TSRGAN: Temporal-aware Adversarial for Sequential Recommendation Walking in the Multi-Latent Space

Anonymous Author(s)

ABSTRACT

Recently, sequential recommendation plays a critical role in our daily life, since it serves as personalized information filters to discover popular users' preferred products over time . Due to the success of the adversarial learning, a mass of research efforts start to strengthen sequential recommendation by the adversarial learning, which is able to learn complex underlying data distribution.

However, existing adversarial sequential recommendation methods suffer from mode collapse and unexplained prediction. To boost the diversity, performance, and interpretability of sequential recommendation system, we propose a novel temporal-aware adversarial framework, namely TSRGAN.

In principle, the input of traditional adversarial-based recommendation system is a noise variable sampled from normal distribution. We argue that it is hard to generate an item cover complex users' preferences(e.g. price, brand and item style) using a single latent space. Therefore, our model employs multiple latent space to generate plausible item which matches user' preferences from multiple views(e.g. Movie style, Movie release date).

Besides, previous adversarial-based recommenders focus on generating active item, but they omits that user's favour is not invariable. With GANs terminology, the recommenders only will be rewarded when seeking the peak mode, but it neglects minor mode, in other word mode collapse. In order to alleviate this issue, we design a novel diversity reward function and diversify regularization to encourage the model exploring minor mode over time and guarantee generating diversity item with reasonable.

Concretely, we propose multiple learnable latent codes to generate item matching user's preferences from different views, then we leverage the diversity reward signal to shape the distribution of multiple latent space over time. It means that the multiple latent space are sampled form different distribution instead of Gaussian distribution. Such a manipulation of the latent space can be treated as walking from plain distribution latent space to diversity distributions latent space. Further, the reward signal is modified over time, therefore, our methods names "Temporal-aware" adversarial framework .

In short, our model has two sequential stages: encode the user' characteristics and historical behaviours under multiple latent space with the Self Attention-based *generator*(G), and *discriminator*(D) try to distinguish the generator's output item from the ground truth. Besides, discriminator attempt to apply reward signal to shape the latent space distribution time by time. Extensive experiments demonstrate remarkable performance with interpretability improvement against the state-of-the-art baselines.

CCS CONCEPTS

• Information systems → Recommender systems; **• Computing methodologies → Adversarial learning**.

KEYWORDS

Sequential Recommendation, Adversarial learning, Interpretability

ACM Reference Format:
Anonymous Author(s). 2018. TSRGAN: Temporal-aware Adversarial for Sequential Recommendation Walking in the Multi-Latent Space. In *Woodstock '18: ACM Symposium on Neural Gaze Detection, June 03–05, 2018, Woodstock, NY*. ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/1122445.1122456

1 INTRODUCTION

In the big data era, the large-scale information overwhelms personal knowledge gain. [7, 15, 28, 30] Thus, modern recommendation systems are necessities in our daily life to filter out users' focus.

Mostly, users' behaviors and item' attributes update dynamically and evolving over time within daily recommendation scenario. Thus, modeling the dynamics of sequential user behaviors for providing users' preferred information serves as extremely hot research trend.

Recent years, to better capture sequential dependencies of the user-item interactions, there emerges several work in sequential recommendation task[8, 12, 26]. Notably, most methods treat the user-item interactions as a dynamic sequence and take the sequential dependencies into account to capture the current and recent preference of a user by RNNs[2] or Transformers[27].

Furthermore, researchers have incorporated rich contextual information (such as item attributes) to neural sequential recommenders[10, 11, 16, 33], which has been demonstrated that contextual information servers as a key factor to boost recommenders.

Despite the success of prior methods, they are still hard to match the users' preference distribution and item's attributes distribution over time. For the example of movie recommendation, user usually choose a movie from multiple movie attributes, such as movie style and its actor. Consequently, it still serves as a tricky issue to model a sequential pattern to provide a user favorite movie according to user's watch history and complex movies' attributes.

Therefore, existing data-driven sequential recommenders adopt an adversarial strategy[19, 34, 35] to learn complex real feature distribution, because of the capability of GANs[4]. However, previous studies has demonstrated that adversarial based training policy suffers from fatal issues(e.g. mode collapse and without

Woodstock '18, June 03–05, 2018, Woodstock, NY

ACM ISBN 978-1-4503-XXXX-X/18/06…$15.00
https://doi.org/10.1145/1122445.1122456

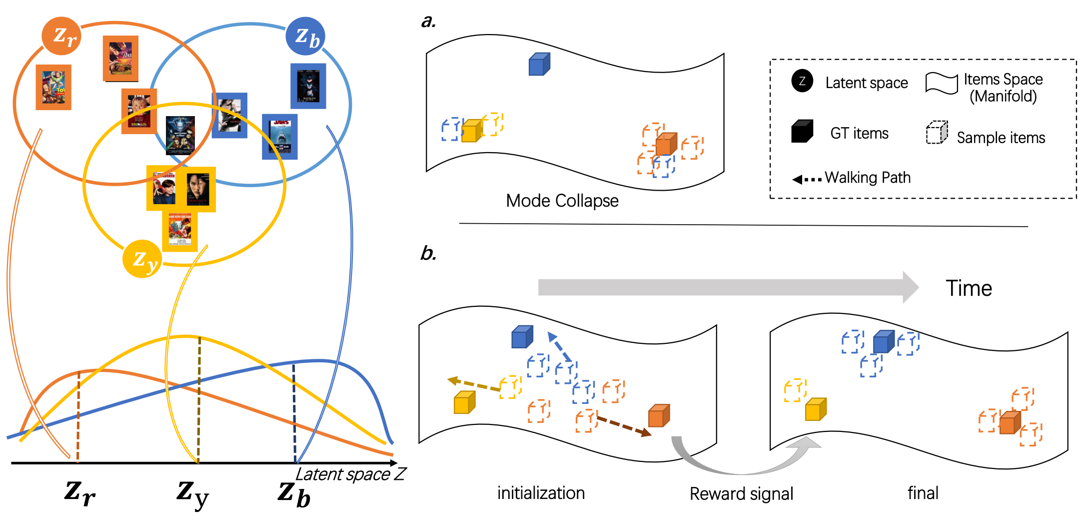

Figure 1: Motivation of our TSRGAN. Given the example of MovieLens-1M dataset, we define Z_r, Z_b and Z_y denotes different views(e.g genres in this example) of movie respectively: children's, action and romance. In detail, r, b and y are short for *red*, *blue* and *yellow* colors. Additionally, different color frames include movie posters in corresponding genres. Similarly, the different color cubes indicate generated movie in corresponding genres *On the left part:* we consider that generator attempts to sample different genres items from different distribution latent space. Therefore, we design multiple latent space to model the complex item space *On the right part:* As the (a) illustrated, previous adversarial-based recommender could recommend many item for user but collapses to a few modes (peak modes), which means the recommender is unable to cover completely user's preference. Briefly, those methods are incapable to provide diverse items. Thus, as the (b) illustrated, we propose a reward signal to modify the mean and variance of each latent codes distribution to encourage samples waking to minor modes

interpretability)[13, 23]. Basically, within existing adversarial strategy, GANs' incapacity indicates that the vanilla generative process(sampled noise from normal distribution) leads to providing similar item(without diversity). The previous research[1] demonstrates the rationale behind mode collapse: when the distributions of target and generated data is non-overlapping, the loss function convergence to constant, in other words, vanishing gradient. In such a scenario the generator will exhibit poor diversity amongst generated samples, which limits the capability of the GAN. Regarding the without interpretability issue, existing adversarial-based recommendations[6, 18, 25, 29, 34] encode the user-items interactions via black-box DNNs. It means that the the black-box recommenders cannot figure out what feature or attribute(e.g. movie style, users' characters) can most match users' preferences . In short, both of the fatal issues hurt recommendation performance disastrously, which they ignore the diversity of recommend item and decisive feature of user-items interactions.

In light of these issues, to deepen the use of adversarial structure for sequential recommendation system, we propose a novel temporal-aware adversarial framework. Notably, we argue that it is hard to generate a item cover every detail of users' preferences(e.g. price, brand and item style) using a single latent space, otherwise, we would have an unbeatable recommendation system. In other words, the expressiveness of the latent space is limited due to its finite dimensionality. Therefore, in order to recommend a vivid item, we propose to integrate multiple latent space to generate diversity sample, while decouple the latent space to investigate what feature makes effort to recommend the sample with interpretability.

Overall, our framework sets *generator G* and *discriminator D*, following standard GANs' architecture[4]. Same as the most of research on the discrete sequence data, the optimize process of the proposed TSRGAN framework is adopted by policy gradient[32]. The generator utilizes users' characters feature and user-item interactions history to predict the next items for recommendation via transformer component, while the discriminator distinguishes the generated item of from the ground truth users' preference and attempts to guide the latent space of generator via policy gradient reward over time. In detail, we adopt a novel multi-latent space instead of a single normal distribution latent space. The reward signal of discriminator shapes the distribution of the multi-latent space via update the mean and variance. In RL setting, The reward function is through that our framework could draw latent space from a plain sample distribution to diversity sample distributions.

To validate the effectiveness of the proposed TSRGAN, we conduct extensive experiments on two benchmark datasets from different domains. Experimental results show that the proposed TSRGAN is able to achieve better performance compared to several competitive methods. We further show the multi-latent space architecture is indeed useful to stabilize the learning process of adversarial policy. Finally, qualitative analysis demonstrates that our proposed framework can explicitly characterize the effect of various feature distribution over time for sequential recommendation, making the recommendation results highly interpretable. In summary, our contributions are outlined as follows

- We propose a novel sequential adversarial framework, namely TSRGAN, which largely leverages temporal information to generate diversity item and model more intuitively the distribution of multi-latent space to match complex users' preferences(items' attribute) distribution with interpretable
- To the best of our knowledge, we are the first bring latent space manipulation of adversarial framework into recommendation system task. It opens a novel path towards better understanding sequential recommendation and mitigating existing issues
- Extensive experiments conducted on two benchmark datasets demonstrate the benefits of our proposed TSRGAN beating state-of-the-art methods in top-N ranking metric

2 RELATED WORK

The trend of sequential recommendation research has raised in the last few years. We have surveyed this task and categorized it into three branches namely traditional sequence approach, matching representation methods and neural architecture. We have illustrated the relationship among different prior work in Fig **??**. Furthermore, we also review the literature of adversarial training from three related perspectives: stability, diversity and discrete.

2.1 Sequential Recommendation

Traditional Approach. Traditional approaches can be classified into sequential mining and markov-chain models. Both of these approaches have the ability of capturing sequential dependencies among the user-item interactions. In detail, sequential mining [31] proposes to mine frequently active items on user-item sequence. So that it can provide user most popularity item which attempt to match user's preferences. On the other hand,

Markov Chain-based recommendation [3] adopt Markov chain models to model the transitions over user-item interactions in a sequence, for the prediction of the next interaction. According to the specific technique used, Markov chain-based RSs are divided into basic Markov Chain-based approaches and latent Markov embedding-based approaches. The former one directly calculates the transition probability based on the explicit observations, while the latter first embeds the Markov chains into an Euclidean space and then calculates the transition probabilities between interactions based on their Euclidean distance

2.2 Generative Adversarial Network

Generative Adversarial Network (GAN) was originally proposed to mimic the generation process of given data samples. Typically, GAN consists of two components: the generative model learns to map from a latent space to a data distribution of interest, while the discriminative model distinguishes candidates produced by the generator from the true data distribution. A surge of follow-up works either improve the GAN framework by introducing more advanced training strategies, like f-divergence, Wasserstein distance, MMD constraints or explore diverse applications under GAN framework. In recommender systems, the idea of GAN has already been explored to some extent. IRGAN is firstly proposed to unify the generative model and discriminative model in the field of information retrieval. Chae et al. follow this line and further improve the training scheme by sampling a real-valued vector instead of a single item index. Moreover, personalized determinantal point process is utilized to improve recommendation diversity via adversarial training.

3 PROBLEM FORMULATION AND PRELIMINARIES

In this section, we first provide preliminaries of sequential recommendation system, and then formulate the studied problem of adversarial latent space diving into the details of the proposed method. First of all, let us denote $\mathcal{U}$ as a users set, while $\mathcal{I}$ as an item set. Respectively, $\mathcal{U} = \{U_1, U_2, \ldots, U_m\}$ and $\mathcal{I} = \left\{I_1, I_2, \ldots, I_q\right\}$ indicate the users set and item set with specific order, where m and q are the numbers of users or item. Notably, user-item interactions are arranged chronologically overtime, so we employ $\{i_1, i_2, \ldots, i_t\}$ to denote the interaction sequence of user with items at time i, where i means the specific timestamp, and t denote total time account.

Secondly, to provide better personalized recommendation, we leverage Matrix Factorization(MF)[15] to incorporate user' characteristic into our sequential procedure. The specific formulation of MF is as following:

$$\arg\min \sum_{|\mathbf{U}|,|\mathbf{I}|} \left\|\mathbf{r} - \mathbf{U}\mathbf{I}^T\right\|_F^2 + \lambda^u \|\mathbf{U}\|_F^2 + \lambda^i \|\mathbf{I}\|_F^2 \tag{1}$$

where r denotes the real user-item interactions(rating or click) history, and U and I represent the user feature after embedding layer and item latent factors sampled from latent space, respectively. Further, λ^u and λ^i are regularization coefficients

Regarding the problem of adversarial latent space, let us first explore the principle of Generative Adversarial Nets: tries to achieve the Nash equilibrium between the two players(generator and discriminator) via optimizing the following mini-max objective:

$$\begin{aligned}\min_G \max_D \mathcal{J}_{GAN} =& \mathbb{E}_{x \sim P_{\text{true}}(x)} [\log D(x)] \\ &+ \mathbb{E}_{z_{\text{noise}} \sim P(z_{\text{noise}})} [\log (1 - D(G(z_{\text{noise}})))]\end{aligned} \tag{2}$$

where x denotes input from training dataset, so $P_{\text{true}}(x)$ stands for sample data distribution. Symmetrically, z_{noise} serves as noise variable, and $P(z_{\text{noise}})$ defines deep network modeling distribution implicitly. Previous studies[19, 34, 35] mostly attempt to map from noise prior distribution $z \sim p(z)$(e.g. Gaussian $N(0, 1)$) to an target output space via signal latent space. However, these approaches suffer from mode collapse issues. Because plain latent space is hard to cover sundry recommend item space. Thus, we establish a novel walking path for multi-latent space mapping to further encourage item diversity.

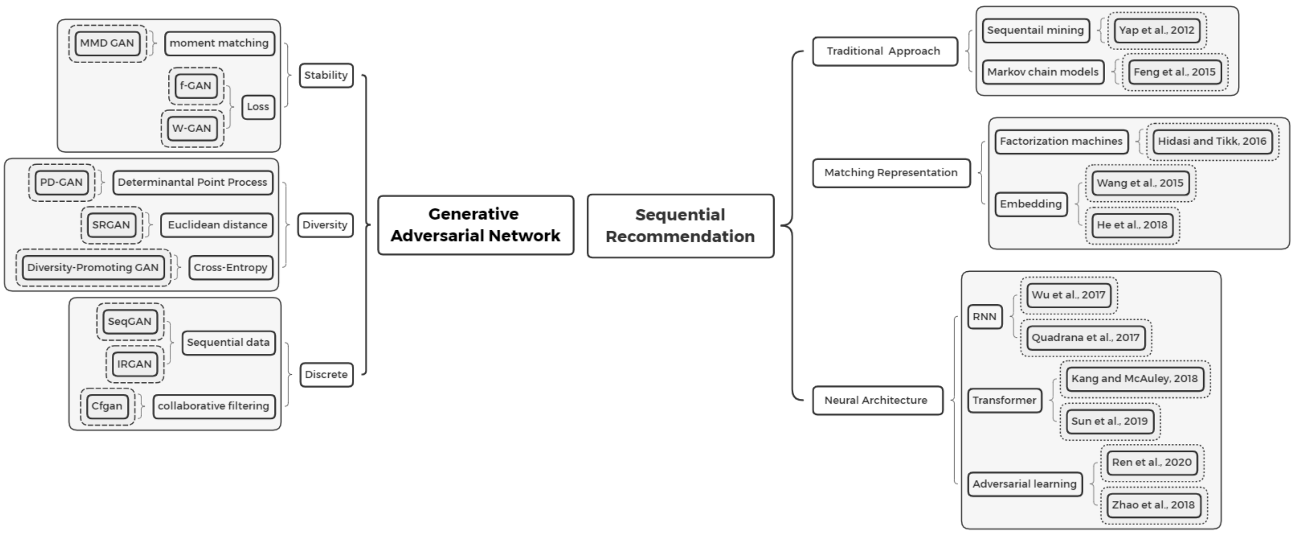

Figure 2: Illustrated the relationship among different previous research in GANs and sequential recommendation

Moreover, we leverage the N_t (μ_j, σ_j^2) to stand for the one among latent distribution shape with μ_j and σ_j^2. Where j is the j-th of the latent space of K-space, and t keeps the record of time update.

Eventually, based on the above notations, we now formulate our task of adversarial sequential recommendation. Formally, within a Item Window of user-item interactions $\{i_1, i_2, \ldots, i_{t-1}\}$ and the specific timestamp t multi-scale latent space, our system serve as personalized filter to provide top-N item at time t.

4 METHODOLOGY

In this section, we would present an overview (Fig. **??**) of the proposed TSRGAN methods.Basically, our model is composed of two components, namely Generator and Discriminator as terms as GANs[4]. The organize of this section is divided into overview procedure, generator model, discriminator model and multi-latent space.

4.1 Overview Procedure

Our neural architecture is demonstrated in Fig. **??**, and same as most GANs: the training process of our approach can been seen as play a min-max game. In detail, given the historical interactions $\{i_1, \ldots, i_{t-1}\}$ with global users' characteristics and multiple latent space N_t (μ_j, σ_j^2) at time t, the generator will recommend **predict item** t so that it can fool the discriminator. Meanwhile, discriminator tries to distinguish the real recommend item in the training set from the generator recommend item listed in item window. To provide a interpretability and diversity recommend list, discriminator utilize a reward signal to guide the generator's latent space walking from plain to diversity distribution, then we also decouple the latent space to figure out back-box training process. To this end, this min-max adversarial process can introduce G to generate rational and high-quality Top-N item list.

4.2 Generator Model

First of this part, let us defined $G_\theta\left(i_t \mid i_{1:t-1}, u_m, z_{div}\right)$ as the generator network g, optimized via θ, where θ represent learning parameters. Besides,X_I denotes the representations of item, while r_i, I_j stand for the user embedding feature and user-items historical behaviors. Notably, generator's historical item input will be encoded into a fixed-length sequence. Within the maximum length m(same as the item number), if input sequence length is less than m, we keep padding zero until the length scale to m.

Then, we elaborate our G_θ network according to the composition order. Our methods establish the generator for sequential recommendation system, consisting embedding historical behaviors, self-attention layer, dense layer(prediction) components.

Embedding. To further leverage the capability to latent space, we employ multiple latent space, where item factors T with specific time t are generated using K latent codes instead of one during training. Specifically, there emerges $\{z_1, z_2, \ldots, z_K\}$ latent codes through normalize then mapping into item factors T. The detail of the manipulation of latent space will be elaborated in *Multi-Latent Space* section 4.4. Then, we embed user vector (e.g. age, occupation) to user feature factor as a global users characteristics u_g^i, where i means the i-th user. Recently, contemporaneous work [17, 19] just embed the item sequence to capture sequential dependencies, which ignores the users' attribute severing as implied dependencies with items' feature. By contrast, to incorporate the global user's characteristic into adversarial framework, these temporal item' factor and global user' factor are element-wise product(similar to MF eq.1) to encoding matrix $E_g \in \mathbb{R}^{m \times d}$, where d is the scale of the encoding matrix. Further, to avoid sparse representation in matrix $E_g \in \mathbb{R}^{m \times d}$, we also establish a learn-able matrix $P_g \in \mathbb{R}^{m \times d}$ as a positional matrix. Thus, the embedding of user's and items' factor can be eventually formulated as $\hat{E}_g = E_g + P_g$.

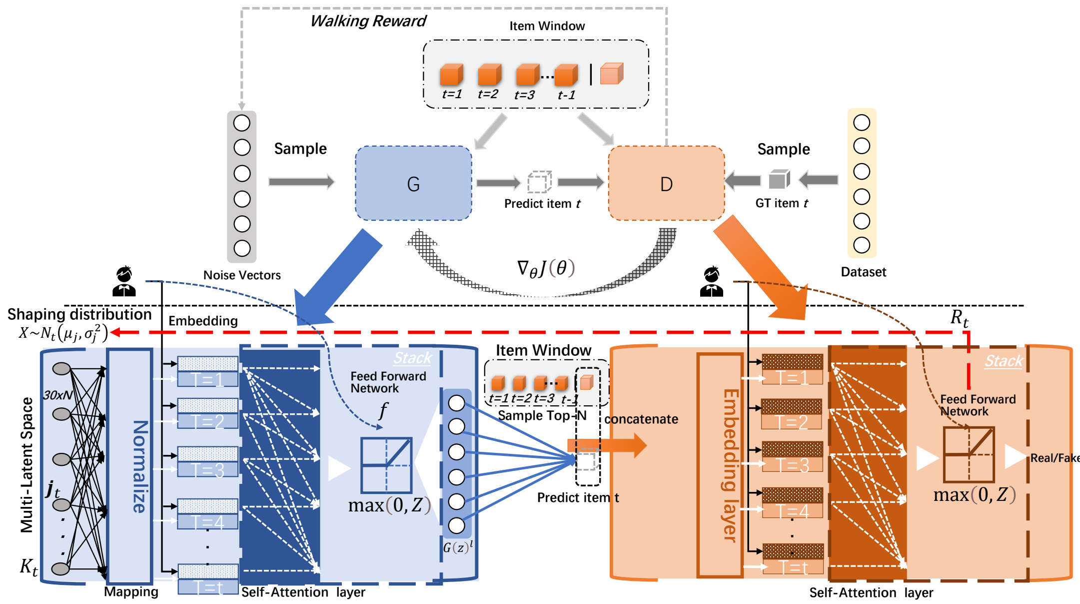

Figure 3: Neural architecture of our TSRGAN.

Self-attention Layer. Basically, most of sequential recommend system adopt various RNNs to encode the temporal relations [8] in the past few decades. Furthermore, there emerge a massive number of research[12, 19, 33] based on transformer [27]. Motivated previously, to better capture complexly sequential user-item dependence, we also leverage the self-attention encode representation matrix $\hat{E}_g$ of the embedded users and item factor. In our opinion, single self-attention layer would be intuitively hard to understand different aspect representations(e.g. item price, item brand and users' age). Thus, inspired by[19], we repeatedly stack self-attention layer to jointly learn feature from different aspect attribute subspaces at different positions. The self-attention layer could be demonstrated as following:

$$\begin{aligned} \text{Multi-Head}\left(\boldsymbol{X}^l\right) &= \left[\text{head}_1; \text{head}_2; \ldots, \text{head}_i; \ldots, \text{head}_h\right] \boldsymbol{W}^O \\ \text{head}_i &= \text{Attention}\left(\boldsymbol{X}^l \boldsymbol{W}_i^Q, \boldsymbol{X}^l \boldsymbol{W}_i^K, \boldsymbol{X}^l \boldsymbol{W}_i^V\right) \end{aligned} \tag{3}$$

where $\boldsymbol{X}^l$ is the aggregated feature at l-th layer. We define the $\boldsymbol{X}^0$ equal to $\hat{E}_g$. And $[;]$ means the concatenation operation, which can leverage multiple scale feature of item-user factor. Besides, h denotes the number of *head*, while *head* stands for output vector after self-attention with different aspect representations. Further, $\boldsymbol{W}^O$ is linear transformation matrix in $R^{d \times d}$, while $\boldsymbol{W}_i^Q \cdot \boldsymbol{W}_i^K, \boldsymbol{W}_i^V \in \mathbb{R}^{d \times d/h}$ respectively indicate the linear transformation matrix of *query, key, value (Q,K,V)* in transformer[27].To be specific, the Attention function can be defined as below:

$$\text{Attention}(\mathrm{Q}, \mathrm{K}, \mathrm{V}) = \operatorname{softmax}\left(\frac{\mathrm{QK}^T}{\sqrt{d}}\right)\mathrm{V} \tag{4}$$

where Q, K and V respectively denote $\boldsymbol{X}^l \boldsymbol{W}_i^Q, \boldsymbol{X}^l \boldsymbol{W}_i^K$ and $\boldsymbol{X}^l \boldsymbol{W}_i^V$ linear transformations embedding matrix that mentioned before. Notably, h is a scale factor, which attempts to prevent overly great value of dot product.

Basically, multiple heads self-attention has strong ability in learning linear relations[27], while it omits the non-linear aspect attribute space. Thus, we incorporate a non-linear network(ReLU-like) into self-attention layer via a point-wise feed forward function

$$\begin{aligned} \operatorname{PFFN}\left(X^l\right) &= \left[\operatorname{FFN}\left(X_1^l\right)^{\top}; \cdots; \operatorname{FFN}\left(X_n^l\right)^{\top}\right] \\ \operatorname{FFN}(x) &= \max\left(0, a_i W_1 + b_1\right) W_2 + b_2 \end{aligned} \tag{5}$$

where $W_1, W_2 \in \mathbb{R}^{d \times d}$ and $b_1, b_2 \in \mathbb{R}^d$ are learnable matrix or vector using different parameters from layer to layer. It is worth mentioning that the state-of-the-art[19] omits user' characteristic in the sequential process. More specific, the user' characteristic implicitly model the complex users' preferences, while close individual often show interest to similar item. To better model fine grained personalized feature, we define a_i as

$$\alpha_i = \frac{\exp\left(\boldsymbol{head}_i^T \boldsymbol{e}_u\right)}{\sum_{i=1}^t \exp\left(\hat{\boldsymbol{head}}_i^T \boldsymbol{e}_u\right)} \tag{6}$$

Dense Layer. To this end, at the final stage of G_θ, multiple feature $\left(X^l\right)$ is decoded through dense fully connected networks(MLPs). Eventually, generator would sample a supposititious $item_t$ as negative sample when at exactly time t

$$G_\theta\left(i_t \mid i_{1:t-1}, u_m\right) = \operatorname{MLPs}\left(X_n^L\right)\left[i_t\right] \tag{7}$$

4.3 Discriminator Model

Same as generator, we define $D_\phi(u_m, i_t)$ as the discriminator network d, which is used to distinguish whether the recommended item i_t match the potential the users' preferences item u_i, parameterized by ϕ. Concretely, we implement the discriminator via similar network that incorporates users' feature and sequential dependencies of the user-item interactions in a ranking scenario. The discriminator D_ϕ has two symmetrical point-wise architecture that share parameters and are updated by cross-entropy. The input of discriminator are predicted negative sample and ground truth sample $GTitemt$. We also model the user characterises factor via embedding to help the judgment of discriminator. The concatenate of use' factor and items' factor will be encoded by self-attention layer (eq. 3), and then processed by the non-linearity function (eq. 5). Well known, transitional GAN[4] generator serves as a filter to distinguish real from fake sample. In other words, it establish a classifier(e.g. softmax) where to determined whether the recommended item is exactly positive example(GT item). We also adopt these classic two-classification setting. Notably, distinct from prior studies that D_ϕ is only designed for qualifying historical item sequence from G_θ. Our novelty is embodied in our latent space that providing reward signal, for guiding multiple latent code to walk into diversity distribution with interpretability.

4.4 Multi-Latent Space

In the sequential recommendation task, we aim to recommend a set of items with diverse and without hurt accuracy when given temporal-aware latent space. The previous approaches[19, 34, 35] mostly random sample latent variable via single normal distribution. However, these work are seriously limited for Gaussian processes, which is not tailored for complex modeling dynamic user-item interactions. Therefore, we proposed to utilize multiple latent codes to build multiple diverse latent space matching intricate preferences' distribution over time. The principle of one among the multiple latent space illustrates in Fig. 4.

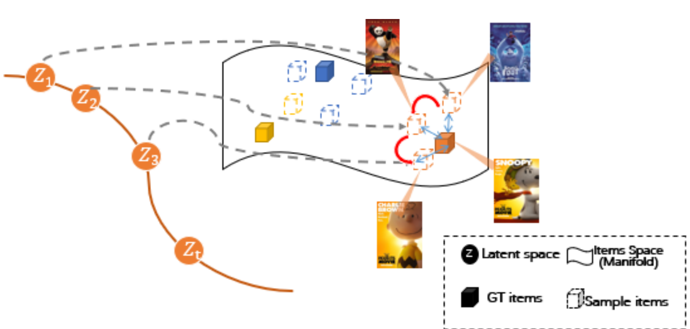

Figure 4: This figure demonstrates the temporal-aware adversarial framework training process.

Setting and RL Formulation. With reinforcement learning terminology, we can modeled the latent sample processing as a Markov Decision Processes (MDPs), formulated by a standard tuple: $(\mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R}, \gamma)$ as follows:

- State space $\mathcal{S}$: $s_t = \{r_1, \cdots, r_{t-1}\} \in r$ is defined as the interactions history of a user (e.g. movies watching history that a user browsed before time t). As aforementioned section 3, that interactions are arranged in chronological-order.
- Action space $\mathcal{A}$: the action $a_t = \{a_t^1, \cdots, a_t^K\} \in \mathcal{A}$ is to select the sample space boundary. More specific, a is to set a pair of mean and variance of a uniform latent space $U(a, b)$ at time t based on present state s_t. Notably, K is the number of action same as the number of latent space.
- Transition function $\mathcal{T}$: let $p(s_{t+1} \mid s_t, a_t)$ define that state s transits to s_{t+1} when takes action a_t. As MDPs process definition, it satisfies $p(s_{t+1} \mid s_t, a_t, \ldots, s_1, a_1) = p(s_{t+1} \mid s_t, a_t)$
- Reward $\mathcal{R}$: After system agent taking action a_t in the state s_t, the recommender agent will receive immediately reward $r(s_t, a_t)$.
- Discount factor γ: $\gamma \in [0, 1]$ is a discount factor which balances the present value and future reward.

With such notations and definitions above, the investigate of multiple latent space can be straightly defined as follows: the goal of recommender agent attempt to select the favorable latent space boundary to maximize the cumulative reward for the recommender system

Diversity Modeling. To diversify the recommender and tackle the challenge of mode collapse, the key point is to encourage G_θ to recommend sufficiently diverse item, so that the network can observe more critical modes. In addition, it is better to have "comfortable" conditioned around the observed modes, and then we could infer latent variables from the reasonable item samples, which guarantees effective inferring among the these latent variables to recommend valid items. A straightforward approach to address the issues of mode collapse, diversifying the items, is increasing the variance parameter of Gaussian distribution via gradient optimization. However, sampling from the item space and latent space are a discrete process, so gradient-based methods cannot server as a optimizer for the latent space modeling. Therefore, we adopt policy gradient-based optimization[32] to update the generator G_θ and for system agent selecting favorable multiple latent space boundary a_t over time.

Concretely, we fist seek to avoid the distance $d(\cdot)$ of two consecutive time generated items being too similar. Notably, if the recommend items at time t and $t-1$ are linearly changed, then the $d(t-1, t)$ also will change. In other word, the recommend items seen to be diversity, but still keep the mode collapse problem without exactly solving. Therefore, we need to evaluate if the mapping preserves the pair-wise distance between latent space $d_z(z_t, z_{t-1})$ and recommend item $d_g(G_\theta(z_t), G_\theta(z_{t-1}))$. In details, the d_z and d_g distance metric as follow:

$$d_z(z_t, z_{t-1}) = \|z_t - z_{t-1}\|_2 \tag{8}$$

$$d_g\left(G(z_t)^l, G(z_{t-1})^l\right) = \left\|G(z_t)^l - G(z_{t-1})^l\right\|_2 \tag{9}$$

We leverage 2-norm(l_2) distance to evaluate the pairwise distance of consecutive samples in the one among the multiple latent space and the output item space respectively. Where z_t is a sampled from $\mathbb{R}^{N \times 30}$ latent variable tensor at time t, and N is hyperparameter will be discussed in experiment section. Additionally, $G(z_t)$ denotes the output feature tensor with z_t input, while l is final(l-th) layer's feature. Furthermore, to generate diverse and reasonable items, we design a diversity regularization between single latent space and generate item space:

$$D_i^z = \frac{d_z(z_{i-1}, z_i)}{\sum_{i=1}^t d_z(z_{i-1}, z_i)} \\ D_i^g = \frac{d_g(G(z)_{i-1}^l, G(z)_i^l)}{\sum_{i=1}^t d_g(G(z)_{i-1}^l, G(z)_i^l)} \tag{10}$$

Where t denotes time and i indicates index between timestamp. To stabilize the generator and smooth gradient during backpropagation, the normalizer of $D_i^z, D_i^g \in \mathbb{R}^{1\times N}$ are treat as constant. That forces network optimizing to find suitable pairwise distance, not only update normalizer to gratify the loss function. Eventually, the formulation of the multiple latent codes diversity constraint is defined as below:

$$\mathcal{L}_{diverse}(g, z) = \frac{1}{NK}\sum_{j=1}^{K}\sum_{i=1}^{N}\max\left(0, \beta(D_i^z)_j - (D_i^g)_j\right) \tag{11}$$

Where β scales the latent space distance, and K is the number of latent codes, which we will investigate in the experiment section. When the network attempt to optimize the diverse loss (Eq. 11), it would draw the model to pay attention to other modes in the high-dimensional item space. Notably, to respect the user's historical preferences, we apply a strict limitation factor $\mathcal{L}_{\text{item}} = \|G(z^m) - I\|_2$, where z^m denotes the middle position temporal-aware latent space and I stands for ground truth item according to user historical behaviors. This item regularization force network to seek another modes rather than too novel items.

Reward Modeling. The goal of the system agent is to take proper a_t(K pairs of (μ, σ^2)) for z-space input of generator G_θ. Notably, to avoid search space oversize, we limit the action space of μ, σ^2 for (-5,5) and (0,1) respectively. Previous policy gradient recommenders[19, 29, 35] only pay attention to model context information, while they omit implicit feedback (user's preferences on relative item). More specifically, the model only will be rewarded when seeking the high peak mode, but it neglects low peak mode, in other word mode collapse. Thus, we design a effective reward function to measure the diversity and encourage the network exploring other peak mode:

$$R_t^{div} = \frac{1}{K}\sum_{k=1}^{K}\gamma^{t-1}\left(\frac{d_g\left(G(\mathbf{z}_{t-1})^l, G(\mathbf{z}_t)^l\right)}{d_z(\mathbf{z}_{t-1}, \mathbf{z}_t)}\right) \tag{12}$$

Where γ is discounted factor $\gamma \in (0, 1]$, which balances rewards between the current state and future state. K is the amount of latent codes. In detail, we leverage the ratio of the distance between recommend item with corresponding latent codes as reward to encourage the recommender to generate valid and plentiful items. More simplify, the goal of maximizing the ratio of item space distance and latent code distance is designed for generator can visit more area of target item distribution, which increases the possibility of sampling items from distinct modes. In addition, it can boost the discriminator G_ϕ when training process, because generated valid and relevantly plausible item would provide the gradients from minor peak instead of ignoring them.

To sum up, the reward function R^{div} (Eq.12) is a diversity metric to evaluate whether the multiple latent space distribution can cover the complex target distribution, and encourage recommender to sample diverse items, which alleviates the mode collapse. In other word, with reinforcement learning terminology, the diversity equation(Eq.12) servers as reward for recommender taking an action a_t^k via a fully network to pick Ks individual mean & variance pair for each latent space to match mazy items space, which forces model to seek uncommon mode peak.

4.5 Learning Framework

In this subsection, let us dive into the the TSRGAN recommending procedure, which consists of two steps: the sample process of multiple latent space, and objective learning framework.

Sample process. Firstly, traditional GAN-based recommenders[19, 29, 34, 35] just sample from single latent code z (e.g. Gaussian or uniform). Given a target item I, then it can be formulated as: $\mathbf{z}^* = \arg\min_{\mathbf{z}\in\mathcal{Z}}\mathcal{L}(G(\mathbf{z}), \mathbf{I})$. Where Z is latent space, and $\mathcal{L}(\cdot)$ denotes the objective function. To appropriately match the users' complex preferences, we proposal the multiple latent codes to produce more plausible item, which is be design as:

$$\left\{\mathbf{z}_j^*\right\}_{j=1}^{K}\left\{\boldsymbol{\alpha}_j\right\}_{j=1}^{K} = \underset{\{\mathbf{z}_j\}_{j=1}^{K}\in\mathcal{Z}}{\arg\min}\ \mathcal{L}(G(\mathbf{z}), \mathbf{I}) \tag{13}$$

Where α is an hyperparameter, which controls the importance of each z latent code. The multiple latent codes are decoupled from items feature, thus we treat each attributes(feature) equally and set α as one in order to avoid generating bias items. Eventually, the objective function $\mathcal{L}$ is cross entropy as usual deep neural recommendation system. The multiple codes will be mapping to feature vectors independently via not share parameter layer, inspired by[14] and the feature vector are concatenated as a input(denoted as z^{div}) for the G_θ. Furthermore, we will demonstrate the interpretability of the latent code feature in section 5.3.

Learning algorithm. According to the generative Adversarial Network formulation (like Eq.2), our model can be unified into a minimax game, between $G_\theta\left(i_t \mid i_{1:t-1}, u_m, z_{t-1}^{div}\right)$ and $D_\phi(u_m, i_{1:t-1})$:

$$\min_G \max_D J(G_\theta, D_\phi) = \sum_{i_t\in\mathcal{I}}\left(\mathbb{E}_{i_{1:t}^+\sim\mathcal{P}_{data}}\left[\log D_\phi\left(u_m, i_{1:t}^+\right)\right] + \mathbb{E}_{i_{1:t}\sim G_\theta}\left[\log\left(1 - D_\phi\left(G_\theta(\cdot)\right)\right)\right]\right) \tag{14}$$

where $i_{1:t}^+ \sim \mathcal{P}_{data}$ samples the positive items *GT item*, while $i_{1:t}^+ \sim G_\theta$ is the generated items by the current time stamp t and $G_\theta(\cdot)$ denotes $G_\theta(i_{1:t-1}, u_m, z_{t-1}^{div})$, including the multiple latent codes, specific user's profile and historical behaviors. In addition, $D_\phi(u_m, i_{1:t-1})$ attempt to distinguish if the item i_t belongs to user u_m preference with a differentiable network. Same as the previous work[29, 36], the sigmoid function acts a scorer $\sigma(D_\phi)$ for the probability of real or not. More specific, for the generator θ^* optimization can be redefined as a maximum process:

$$\begin{aligned}\theta^* &= \arg\min_{\theta} \sum_{i_t \in \mathcal{I}} \mathbb{E}_{i_{1:t} \sim G_\theta} \log\left(1 - \sigma\left(D_\phi\left(u_m, i^+_{1:t}\right)\right)\right) \\ &= \arg\max_{\theta} \sum_{i_t \in \mathcal{I}} \mathbb{E}_{i_{1:t} \sim G_\theta} \log\left(1 + \exp\left(D_\phi\left(u_m, i^+_{1:t}\right)\right)\right)\end{aligned} \tag{15}$$

where the optimizing of G_θ is proposal by SeqGAN[32], and adopt in most previous work[24, 29]. However, they omit the minor modes in their item searching space, which will contribute to generative poor quality items without diversity. Thus, to explicitly explore item space and cover the minor modes, we proposal a new perspective about the objective function:

$$\theta^* = \arg\max_{\theta} \sum_{i_t \in \mathcal{I}} \underbrace{\mathbb{E}_{i_{1:t} \sim G_\theta} \log\left(\lambda R^{div} + (1-\lambda)\exp\left(D_\phi\left(u_m, i^+_{1:t}\right)\right)\right)}_{\mathcal{J}^G(\theta)} \tag{16}$$

where hyperparameter λ scales between the mode explore factor R^{div} (11) and discriminator factor $\exp\left(D_\phi\left(u_m, i^+_{1:t}\right)\right)$. However, due to discrete items cannot handle the gradient optimize process[4]. Therefore, we also utilize the policy gradient for optimizing the network $\mathcal{J}^G$, after D_ϕ is fixed, to seek diversity and reasonable item:

$$\begin{aligned}\nabla_\theta J^G(\theta) &= \nabla_\theta \mathbb{E}_{i_{1:t} \sim G_\theta} \log\left(\lambda R^{div} + (1-\lambda)\exp\left(D\left(u_m, i^+_{1:t}\right)\right)\right) \\ &= \sum_{i_t \in \mathcal{I}} \nabla_\theta G_\theta\left(i_t \mid i_{1:t-1}, u_m, z^{div}_{t-1}\right) \log D^* \\ &= \sum_{i_t \in \mathcal{I}} G_\theta\left(i_t \mid i_{1:t-1}, u_m, z^{div}_{t-1}\right) \nabla_\theta G_\theta\left(i_t \mid i_{1:t-1}, u_m, z^{div}_{t-1}\right) \log D^* \\ &= \nabla_\theta \mathbb{E}_{i_{1:t} \sim G_\theta}\left[\nabla_\theta G_\theta\left(i_t \mid i_{1:t-1}, u_m, z^{div}_{t-1}\right) \log D^*\right] \\ &\simeq \frac{1}{H}\sum_{h=1}^{H} \nabla_\theta G_\theta\left(i_h \mid i_{1:t-1}, u_m, z^{div}_{t-1}\right) \log D^*\end{aligned} \tag{17}$$

where D^* is the abbreviation for $\left(\lambda R^{div} + (1-\lambda)\exp\left(D\left(u_m, i^+_{1:t}\right)\right)\right)$. In addition, H is the number of items recommended by the generator, while i_h is the h-th sampled item from the sequence. Eventually, the training procedure of proposed TSRGAN overall is presented in Algorithm **??**

5 EXPERIMENT

In this section, we first introduce some preliminaries of experiments dataset and metrics, and then analyse experiments results with several representative and state-of-the-art baselines. Secondly, we explore the interpretability of multiple latent space. To this end, we show some implementation details and our code of model will be published upon paper accepted

5.1 Preliminaries

Datasets. In order to measure the performance of our proposed, our model is conducted in two benchmark real-life datasets: MovieLens 1M movie[5], and Yahoo music. For fair comparison, we construct

Algorithm 1 The learning algorithm for our TSRGAN

```
// for each
for each i ∈ [1, 9] do
    initialize a tree T_i with only a leaf (the root);
    T = T ∪ T_i;
end for
// for all
for all c such that c ∈ RecentMBatch(E_{n−1}) do
    T = T ∪ PosSample(c);
end for
//cdscsd
for i = 1; i < n; i++ do
    // Your source here;
end for
// while
while (|E_n| ≤ L_1) and (D ≠ ϕ) do
    Selecting the most recent classifier c_i from D;
    D = D − c_i;
    E_n = E_n + c_i;
end while
```

experiment setting similar with the state-of-the-art[19]. Specifically, Yahoo! and MovieLens-1M datasets are too miscellaneous, so we filter items and users according to interaction number which are greater than ten and five, respectively. It means that selecting item with popularity and active users. Note that we divide each user-item interaction sequence into three subset following[12]: the last item in the interaction sequence serves as test set, while penultimate item is treated as validation set, and the rest items are fed into network as training set. The statistics of our two pre-process datesets as list in Tab 1:

Table 1: Summarize of two pre-process datasets

Detest	#Users	#Items	#Interactions
Movielens-1M	6,040	3,416	0.98M
Yahoo! Music[1]	15,280	140,782	3.68M

[1] https://webscope.sandbox.yahoo.com/catalog.php?datatype=r

Evaluate Metrics. We adopt two widespread Top-N metrics to measure our TSRGAN performance, Hit Rate@10 and NDCG@10, which is fair comparison with previous work[12, 19]. Clearly, Hit@10 is a metric of counting how many interactions user involve in a ten-sized list of ranked items, while NDCG@10 is normalized discounted cummulative gain which can measure relevant position of item. Additionally, we follow the training tricks[12, 19] which can reduce computation, that randomly sample 100 inactive items as negative sample to initialize D_ϕ. Eventually, we present the average performance over both metrics next section.

5.2 Results and Analysis

Let us show our develop adversarial framework TSRGAN beating against several competitive baselines. To better understand the characterises of our picking baselines, we illustrate it as a table (Tab.2):

Table 2: Summarize of baselines characterises

Baselines	Sequential recommender	Transformer-based	User characterises	Adversarial framework	Interpretability
PopRec					Strong
BPR[21]					Strong
MF[20]					Strong
IRGAN[29]				√	Weak
FPMC[22]	√		√		Weak
GRU4Rec[9]	√				Weak
Self-Attentive[12]	√	√			Weak
PLASTIC[35]	√		√	√	Weak
MFGAN[19]	√	√		√	Weak
Our	√	√	√	√	Strong

Baselines.

- PopRec: PopRec serves as a trivial baseline that recommend items based on their the number of interaction(popularity).
- BPR[21]: Bayesian Personalized Ranking is a superior work that learning personalized rankings from implicit feedback.
- MF[20]: Matrix factorization is a classical methods which utilizes coefficients of different factor in a matrix, and captures the correlation of different features
- IRGAN[29]: IRGAN is the first using GANs framework in information retrieval. This work unifies generative and discriminative methods to boost matrix factorization technology and optimize the gradient in discrete data.
- FPMC[22]: It jointly takes advantage of first-order Markov Chain and Matrix Factorization, which extracts both dynamic user-item interactions and static user preference.
- GRU4Rec[9]: GRU4Rec is a founder recommender that opening RNNs to simulate user-item interaction in session-based scenario.
- Self-Attentive[12]: It is the first Transformer-based sequential recommendation, which can capture long-term semantics feature and pay attention to relatively few actions.
- PLASTIC[35]: This methods is particularly designed for sequential movie recommendation. PLASTIC considers both user and movie information in long and short-term perspectives to model the movie preferences via adversarial training. Note that PLASTIC is implemented in MovieLens-100K, and we do our best reproduce it in our experiment setting.
- MFGAN[19]: MFGAN is an adversarial framework and it also leverages the Transformer architecture similar to [12]. This framework build more than one discriminators to decouple implied factors in context.

Table 3: Comparisons of Different Methods

Movielens-1M	**HR@10**	**NDCG@10**	**MRR**	**Diversity**
PopRec	0.4053	0.2211	0.1829	
BPR[21]	0.5381	0.2956	0.2362	
MF[20]	0.3399	0.1867	0.1593	
IRGAN[29]	0.4054	0.2187	0.1794	
FPMC[22]	0.5782	0.3625	0.3052	
GRU4Rec[9]	0.6415	0.3875	0.3548	
Self-Attentive[12]	0.7847	0.5644	0.4916	
PLASTIC[35]	0.7245	0.5489	0.4599	
MFGAN[19]	0.8026	0.6192	0.5251	
Our	**0.8244**	**0.6320**	**0.5421**	
w/o reward	0.7920	0.6208	0.5018	
Our w/o latent space	0.7413	0.5684	0.4754	
Yahoo! Music	**HR@10**	**NDCG@10**	**MRR**	**Diversity**
PopRec	0.6132	0.3864	0.3305	
BPR[21]	0.6494	0.4401	0.3872	
MF[20]	0.4040	0.2130	0.1757	
IRGAN[29]	0.4966	0.2771	0.2262	
FPMC[22]	0.5306	0.4148	0.3893	
GRU4Rec[9]	0.7155	0.4639	0.3923	
Self-Attentive[12]	0.8549	0.8005	0.7815	
PLASTIC[35]	0.8038	0.7489	0.4789	
MFGAN[19]	0.8663	0.8196	0.8020	
Our	**0.8712**	**0.8321**	**0.8392**	
w/o reward	0.8537	0.7712	0.7921	
w/o latent space	0.7329	0.7343	0.7239	

Comparison. The comparisons are illustrated in Table 3. Thus, we can observe that:

(1) Regarding the traditional recommendation system baselines(non-sequential), which omits items' time attribute, it can be clearly found that they performs weaker than sequential methods. Notably, BRR[21] has the best result among the traditional methods, because of its strong assumption between two similar items. In contrast, MF[20] cannot get used to matching sequential data, due to regression loss. Though, PopRec serves as a simple baseline just ranking according to item' popularity, it has robust performance in sequential scenario. We suppose that the k-core pre-processing policy affects the data distribution, while inadvertently boost the popularity ranking methods.

(2) Regarding Transformer-based sequential recommendation baselines, both Self-Attentive[12] and MFGAN[19] achieve remarkable performance rather than other sequential baselines. From the short review table above, key finding emerges: such an Transformer architecture[27] is inimitably suitable for capturing feature from sequence data, which also make sense to sequential recommendation. On the other hands, non-Transformer architecture methods

still performs better than traditional recommenders. Markov Chain wise method[22] could capture the both dynamic user-item relations and static user preference. RNN-based architecture [9, 35] used to be general comparison baselines in sequential data, which it can encoder context information to boost recommendation system. Further, PLASTIC[35] has better modeling performance than GRU[9] baseline, because of PLASTIC leveraging temporal-aware user' interests.

(3) Regarding incorporating user characterises baselines, it can be observed that user characterises generally can enhance the model matching ability. FPMC[22] encoder the user features in a matrix as the input of Markov Chain, which contributes to learning long-term user preference. Additionally, PLASTIC encodes both user and movie feature via RNNs to modelling long- and short-term dynamic feature. However, it is very tricky to incorporate the user feature to the user-items interactions. Thus, we set the user profile as global feature to match the complex item space via a simple attention mechanism. That policy also achieves substantial performance than user feature matrix policy.

(4) Regarding adversarial-based recommendation system baselines, it can be demonstrated that the capacity of negative samples. IRGAN opens the GAN-based recommendation system research, which utilizes the policy gradient to optimize the generator and the negative sample can boost the personalize recommender. Thus, IRGAN achieves the best performance among the non-sequential baselines. Further, PLASTIC considers extra information(e.g. user characterises, movie poster) to improve the judgment of discriminator. Our state-of-the-art baseline MFGAN leverages multiple discriminators to determine which factor servers as the key feature when generating item. The reason why MFGAN has better performance than PLASTIC is credited to encoder difference. Although, both PLASTIC and MFGAN are adversarial framework, MFGAN is based on Transformer architecture rather than RNNs encoder. Besides, overall adversarial-based suggests that negative sample provides strong signal for generating good quality items for users. Notably, the key idea of previous research using adversarial framework is feeding extra information to help discriminator judgement ability.

Eventually, our work is demonstrated that achieve remarkable performance rather than other competitive baselines. Clearly, our methods models the interactions taking advantages of above perspectives(e.g. Transformer, user characterise and adversarial framework). However, our motivation is orthogonal to previous studies that we focus building comprehensive recommendation system with diversity and interpretability. More specific, we establish multiple latent space that models different views of complex item space over time. Furthermore, we leverage a novel diversity reward to encourage the network exploring the item space, and cover the minor modes so that recommend diversity but reasonable item.

5.3 Latent Space Interpretability

5.4 Implementation Details

6 CONCLUSION

In this paper, we have proposed a Temporal-aware Generative Adversarial Network (TSRGAN) for sequential recommendation. In our framework, the generator taking user behavior sequences and user' characterises as input is used to generate possible next items, and multiple latent space are used to cover users' favourite distribution. We have constructed extensive experiments on two real-world datasets. Experimental results have shown that our approach outperforms several competitive baselines. Especially, we have found that using multiple latent space is useful to enhance the interpretability of recommendation algorithms. Currently, we consider a simple setting where multiple latent codes are separately designed. As future work, we will investigate how to design a more principled way to share relational feature across different layers. We will also consider incorporating explicit α to control the recommend types and items.

REFERENCES

[1] Arjovsky, M., and Bottou, L. Towards principled methods for training generative adversarial networks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings* (2017), OpenReview.net.

[2] Cho, K., van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (Doha, Qatar, Oct. 2014), Association for Computational Linguistics, pp. 1724–1734.

[3] Feng, S., Li, X., Zeng, Y., Cong, G., Chee, Y. M., and Yuan, Q. Personalized ranking metric embedding for next new poi recommendation. In *Proceedings of the 24th International Conference on Artificial Intelligence* (2015), IJCAI'15, AAAI Press, p. 2069–2075.

[4] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. Generative adversarial nets. In *Advances in Neural Information Processing Systems 27*, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2014, pp. 2672–2680.

[5] Harper, F. M., and Konstan, J. A. The movielens datasets: History and context. *ACM Trans. Interact. Intell. Syst. 5*, 4 (Dec. 2015).

[6] He, X., He, Z., Du, X., and Chua, T.-S. Adversarial personalized ranking for recommendation. In *The 41st International ACM SIGIR Conference on Research amp; Development in Information Retrieval* (New York, NY, USA, 2018), SIGIR '18, Association for Computing Machinery, p. 355–364.

[7] He, X., Liao, L., Zhang, H., Nie, L., Hu, X., and Chua, T.-S. Neural collaborative filtering. In *Proceedings of the 26th International Conference on World Wide Web* (Republic and Canton of Geneva, CHE, 2017), WWW '17, International World Wide Web Conferences Steering Committee, p. 173–182.

[8] Hidasi, B., and Karatzoglou, A. Recurrent neural networks with top-k gains for session-based recommendations. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management* (New York, NY, USA, 2018), CIKM '18, Association for Computing Machinery, p. 843–852.

[9] Hidasi, B., Karatzoglou, A., Baltrunas, L., and Tikk, D. Session-based recommendations with recurrent neural networks. *CoRR abs/1511.06939* (2016).

[10] Hidasi, B., Quadrana, M., Karatzoglou, A., and Tikk, D. Parallel recurrent neural network architectures for feature-rich session-based recommendations. In *Proceedings of the 10th ACM Conference on Recommender Systems* (New York, NY, USA, 2016), RecSys '16, Association for Computing Machinery, p. 241–248.

[11] Huang, J., Ren, Z., Zhao, W. X., He, G., Wen, J.-R., and Dong, D. Taxonomy-aware multi-hop reasoning networks for sequential recommendation. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining* (New York, NY, USA, 2019), WSDM '19, Association for Computing Machinery, p. 573–581.

[12] Kang, W., and McAuley, J. Self-attentive sequential recommendation. In *2018 IEEE International Conference on Data Mining (ICDM)* (2018), pp. 197–206.

[13] Karras, T., Aila, T., Laine, S., and Lehtinen, J. Progressive growing of GANs for improved quality, stability, and variation. In *International Conference on Learning Representations* (2018).

[14] Karras, T., Laine, S., and Aila, T. A style-based generator architecture for generative adversarial networks. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2019), pp. 4396–4405.

[15] Koren, Y., Bell, R., and Volinsky, C. Matrix factorization techniques for recommender systems. *Computer 42*, 8 (2009), 30–37.

[16] Li, C., Niu, X., Luo, X., Chen, Z., and Quan, C. A review-driven neural model for sequential recommendation. In *IJCAI* (2019).

[17] Lian, D., Wu, Y., Ge, Y., Xie, X., and Chen, E. Geography-aware sequential location recommendation. In *Proceedings of the 26th ACM SIGKDD International*

Conference on Knowledge Discovery amp; Data Mining (New York, NY, USA, 2020), KDD '20, Association for Computing Machinery, p. 2009–2019.

[18] Liu, Y., Xia, X., Chen, L., He, X., Yang, C., and Zheng, Z. Certifiable robustness to discrete adversarial perturbations for factorization machines. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval* (New York, NY, USA, 2020), SIGIR '20, Association for Computing Machinery, p. 419–428.

[19] Ren, R., Liu, Z., Li, Y., Zhao, W. X., Wang, H., Ding, B., and Wen, J.-R. Sequential recommendation with self-attentive multi-adversarial network. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval* (New York, NY, USA, 2020), SIGIR '20, Association for Computing Machinery, p. 89–98.

[20] Rendle, S. Factorization machines with libfm. *ACM Trans. Intell. Syst. Technol. 3*, 3 (May 2012).

[21] Rendle, S., Freudenthaler, C., Gantner, Z., and Schmidt-Thieme, L. Bpr: Bayesian personalized ranking from implicit feedback. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence* (Arlington, Virginia, USA, 2009), UAI '09, AUAI Press, p. 452–461.

[22] Rendle, S., Freudenthaler, C., and Schmidt-Thieme, L. Factorizing personalized markov chains for next-basket recommendation. In *Proceedings of the 19th International Conference on World Wide Web* (New York, NY, USA, 2010), WWW '10, Association for Computing Machinery, p. 811–820.

[23] Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., Chen, X., and Chen, X. Improved techniques for training gans. In *Advances in Neural Information Processing Systems 29*, D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, Eds. Curran Associates, Inc., 2016, pp. 2234–2242.

[24] Sun, C., Liu, H., Liu, M., Ren, Z., Gan, T., and Nie, L. Lara: Attribute-to-feature adversarial learning for new-item recommendation. In *Proceedings of the 13th International Conference on Web Search and Data Mining* (New York, NY, USA, 2020), WSDM '20, Association for Computing Machinery, p. 582–590.

[25] Tang, J., Du, X., He, X., Yuan, F., Tian, Q., and Chua, T. Adversarial training towards robust multimedia recommender system. *IEEE Transactions on Knowledge and Data Engineering 32*, 5 (2020), 855–867.

[26] Tang, J., and Wang, K. Personalized top-n sequential recommendation via convolutional sequence embedding. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining* (New York, NY, USA, 2018), WSDM '18, Association for Computing Machinery, p. 565–573.

[27] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, u., and Polosukhin, I. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems* (Red Hook, NY, USA, 2017), NIPS'17, Curran Associates Inc., p. 6000–6010.

[28] Wang, H., Wang, N., and Yeung, D.-Y. Collaborative deep learning for recommender systems. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (New York, NY, USA, 2015), KDD '15, Association for Computing Machinery, p. 1235–1244.

[29] Wang, J., Yu, L., Zhang, W., Gong, Y., Xu, Y., Wang, B., Zhang, P., and Zhang, D. Irgan: A minimax game for unifying generative and discriminative information retrieval models. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval* (New York, NY, USA, 2017), SIGIR '17, Association for Computing Machinery, p. 515–524.

[30] Xiao, H., Chen, Y., Shi, X., and Xu, G. Multi-perspective neural architecture for recommendation system. *Neural Networks 118* (2019), 280 – 288.

[31] Yap, G.-E., Li, X.-L., and Yu, P. S. Effective next-items recommendation via personalized sequential pattern mining. In *Proceedings of the 17th International Conference on Database Systems for Advanced Applications - Volume Part II* (Berlin, Heidelberg, 2012), DASFAA'12, Springer-Verlag, p. 48–64.

[32] Yu, L., Zhang, W., Wang, J., and Yu, Y. Seqgan: Sequence generative adversarial nets with policy gradient. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence* (2017), AAAI'17, AAAI Press, p. 2852–2858.

[33] Zhang, T., Zhao, P., Liu, Y., Sheng, V. S., Xu, J., Wang, D., Liu, G., and Zhou, X. Feature-level deeper self-attention network for sequential recommendation. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19* (7 2019), International Joint Conferences on Artificial Intelligence Organization, pp. 4320–4326.

[34] Zhao, W., Wang, B., Yang, M., Ye, J., Zhao, Z., Chen, X., and Shen, Y. Leveraging long and short-term information in content-aware movie recommendation via adversarial training. *IEEE Transactions on Cybernetics* (2019), 1–14.

[35] Zhao, W., Wang, B., Ye, J., Gao, Y., Yang, M., and Chen, X. Plastic: Prioritize long and short-term information in top-n recommendation using adversarial training. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18* (7 2018), International Joint Conferences on Artificial Intelligence Organization, pp. 3676–3682.

[36] Zhou, F., Yin, R., Zhang, K., Trajcevski, G., Zhong, T., and Wu, J. Adversarial point-of-interest recommendation. In *The World Wide Web Conference* (New York, NY, USA, 2019), WWW '19, Association for Computing Machinery, p. 3462–34618.