps taken in the latent space is 50 for computing the correlation metrics.

5.2.2 RQ2 To what extent does disentanglement affect controllability of models? The relationship between disentanglement and controllability is explored in this section, where we measure if models with disentanglement (\( \beta = 1 \)) perform better than models without (\( \beta = 0 \)). Note that \( \beta = 0 \) implies no disentanglement, but for the Multi-TCVAE, only the TC constraint is set to 0, i.e. which means \( \beta = 0, \gamma = \alpha = 1 \). In addition, we investigate if models that have higher disentanglement (MIG) scores perform better based on recommendation/controllability metrics. MIG is computed using code from [20] modified for use in pytorch. Since the exact M.I is intractable, the empirical M.I is computed by discretizing the values, following [8, 20]. The number of bins for the discretizer is set to 20, and it is computed for on 10,000 samples from the test set.

5.2.3 RQ3 What effect does introducing controllability into a model have on recommendation performance? This question deals with the relevance of both controlled and ‘default’ recommendations. This is paramount, otherwise recommendations can be irrelevant. As such, NDCG@100 is computed for all models, and model selection is done on the basis of recommendation performance on the validation set. We remind the readers that \( \delta_{irrel} \) has to be interpreted with \( \delta_{ctrl} \). That is, if \( \delta_{irrel} \) is low and \( \delta_{ctrl} \) is high, recommended items may be personalized; however, if \( \delta_{irrel} \) is higher, the models do exhibit controllability, but the recommended items might not be all personalized. Note that these metrics might be limited since a user might like these items regardless, or, such recommendations might be explicitly sought by the user. Performance for models with no disentanglement (\( \beta = 0 \)) and no supervision (\( \% = 0 \)) are also reported.

5.3 Experimental Setup

This section details specifics of models being used, along with hyperparameters:

5.3.1 Variational Distribution. Both Multi-\(\beta\)-VAE and Multi-TCVAE use an isotropic Gaussian distribution for a prior \( p(z) = N(0, 1) \), and a Gaussian distribution for the variational posterior \( q_\phi \):

\[
q_\phi(z|x) = N(z|\mu_\phi(x), \text{diag}(\sigma^2_\phi(x)))
\]

where \( \mu_\phi(x) \) and \( \sigma^2_\phi(x) \) are outputs of the encoder network.

5.3.2 Multinomial loss. We use the Multinomial log-likelihood for \( p_\theta \), which has been shown to perform well over other losses like the Gaussian/Logistic losses [19]:

\[
\log p_\theta(x|z) = \sum_i x_i \log p_i(z)
\]

The final loss used for both Multi-\(\beta\)-VAE and Multi-TCVAE methods are given in Equation 4, where \( L_{unsup} \) is either Equation 2 for the Multi-\(\beta\)-VAE or Equation 3 for the Multi-TCVAE, and second term is the binary cross entropy loss (Equation 5). In addition, \( y_{ss} = 0 \) for models with no supervision (\( \% = 0 \)), and \( y_{ss} = 1 \) otherwise (\( \% > 0 \)). To obtain recommendations, items are sorted in the descending order of the likelihood predicted by the decoder.

5.3.3 Model Hyperparameters. All models use the same neural network with a 3 layer encoder, with dimensions \( |J| \rightarrow 600 \rightarrow 600 \rightarrow 200 \) and a single layer decoder which was found optimal for a variety of settings [30]. The outputs of the encoder are two 200 dimensional real vectors representing the mean/log-variance vectors. All models were trained with Adam [16] with a learning rate of 0.001 and a batch size of 500 for 200 epochs for ML-20M and 100 epochs for MSD. For both, we tried \( \beta \in \{1, 2, 5, 10, 100\} \), with the ‘best’ model selected using NDCG@100 computed on the validation set. For the Multi-TCVAE, we set \( \alpha = \gamma = 1 \).

\[\text{Dataset (ID)} \quad \# \text{users} \quad \# \text{items} \quad \text{Generative Factors} \quad \text{GT MIG}\]

<table>
<thead>
<tr>
<th>Dataset (ID)</th>
<th># users</th>
<th># items</th>
<th>Generative Factors</th>
<th>GT MIG</th>
</tr>
</thead>
<tbody>
<tr>
<td>ML-20M</td>
<td>136,677</td>
<td>20108</td>
<td>genre distribution</td>
<td>0.2019</td>
</tr>
<tr>
<td>MSD (Genre)</td>
<td>571,355</td>
<td>41140</td>
<td>genre distribution</td>
<td>0.3238</td>
</tr>
<tr>
<td>MSD (Tag)</td>
<td>571,355</td>
<td>41140</td>
<td>tag distribution</td>
<td>0.2525</td>
</tr>
</tbody>
</table>

\[\text{values, following [8, 20]. The number of bins for the discretizer is set to 20, and it is computed for on 10,000 samples from the test set.}\]

\[\text{5.2.3 RQ3 What effect does introducing controllability into a model have on recommendation performance? This question deals with the relevance of both controlled and ‘default’ recommendations. This is paramount, otherwise recommendations can be irrelevant. As such, NDCG@100 is computed for all models, and model selection is done on the basis of recommendation performance on the validation set. We remind the readers that \( \delta_{irrel} \) has to be interpreted with \( \delta_{ctrl} \). That is, if \( \delta_{irrel} \) is low and \( \delta_{ctrl} \) is high, recommended items may be personalized; however, if \( \delta_{irrel} \) is higher, the models do exhibit controllability, but the recommended items might not be all personalized. Note that these metrics might be limited since a user might like these items regardless, or, such recommendations might be explicitly sought by the user. Performance for models with no disentanglement (\( \beta = 0 \)) and no supervision (\( \% = 0 \)) are also reported.}\]

\[\text{5.3 Experimental Setup}\]

\[\text{This section details specifics of models being used, along with hyperparameters:}\]

\[\text{5.3.1 Variational Distribution. Both Multi-\(\beta\)-VAE and Multi-TCVAE use an isotropic Gaussian distribution for a prior \( p(z) = N(0, 1) \), and a Gaussian distribution for the variational posterior \( q_\phi \):}\]

\[q_\phi(z|x) = N(z|\mu_\phi(x), \text{diag}(\sigma^2_\phi(x)))\]

\[\text{where \( \mu_\phi(x) \) and \( \sigma^2_\phi(x) \) are outputs of the encoder network.}\]

\[\text{5.3.2 Multinomial loss. We use the Multinomial log-likelihood for \( p_\theta \), which has been shown to perform well over other losses like the Gaussian/Logistic losses [19]:}\]

\[\log p_\theta(x|z) = \sum_i x_i \log p_i(z)\]

\[\text{The final loss used for both Multi-\(\beta\)-VAE and Multi-TCVAE methods are given in Equation 4, where \( L_{unsup} \) is either Equation 2 for the Multi-\(\beta\)-VAE or Equation 3 for the Multi-TCVAE, and second term is the binary cross entropy loss (Equation 5). In addition, \( y_{ss} = 0 \) for models with no supervision (\( \% = 0 \)), and \( y_{ss} = 1 \) otherwise (\( \% > 0 \)). To obtain recommendations, items are sorted in the descending order of the likelihood predicted by the decoder.}\]

\[\text{5.3.3 Model Hyperparameters. All models use the same neural network with a 3 layer encoder, with dimensions \( |J| \rightarrow 600 \rightarrow 600 \rightarrow 200 \) and a single layer decoder which was found optimal for a variety of settings [30]. The outputs of the encoder are two 200 dimensional real vectors representing the mean/log-variance vectors. All models were trained with Adam [16] with a learning rate of 0.001 and a batch size of 500 for 200 epochs for ML-20M and 100 epochs for MSD. For both, we tried \( \beta \in \{1, 2, 5, 10, 100\} \), with the ‘best’ model selected using NDCG@100 computed on the validation set. For the Multi-TCVAE, we set \( \alpha = \gamma = 1 \).}\]
Table 3: Results for ML-20M: MIG measures disentanglement, $\delta_{\text{ctrl}}$ and Corr measure control, with EasyCorr$\text{ctrl}$, EasyCorr$\text{rand}$, DiffCorr$\text{ctrl}$ and DiffCorr$\text{rand}$ contrasting control of one genre against another. NDCG and $\delta_{\text{rel}}$ (when contrasted with $\delta_{\text{ctrl}}$) measure personalization of non- and controlled recommendations respectively. Values in braces are standard errors.

<table>
<thead>
<tr>
<th>Model</th>
<th>%</th>
<th>$\beta$</th>
<th>NDCG</th>
<th>MIG</th>
<th>$\uparrow \delta_{\text{ctrl}}$</th>
<th>$\downarrow \delta_{\text{ctrl}}$</th>
<th>$\uparrow \text{Corr}$</th>
<th>EasyCorr$\text{ctrl}$</th>
<th>EasyCorr$\text{rand}$</th>
<th>DiffCorr$\text{ctrl}$</th>
<th>DiffCorr$\text{rand}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mult-$\beta$-VAE</td>
<td>0</td>
<td>0.2782</td>
<td>0.0016</td>
<td>0.2145 (0.0153)</td>
<td>0.0658 (0.0100)</td>
<td>0.8559</td>
<td>0.8880 (0.0045)</td>
<td>-0.2487 (0.0132)</td>
<td>0.8547 (0.0055)</td>
<td>-0.5793 (0.0115)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>50</td>
<td>0.2627</td>
<td>0.1265</td>
<td>0.2145 (0.0153)</td>
<td>0.0658 (0.0100)</td>
<td>0.8559</td>
<td>0.8880 (0.0045)</td>
<td>-0.2487 (0.0132)</td>
<td>0.8547 (0.0055)</td>
<td>-0.5793 (0.0115)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>0.2587</td>
<td>0.1417</td>
<td>0.2246 (0.0190)</td>
<td>0.0764 (0.0127)</td>
<td>0.8628</td>
<td>0.8677 (0.0053)</td>
<td>-0.1994 (0.0119)</td>
<td>0.8408 (0.006)</td>
<td>-0.5911 (0.0113)</td>
<td></td>
</tr>
<tr>
<td>Mult-TCAVE</td>
<td>0</td>
<td>0.2904</td>
<td>0.0007</td>
<td>0.2145 (0.0157)</td>
<td>0.0765 (0.0104)</td>
<td>0.8901</td>
<td>0.8907 (0.0044)</td>
<td>-0.2869 (0.0125)</td>
<td>0.8599 (0.0053)</td>
<td>-0.6448 (0.0104)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>50</td>
<td>0.2802</td>
<td>0.0688</td>
<td>0.2145 (0.0157)</td>
<td>0.0765 (0.0104)</td>
<td>0.8901</td>
<td>0.8907 (0.0044)</td>
<td>-0.2869 (0.0125)</td>
<td>0.8599 (0.0053)</td>
<td>-0.6448 (0.0104)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>0.2791</td>
<td>0.1819</td>
<td>0.2210 (0.0198)</td>
<td>0.0755 (0.0129)</td>
<td>0.8522</td>
<td>0.8557 (0.0055)</td>
<td>-0.2054 (0.0113)</td>
<td>0.8357 (0.0061)</td>
<td>-0.6590 (0.0101)</td>
<td></td>
</tr>
</tbody>
</table>

Table 4: Results for MSD (Genre): Models for this dataset in particular struggle to distinguish between genres (very high Corr scores).

<table>
<thead>
<tr>
<th>Model</th>
<th>%</th>
<th>$\beta$</th>
<th>NDCG</th>
<th>MIG</th>
<th>$\uparrow \delta_{\text{ctrl}}$</th>
<th>$\downarrow \delta_{\text{ctrl}}$</th>
<th>$\uparrow \text{Corr}$</th>
<th>EasyCorr$\text{ctrl}$</th>
<th>EasyCorr$\text{rand}$</th>
<th>DiffCorr$\text{ctrl}$</th>
<th>DiffCorr$\text{rand}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mult-$\beta$-VAE</td>
<td>0</td>
<td>0.3740</td>
<td>0.0184</td>
<td>0.2145 (0.0157)</td>
<td>0.0765 (0.0104)</td>
<td>0.8901</td>
<td>0.8907 (0.0044)</td>
<td>-0.2869 (0.0125)</td>
<td>0.8599 (0.0053)</td>
<td>-0.6448 (0.0104)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>50</td>
<td>0.3807</td>
<td>0.0208</td>
<td>0.2145 (0.0157)</td>
<td>0.0765 (0.0104)</td>
<td>0.8901</td>
<td>0.8907 (0.0044)</td>
<td>-0.2869 (0.0125)</td>
<td>0.8599 (0.0053)</td>
<td>-0.6448 (0.0104)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>0.3703</td>
<td>0.1360</td>
<td>0.2145 (0.0157)</td>
<td>0.0765 (0.0104)</td>
<td>0.8901</td>
<td>0.8907 (0.0044)</td>
<td>-0.2869 (0.0125)</td>
<td>0.8599 (0.0053)</td>
<td>-0.6448 (0.0104)</td>
<td></td>
</tr>
<tr>
<td>Mult-$\beta$-VAE</td>
<td>0</td>
<td>0.3675</td>
<td>0.1476</td>
<td>0.2145 (0.0157)</td>
<td>0.0770 (0.0126)</td>
<td>0.8623</td>
<td>0.8681 (0.0053)</td>
<td>-0.2304 (0.0119)</td>
<td>0.8337 (0.0062)</td>
<td>-0.6483 (0.0106)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>50</td>
<td>0.3897</td>
<td>0.0062</td>
<td>0.2145 (0.0157)</td>
<td>0.0770 (0.0126)</td>
<td>0.8623</td>
<td>0.8681 (0.0053)</td>
<td>-0.2304 (0.0119)</td>
<td>0.8337 (0.0062)</td>
<td>-0.6483 (0.0106)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>0.3898</td>
<td>0.0071</td>
<td>0.2145 (0.0157)</td>
<td>0.0770 (0.0126)</td>
<td>0.8623</td>
<td>0.8681 (0.0053)</td>
<td>-0.2304 (0.0119)</td>
<td>0.8337 (0.0062)</td>
<td>-0.6483 (0.0106)</td>
<td></td>
</tr>
</tbody>
</table>

6 RESULTS

The results of all experiments on ML-20M is reported in Table 3, on MSD (Tag) in Table 4 and finally on MSD (Genre) in Table 5. The results indicate that while increasing supervision generally tends to increase controllability, too much supervision can sometimes harm controllability, or alternatively increase controllability at the cost of more non-personalized recommendations (Section 6.1). The degree of supervision required for controllability varies across models/datasets/signals. In addition, adding a disentanglement objective helps across all metrics, but there appears to be no conclusive trend between degree of disentanglement and controllability (Section 6.2). Finally, controlled recommendations seem to be more personalized for ML-20M compared to MSD, where more non-personalized items get recommended (Section 6.3). We discuss each result in detail in the sections that follow.

6.1 Semi-Supervision and Controllability

Surprisingly, apparent control over recommendations seems to be achieved with as little as 10% of data for both datasets, as measured by high Corr and $\delta_{\text{ctrl}}$. In addition, these values increase as supervision is increased to 50% for all data and models. Supervision beyond that, however, can produce mixed results; for instance, Mult-$\beta$-VAE trained on 100% of ML-20M scores worse on $\delta_{\text{ctrl}}$ and Corr. In contrast, for MSD(Genre) and MSD(Tag) (with the exception of Mult-$\beta$-VAE for 100%), $\delta_{\text{ctrl}}$ increases with supervision. These trends also hold for the $\beta = 0$ models. Therefore, in most cases, supervision increases apparent controllability.
Table 5: Results for MSD (Tag): Models confuse between tags to a lesser extent compared to Genre

<table>
<thead>
<tr>
<th>Model</th>
<th>%</th>
<th>β</th>
<th>↑ NDCG</th>
<th>MIG</th>
<th>↑ EasyCorrcontrolled</th>
<th>↓ δcontrolled</th>
<th>↑ Corr</th>
<th>↑ EasyCorrcontrolled</th>
<th>↓ DiffCorrcontrolled</th>
<th>↑ EasyCorrrandom</th>
<th>↓ DiffCorrrandom</th>
</tr>
</thead>
<tbody>
<tr>
<td>Multi-β-VAE</td>
<td>0</td>
<td>0</td>
<td>0.2537</td>
<td>0.0006</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>0.7608</td>
<td>0.7622 (0.0056)</td>
<td>-0.0824 (0.0068)</td>
<td>0.7716 (0.0061)</td>
</tr>
<tr>
<td></td>
<td>50</td>
<td>0</td>
<td>0.2419</td>
<td>0.0006</td>
<td>0.0832 (0.0035)</td>
<td>0.0835 (0.0092)</td>
<td>-</td>
<td>0.7608</td>
<td>0.7622 (0.0056)</td>
<td>-0.0824 (0.0068)</td>
<td>0.7716 (0.0061)</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>0</td>
<td>0.2370</td>
<td>0.0006</td>
<td>0.0776 (0.0050)</td>
<td>0.0790 (0.0111)</td>
<td>-</td>
<td>0.6812</td>
<td>0.6804 (0.0064)</td>
<td>-0.0405 (0.0052)</td>
<td>0.6685 (0.0065)</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>2.5</td>
<td>0.2620</td>
<td>0.0006</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>0.7546</td>
<td>0.7577 (0.0055)</td>
<td>-0.0638 (0.0058)</td>
<td>0.7292 (0.0058)</td>
</tr>
<tr>
<td></td>
<td>50</td>
<td>0</td>
<td>0.2444</td>
<td>0.0006</td>
<td>0.0905 (0.0039)</td>
<td>0.0925 (0.0097)</td>
<td>-</td>
<td>0.6960</td>
<td>0.6989 (0.0062)</td>
<td>-0.0539 (0.0053)</td>
<td>0.6858 (0.0062)</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>0</td>
<td>0.2403</td>
<td>0.0006</td>
<td>0.0828 (0.0044)</td>
<td>0.0807 (0.0100)</td>
<td>-</td>
<td>0.7546</td>
<td>0.7577 (0.0055)</td>
<td>-0.0638 (0.0058)</td>
<td>0.7292 (0.0058)</td>
</tr>
<tr>
<td>Multi-TCVAE</td>
<td>0</td>
<td>1</td>
<td>0.2749</td>
<td>0.0107</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>0.5815</td>
<td>0.5798 (0.0083)</td>
<td>-0.0865 (0.0083)</td>
<td>0.5827 (0.0082)</td>
</tr>
<tr>
<td></td>
<td>50</td>
<td>1</td>
<td>0.2737</td>
<td>0.0233</td>
<td>0.1572 (0.0120)</td>
<td>0.5032 (0.0310)</td>
<td>-</td>
<td>0.7946</td>
<td>0.7954 (0.0041)</td>
<td>-0.1542 (0.0083)</td>
<td>0.8074 (0.0039)</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>1</td>
<td>0.2682</td>
<td>0.0847</td>
<td>0.1886 (0.0143)</td>
<td>0.5553 (0.0455)</td>
<td>-</td>
<td>0.8194</td>
<td>0.8161 (0.0039)</td>
<td>-0.1194 (0.0075)</td>
<td>0.8141 (0.0038)</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>5</td>
<td>0.2655</td>
<td>0.2244</td>
<td>0.1874 (0.0121)</td>
<td>0.5763 (0.0473)</td>
<td>-</td>
<td>0.8170</td>
<td>0.8192 (0.0037)</td>
<td>-0.1010 (0.007)</td>
<td>0.8210 (0.0038)</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>1</td>
<td>0.2781</td>
<td>0.0072</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>0.4415</td>
<td>0.4501 (0.0095)</td>
<td>-0.0359 (0.0088)</td>
<td>0.4498 (0.0095)</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>1</td>
<td>0.2761</td>
<td>0.0065</td>
<td>0.0925 (0.0086)</td>
<td>0.2490 (0.0404)</td>
<td>-</td>
<td>0.7915</td>
<td>0.7948 (0.0048)</td>
<td>-0.1374 (0.0086)</td>
<td>0.7872 (0.0049)</td>
</tr>
<tr>
<td></td>
<td>50</td>
<td>1</td>
<td>0.2670</td>
<td>0.1123</td>
<td>0.2018 (0.0151)</td>
<td>0.4289 (0.0437)</td>
<td>-</td>
<td>0.8481</td>
<td>0.8494 (0.0036)</td>
<td>-0.1290 (0.0084)</td>
<td>0.8451 (0.0036)</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>1</td>
<td>0.2628</td>
<td>0.2272</td>
<td>0.2029 (0.0153)</td>
<td>0.5042 (0.0454)</td>
<td>-</td>
<td>0.8224</td>
<td>0.8224 (0.0041)</td>
<td>-0.0709 (0.0062)</td>
<td>0.8237 (0.0039)</td>
</tr>
</tbody>
</table>

Given a model which scores high on δcontrolled (or Corr), finer grained RandCorr metrics can be analyzed, to check if changing a knob inadvertently changes other genres. Increasing supervision for ML-20M makes models score better on DiffCorrrandom (i.e., more negative), while only slightly decreasing DiffCorrcontrolled, indicating that supervision can help with similar genres in particular. While supervision initially improves EasyCorrrandom, 100% supervision actually worsens it slightly from 50%. Therefore, for ML-20M, supervision can help the model avoid confusing similar genres, at the (slight) cost of controlling dissimilar genres, especially if fully supervised.

For MSD (Genre), we first note that values of EasyCorrrandom and DiffCorrrandom are positive for this dataset, indicating that other genres are also being manipulated. Increasing supervision increases EasyCorrcontrolled and DiffCorrcontrolled, indicating that full supervision can help with confusion. In conclusion, for MSD (Genre), increasing supervision generally improves control, at the cost of other genres being also recommended, which improves with 100% supervision for MSD(Genre).

In MSD (Tag), supervision generally increases EasyCorrcontrolled and DiffCorrcontrolled. For the Multi-TCVAE, increasing supervision to 100% makes the model score better on DiffCorrrandom but worse on EasyCorrrandom. In contrast, for increasing supervision to 100% for the Multi-β-VAE makes the model score worse on both DiffCorrrandom and EasyCorrrandom. For the MSD (Tag) dataset, therefore, while supervision increases controllability, too much can cause confusion between tags.

In summary, while an increase in supervision generally increases controllability across datasets and models, and additionally reduces confusion between factors, too much supervision can harm performance by causing other genres to also be recommended alongside the one being controlled.

6.2 Disentanglement and Controllability

This section discusses if disentanglement (β >= 1) models are necessary, and if increased disentanglement as measured by the MIG score contributes to controllability.

6.2.1 Comparison with baselines. Models without disentanglement β = 0, generally score worse in most respects, compared to models with disentanglement. For the ML-20M dataset, β = 0 models have comparable controllability scores compared with models with disentanglement. However, this comes at a drastic reduction in recommender performance i.e., the best NDCG (0.2904) among the baselines is worse than the lowest NDCG among models with disentanglement (0.3675). This is also seen for the MSD (Genre) datasets, although the performance drop is smaller. For both the MSD datasets, however, the gap observed in the controllability performance of entangled/disentangled models is much greater than the gap in the ML-20M dataset, indicating that disentanglement is necessary for controllability in these datasets. Overall, models with disentanglement perform better than models without disentanglement. The next section compares the degree of disentanglement with controllability.

6.2.2 Does increased disentanglement help? For the ML-20M dataset, increased disentanglement results in lower recommendation (NDCG) performance. ML-20M models with high MIG scores perform better on δcontrolled, although this doesn’t seem to be necessary for Multi-TCVAE with % = 50, which scores high on many metrics despite having a low MIG score. For the MSD (Genre) models, a jump in MIG is accompanied by large increases in δcontrolled and δirrelevant. However, as the values of MIG scores here are either high or near zero, it’s difficult to make concrete conclusions about the precise relationship between disentanglement scores and controllability.

Note that EasyCorrcontrolled (or EasyCorrrandom) cannot be compared with DiffCorrcontrolled (or DiffCorrrandom), as the inputs for computing these metrics can be very different. In addition, we observed that a lot of values of EasyCorrrandom (at user/genre level) are equal to 0.
6.3 Controlled Recommendation Performance

We first note that as the amount of supervision, and consequently, controllability increases, average recommender performance (NDCG) decreases. However, we argue that the benefits might outweigh the cost, as this drop is relatively small while providing control for users (or practitioners). While NDCG measures performance of non-controlled recommendations, the degree of personalization of controlled recommendations can be measured by comparing $\delta_{\text{ctrl}}$ and $\delta_{\text{irrel}}$.

The results vary depending on the model or dataset. For instance, since $\delta_{\text{ctrl}}$ is much higher than $\delta_{\text{irrel}}$ for Mult-$\beta$-VAE with 100% supervision on the ML-20M, we can conclude that it recommends personalized items instead of random items that a user might like. However, the opposite is true for Mult-TCVAE, as $\delta_{\text{ctrl}}$ and $\delta_{\text{irrel}}$ are at a similar level, which means that the items recommended on controlling a genre might not be as personalized to the user. Mult-TCVAE produces the highest $\delta_{\text{ctrl}}$ scores for the ML-20M and MSD(Tag) datasets, while Mult-$\beta$-VAE produces the highest $\delta_{\text{ctrl}}$ scores for MSD(Genre). In addition, while $\delta_{\text{ctrl}}$ is higher (more items of that genre are being pushed to the top), $\delta_{\text{irrel}}$ is also very high in MSD(Genre), which means recommendations are controllable, but might not be personalized. However, as data sparsity increases, $\delta_{\text{irrel}}$ might be overestimated because of missing relevance judgements, as evidenced by high $\delta_{\text{irrel}}$ for almost all models trained on MSD. In addition to incompleteness, this evaluation also does not account for the exploratory/interactive/serendipitous nature of controllable recommendations.

In summary, controlled recommendations appear to be personalized, as evidenced by high scores of $\delta_{\text{ctrl}}$, showing that users can exert control and expect personalized recommendations to an extent. However, other items of the given genre might also be recommended, which is especially true for the MSD dataset.

7 CONCLUSION

We showed that controllable recommendations can be achieved by leveraging the generative nature of recommenders. (Semi-)supervised disentanglement is used both to tie a generative factor to a known dimension, and to enforce disentanglement. Consequently, manipulating a dimension produces controlled recommendations, allowing a user to express short-term/dynamic preferences, or to explore these recommendations in an interactive manner. We experimented with genre/tag distributions as supervision signals on two datasets. We proposed metrics to measure the extent of controllability in addition to the degree of personalization of controlled recommendations. Using these metrics, we showed that a user can control recommendations, while retaining personalization to an extent. We showed that such control comes only at a slight reduction in recommendation performance, but might require different degrees of supervision depending on the data, model or signals used.

While the experiments here detail controlling only a single dimension, multiple dimensions can be manipulated allowing for greater or more nuanced control, which we leave for future work. In addition, the dimensionality of the supervision signal is limited by the dimensionality of the latent space, which limits the applicability to large sets of knobs. Evaluating the personalization of controlled recommendations presents a challenge as outlined in the previous section, which we plan to pursue in the future. In addition, this method can be used to ‘boost’ recommendations of items belonging to a particular category, for instance, if items of category $X$ are being under-recommended (possibly due biases), they can be boosted to compensate for it.

ACKNOWLEDGMENTS

The authors would like to thank Mohammad Aliannejadi, Wilker Aziz, Maurits Bleeker, Jin Huang, Antonis Krasakis, Anna Seplarskaia, Georgios Sidiropoulos and Svitlana Vakulenko for helpful comments and feedback. The authors also thank the feedback received by reviewers during prior review processes. This research was supported by the NWO Innovational Research Incentives Scheme Vidi (016.Vidi.189.039), the NWO Smart Culture - Big Data / Digital Humanities (314-99-301), and the H2020-EU.3.4. - SOCIETAL CHALLENGES - Smart, Green And Integrated Transport (814961). All content represents the opinion of the authors, which is not necessarily shared or endorsed by their respective employers and/or sponsors.
REFERENCES


[38] Shuai Zhang, Lina Yao, Aixin Sun, and Yi Tay. 2019. Deep learning based rec-

Bhargav et al.