Personalized Approximate Pareto-Efficient Recommendation

Ruobing Xie*
WeChat, Tencent
Beijing, China
ruobingxie@tencent.com

Yanlei Liu*
WeChat, Tencent
Beijing, China
yanleiliu@tencent.com

Shaoliang Zhang
WeChat, Tencent
Beijing, China
modriczhang@tencent.com

Rui Wang
WeChat, Tencent
Beijing, China
rysanwang@tencent.com

Feng Xia
WeChat, Tencent
Beijing, China
xiafengxia@tencent.com

Leyu Lin
WeChat, Tencent
Beijing, China
goshawklin@tencent.com

ABSTRACT

Real-world recommendation systems usually have different learning objectives and evaluation criteria on accuracy, diversity or novelty. Therefore, multi-objective recommendation (MOR) has been widely explored to jointly model different objectives. Pareto efficiency, where no objective can be further improved without hurting others, is viewed as an optimal situation in multi-objective optimization. Recently, Pareto efficiency model has been introduced to MOR, while all existing scalarization methods only have shared objective weights for all instances. To capture users' objective-level preferences and enhance personalization in Pareto-efficient recommendation, we propose a novel Personalized Approximate Pareto-Efficient Recommendation (PAPERec) framework for multi-objective recommendation. Specifically, we design an approximate Pareto-efficient learning based on scalarization with KKT conditions that closely mimics Pareto efficiency, where users have personalized weights on different objectives. We propose a Pareto-oriented reinforcement learning module to find appropriate personalized objective weights for each user, with the weighted sum of multiple objectives' gradients considered in reward. In experiments, we conduct extensive offline and online evaluations on a real-world recommendation system. The significant improvements verify the effectiveness of PAPERec in practice. We have deployed PAPERec on WeChat Top Stories, affecting millions of users. The source codes are released in https://github.com/onepunch-cyber/PAPERec.

CCS CONCEPTS

• **Information systems** → **Recommender systems**.

KEYWORDS

Pareto efficiency, recommendation, multi-objective optimization

ACM Reference Format:
Ruobing Xie, Yanlei Liu, Shaoliang Zhang, Rui Wang, Feng Xia, and Leyu Lin. 2021. Personalized Approximate Pareto-Efficient Recommendation. In *Proceedings of the Web Conference 2021 (WWW '21), April 19–23, 2021, Ljubljana, Slovenia.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3442381.3450039

1 INTRODUCTION

Personalized recommendation aims to provide appropriate items according to user preferences, which has been widely used in various real-world scenarios of video [7], news [50], and E-commerce [32]. Most recommendation systems mainly concern recommendation accuracy measured by Click-Through-Rate (CTR), while too much dependence on CTR-oriented objectives may result in homogenization. Moreover, recommendation systems of different scenarios usually have different learning objectives and evaluation criteria. For example, novelty [26], conversion [20] and dwell time [44] are different essential factors that should be focused in news, E-commerce and video recommendations. Therefore, real-world recommendation systems should consider multiple objectives simultaneously to satisfy various demands in different scenarios.

Multi-objective optimization aims to jointly fulfill multiple objectives [17]. **Multi-objective recommendation** (MOR) has been widely adopted to jointly model diversity [23], user activeness [50], conversion [47], long-tail result [35], fairness [39], recency and relevancy [2] with recommendation accuracy. In MOR, these objectives inevitably conflict with each other during model optimization, thus it is challenging to simultaneously optimize all objectives.

To facilitate the multi-objective optimization, **Pareto efficiency** is introduced, which is regarded as an optimal state where no objective could be further improved without hurting others. Pareto optimization aims to train the model to reach the Pareto efficiency, which can be roughly categorized into two groups, namely heuristic search [8] and scalarization [26]. Heuristic search often uses evolutionary algorithms to detect Pareto-efficient statues. In contrast, scalarization combines multiple objectives into a joint loss with predefined or dynamic weights, and then optimizes the reformulated joint objective. In scalarization, Désidéri [10] proposes a Multiple-gradient descent algorithm (MGDA) for gradient-based Pareto optimization under the Karush-Kuhn-Tucker (KKT) conditions. Sener and Koltun [29] successfully applies the gradient-based scalarization to practical Pareto optimization with large-scale embeddings. Recently, Lin et al. [20] further extends MGDA to jointly optimize multiple objectives in E-commerce. All existing scalarization-based methods in MOR optimize a set of shared objective weights for all users. However, in practical recommendation, the personalization should locate in not only *item level* but also *objective level*, since

[*] Both authors contributed equally to this research. Ruobing Xie is the corresponding author (ruobingxie@tencent.com).

WWW '21, April 19–23, 2021, Ljubljana, Slovenia

ACM ISBN 978-1-4503-8312-7/21/04.
https://doi.org/10.1145/3442381.3450039

users may have different preferences on multiple objectives (e.g., some users of news recommendation may concern more about recency, while others may pay more attention to relevancy). The objective-level preferences should be considered in scalarization-based Pareto-efficient recommendations.

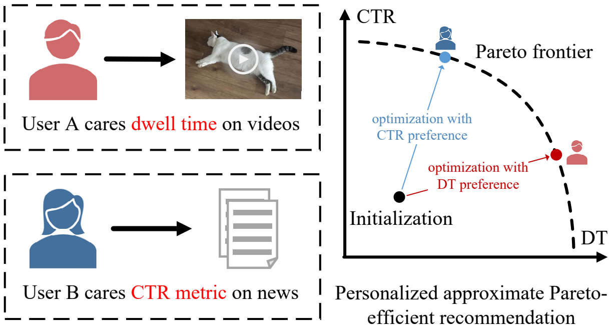

User A cares dwell time on videos
User B cares CTR metric on news
CTR
DT
Pareto frontier
optimization with CTR preference
optimization with DT preference
Initialization
Personalized approximate Pareto-efficient recommendation

Figure 1: User objective-level preferences and PAPERec.

To bring objective-level personalization into Pareto optimization in recommendation, we propose a novel framework named **Personalized Approximate Pareto-Efficient Recommendation (PAPERec)** for multi-objective recommendation. PAPERec is an approximate model that closely mimics Pareto efficiency, where users have personalized weights on different objectives. These objective weights could be viewed as certain reflections of users' objective-level preferences. Precisely, PAPERec considers multiple objectives based on scalarization, and adopts a Pareto-oriented reinforcement learning (RL) module to learn the personalized objective weights for all users in list-wise recommendation. Désidéri [10] proves that the *Pareto stationary point* is the necessary condition of Pareto efficiency, and builds an optimization problem that minimizes the L2-norm of weighted sum of all objectives' gradients to reach the Pareto efficiency. Inspired by this, our Pareto-oriented RL module directly utilizes the negative L2-norm as the reward to achieve the approximate Pareto-efficient personalized objective weights with KKT conditions. Different from other Pareto-based recommendation [20, 26], PAPERec could provide personalized objective weights to meet the diversified objective-level preferences for different users and items, improving multiple objectives simultaneously. It helps both model optimization and fusion.

In experiments, we evaluate PAPERec with competitive baselines on a real-world MOR dataset of WeChat Top Stories to verify the necessity of objective-level personalization and the effectiveness of our PAPERec in modeling such personalization. An online evaluation is also conducted to confirm the online effectiveness. PAPERec achieves the best overall performances on both offline and online evaluations, and has been deployed online. The main contributions of this work are concluded as follows:

- We propose a new Personalized Approximate Pareto-Efficient Recommendation model for multi-objective recommendation. To the best of our knowledge, we are the first attempt to bring objective-level personalization into scalarization-based Pareto-oriented recommendation.
- We design a novel Pareto-oriented RL module to learn the personalized objective weights for all users, which directly minimizes the L2-norm of weighted aggregation of multi-objective gradients to reach the Pareto stationarity.
- Sufficient offline and online evaluations have been conducted to verify the significance of objective-level personalization and the effectiveness of PAPERec in MOR. We also give an analysis to better understand the objective weights.
- PAPERec achieves the best overall performances in both online and offline evaluations with various metrics. Currently, it has been deployed on WeChat Top Stories, which affects millions of users.

2 RELATED WORKS

2.1 Recommendation

Classical recommendation algorithms such as Collaborative filtering (CF) [28], Matrix factorization [18] and Factorization machine (FM) [24] mainly concentrate on modeling user-item interactions. Recently, deep learning based models are proposed for feature interaction (e.g., FNN [46], Deep Crossing [30]) and sequential modeling (e.g., GRU4Rec [15], DSIN [11]). Wide&Deep [5] flexibly combines deep neural networks in the Deep part with feature engineering in the Wide part, which has been widely used in practical recommendation systems. DeepFM [13], NFM [14] and AFM [38] adapt FM to neural networks with DNN and attention. Autoint [31], BERT4Rec [32] and ICAN [41] further consider self-attention for feature interactions. AFN [6] uses a logarithmic transformation layer to learn adaptive-order feature interactions. Graph neural networks are also used in recommendation [22, 37]. Inspired by their successes, we rely on deep neural models to model feature interactions.

Reinforcement learning (RL) has also been verified in recommendation. It is usually designed for modeling other indirect or long-term rewards besides CTR-oriented objectives, such as user activeness [50], long-term accuracy [16, 21], diversity [51], negative feedbacks [49], and page-wise rewards [48]. Moreover, adversarial training [4] and supervised training [34, 43] are also combined to enhance RL-based recommendation. Chen et al. [3] proposes the top-k off-policy correction to balance exploitation and exploration. Fujimoto et al. [12] explores the off-policy RL without exploration. Hierarchical RL is also adapted to recommendation [42, 45]. PAPERec attempts to consider session-based objectives in recommendation feed. Therefore, it conducts a Pareto-oriented RL module to learn the personalized objective weights, and also uses an RL-based structure for each single-objective model.

2.2 Multi-objective Recommendation

Multi-objective recommendation attempts to simultaneously consider multiple objectives in a joint recommendation framework. Recommendation accuracy (e.g., CTR-oriented objectives) is the dominating factor in most real-world systems, while user friendly systems usually involve with other objectives. Multiple factors such as diversity [23, 36], user long-term activeness [50], conversion [47], long-tail performance [35], fairness [39], recency and relevancy [2] are considered in multi-objective recommendation. Some works use a general recommendation optimization framework with constraints to jointly optimize multiple goals [17, 27, 40]. Reinforcement learning is also effective that encodes multiple objectives with appropriate reward combinations [23, 47].

Pareto efficiency is a situation where no objective can be better off without making at least one objective worse off, which is usually considered in multi-objective optimization. Existing Pareto optimization can be main categorized into two groups: heuristic search [8] and scalarization [20]. Désidéri [10] proposes a Multiple-gradient descent algorithm (MGDA) to combine scalarization with Pareto-efficient SGD, using KKT condition to guide the updates of scalarization weights. Sener and Koltun [29] further improves MGDA with Frank-Wolfe algorithm to fit the large-scale learning problems in practice. Specifically in recommendation, Ribeiro et al. [25, 26] jointly consider multiple trained recommendation algorithms with a Pareto-efficient manner. It conducts an evolutionary algorithm to find the appropriate parameters for weighted model combination. Some works build their algorithms on trust-aware recommendation [1], fairness [39] and relation chaining [9] from a Pareto-efficiency aspect. Lin et al. [20] jointly models GMV and CTR in E-commerce based on MGDA with a relaxed KKT condition. All existing scalarization-based methods optimize the global weights for different objectives. However, the personalization in recommendation should locate in not only the item level, but also the objective level. Therefore, our PAPERec conducts a personalized approximate Pareto-efficient RL framework to understand users' preferences on different objectives for model training and fusion.

3 METHODOLOGY

To capture users' objective-level preferences in MOR, we propose PAPERec for better overall performance in list-wise multi-objective recommendation. In this section, we first show the basic notions used in this paper (Sec. 3.1), and then give the overall framework of PAPERec (Sec. 3.2). Next, we will introduce the core Pareto-oriented RL module with detailed discussions on its network structure and training paradigm (Sec. 3.3). Finally, we will introduce its implementation details (Sec. 3.4) and online development (Sec. 3.5).

3.1 Preliminaries

We first give a brief introduction to Pareto efficiency and the notions used in this paper. **Pareto efficiency**, also noted as Pareto optimality, is a situation in multi-objective optimization where no objective can be further improved without making at least one objective worse off. Precisely, for K learning objectives $\{L_1, \cdots, L_K\}$, we have the following definitions:

Definition 1 (Situation domination). Given a situation $L(\theta) = \{L_1(\theta), \cdots, L_K(\theta)\}$ and another one $L(\theta') = \{L_1(\theta'), \cdots, L_K(\theta')\}$, we can say the situation $L(\theta)$ *dominates* the situation $L(\theta')$, if we have $L_k(\theta) \leq L_k(\theta')$ for all objective L_k and $L(\theta) \neq L(\theta')$.

Definition 2 (Pareto efficiency). A situation $L(\theta)$ is regarded as Pareto efficient or Pareto optimal, if there is no situation in the overall situation space that dominates $L(\theta)$.

Generally, Pareto efficient situations are viewed as the optimal results for multi-objective optimization. All Pareto-efficient situations are combined to form the Pareto frontier.

3.2 Overall Framework

In PAPERec, we attempt to achieve the approximate Pareto efficiency with objective-level personalization in *list-wise MOR*. The inputs are item candidates with contextual and user features, the output is a recommended list (containing top 10 items in our system). Specifically, assuming that there are K objectives in a MOR system noted as $\{L_1(\theta), \cdots, L_K(\theta)\}$, where $L_i(\theta)$ represents the loss for the i-th objective and θ is the model parameter. It is quite difficult to simultaneously optimize all objectives, since different objectives often have conflicts. Hence, we adopt the scalarization method [20, 29], which aggregates these objectives $L_i(\theta)$ into a single $L(\theta)$ with different weights $\omega_i(u_j)$ as follows:

$$\begin{gathered} L(\theta) = \sum_{u_j \in U} \sum_{i=1}^{K} \omega_i(u_j) L_i(\theta). \\ \sum_{i=1}^{K} \omega_i(u_j) = 1,\ \omega_i(u_j) \geq 0,\ \forall u_j \in U. \end{gathered} \tag{1}$$

$u_j \in U$ indicates the j-th user u_j in the overall user set U. $\omega_i(u_j)$ represents the i-th **objective weight** of the j-th user. Different from conventional scalarization methods, we bring in the objective-level personalization to these objective weights, which could improve the overall user experience in MOR.

Specifically, PAPERec mainly contains two parts, including the *Pareto-oriented reinforcement learning* module and the specific *single-objective model* module. Pareto-oriented RL is the central module that aims to generate personalized objective weights for all users in MOR, while the specific single-objective model module consists of K separate models that are designed for multiple objectives. In model learning, the K single-objective models are first updated via the joint scalarization loss function in Eq. (1), with the personalized objective weights generated by the Pareto-oriented RL. Next, we update the Pareto-oriented RL by minimizing the L2-norm of weighted sum of all objectives' gradients as rewards. This iterative optimization can lead PAPERec to an approximate Pareto-efficient situation. Algorithm 1 gives the detailed pseudo-code of PAPERec.

Algorithm 1 Personalized Approximate Pareto-Efficient Recommendation (PAPERec):

```
Input: The $K$ loss functions $\{L_1(\theta), \cdots, L_K(\theta)\}$ of different objectives;
    The personalized objective weights $\omega_i(u_j)$ generated by the Pareto-
    efficient RL for all users and objectives.
Output: The $K$ detailed single-objective models' parameters $\theta$; The Pareto-
    oriented RL module's parameters $\psi$ and $\phi$.
Randomly initialize $\theta$, $\psi$ and $\phi$;
while not converge do
    Calculate $K$ objectives $L_i(\theta)$ via single-objective models and $\theta$;
    Update all Pareto-oriented RL parameters $\psi$ and $\phi$ via the RL loss
        $L_{RL} = L(\psi) + \beta L(\phi)$ in Eq. (13);
    Generate new objective weights $\omega_i'(u_j)$ via the Pareto-oriented
        RL module with updated $\psi'$ and $\phi'$;
    Update all single-objective models' parameters $\theta$ by optimizing
        $L_{model}$ in Eq. (18) with losses weighted by $\omega_i'(u_j)$;
end while
```

3.3 Pareto-oriented Reinforcement Learning

The Pareto-oriented reinforcement learning module attempts to generate personalized objective weights in list-wise MOR, which is the central module in PAPERec. We first introduce the definition of Pareto stationarity and its relation to Pareto efficiency, and then give a detailed introduction to the network structure and training paradigm of the Pareto-oriented RL.

3.3.1 *Pareto Stationarity.*

Désidéri [10] proposes a multiple gradient descent algorithm (MGDA) for multi-objective optimization based on scalarization. Precisely, MGDA leverages the KKT conditions and define the Pareto stationarity as follows:

Definition 3 (Pareto stationarity). A situation is regarded to be Pareto stationary, if there exist $\omega_1, \cdots, \omega_K \geq 0$ such that $\sum_{i=1}^{K} \omega_i = 1$ and $\sum_{i=1}^{K} \omega_i \nabla_\theta L_i(\theta) = 0$.

Désidéri [10] proves that if a situation is Pareto efficient, then it is Pareto stationary. Hence, the Pareto stationarity situation can be transformed into the optimization problem as:

$$\begin{aligned} &\min. \ \| \sum_{i=1}^{K} \omega_i \nabla_\theta L_i(\theta) \|_2^2. \\ &\text{s.t.} \ \sum_{i=1}^{K} \omega_i = 1, \ \omega_1, \cdots, \omega_K \geq 0. \end{aligned} \tag{2}$$

Désidéri [10] also verifies that (1) either the solution to this minimization optimization is 0, which indicates that the situation satisfies the KKT conditions and reaches the Pareto stationarity, (2) or the solution gives a descent direction that can simultaneously optimize all tasks. Moreover, Sener and Koltun [29] proves that such Pareto-stationary situation is Pareto efficient in realistic and mild conditions. Hence, some Pareto-based MOR [20] finds appropriate objective weights via this minimization optimization problem in Eq. (2) with weighted sum of objective gradients.

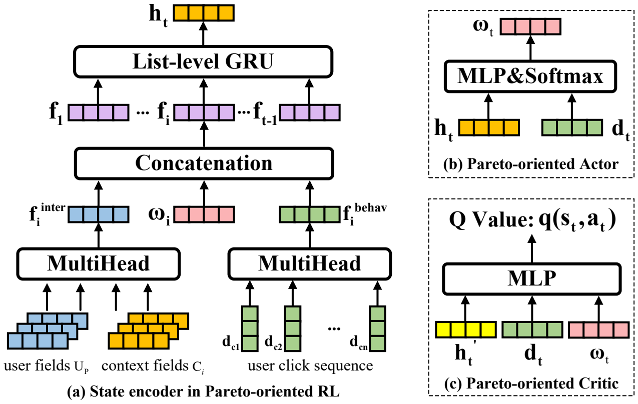
In PAPERec, we replace the shared ω_i with the personalized objective weights $\omega_i(u_j)$. Instead of directly minimizing Eq. (2), we propose an RL framework to simulate the scalarization-based Pareto optimization to approach toward the approximate Pareto stationarity. The minimization of Eq. (2) is indirectly reached as the reward in Pareto-oriented RL.

3.3.2 *Overall Architecture of Pareto-oriented RL.*

PAPERec is deployed in a list-wise recommendation. Hence, we design an RL framework to maximize long-term rewards in all positions. Precisely, we use the classical DDPG with Actor-Critic framework [19] in RL, and define the key notions as follows:

- **State** s_t: the t-th state describes the current situation when RL has recommended objective weights for previous $t-1$ items in the list. s_t contains information of user profiles, user historical behaviors, previous $t-1$ objective weights, recommendation contexts, and the t-th item features.
- **Action** a_t: the t-th action a_t is a set of objective weights at the t-th position in the list given by the Actor.
- **Reward** r_t: the t-th reward r_t is the negative L2-norm of weighted sum of all objectives' gradients at the t-th position introduced in Eq. (2).
- **Discount factor** γ: the discount factor $\gamma \in [0, 1]$ measures the importance of future rewards in list-wise MOR.

Note that the Pareto-oriented RL module only generates personalized objective weights $\omega_i(u_j)$ for different positions and items in the recommended list. The objective-specific losses $L_i(\theta)$ are given by their corresponding single-objective models in Sec. 3.4.



(a) State encoder in Pareto-oriented RL

(b) Pareto-oriented Actor

(c) Pareto-oriented Critic

Figure 2: Overall architecture of Pareto-oriented RL. The left is (a) state encoder, the right shows (b) PCritic and (c) PActor.

3.3.3 *Pareto-oriented Actor (PActor).*

PActor aims to give appropriate personalized objective weights $\omega_i(u_j)$. Fig. 2 (a) and (b) give the architecture of PActor. Precisely, when predicting at the t-th position in the final recommended list, PActor should understand the current situation of (1) user profiles and historical behaviors, (2) previous $t-1$ objective weights, (3) recommendation contexts, and (4) current t-th item features. Therefore, we take the previous $t-1$ recommended item features $\{f_1, \cdots, f_{t-1}\}$ already recommended by PAPERec as the input sequence.

The i-th item feature embedding $\boldsymbol{f_i}$ contains information of user profiles, user historical behaviors, i-th contexts and i-th item profiles. Precisely, we first calculate feature interactions $\boldsymbol{f}_i^{inter}$ between feature fields in user profiles U_p (e.g., age, gender) and the i-th context C_i (e.g., position) via self-attention as follows:

$$\boldsymbol{f}_i^{inter} = \text{Flatten}(\text{MultiHead}(\boldsymbol{U_p}, \boldsymbol{C_i})), \tag{3}$$

where MultiHead(·) is the Multi-head self-attention layer in Song et al. [31], and Flatten(·) is the 1D flatten operation. $\boldsymbol{U_p}$ and $\boldsymbol{C_i}$ are feature field embedding sets of the current user profiles and the i-th contexts. Next, we also build user behavior feature $\boldsymbol{f}^{behav}$ with user's recent click sequence $\{\boldsymbol{d}_{c_1}, \cdots, \boldsymbol{d}_{c_n}\}$ as:

$$\boldsymbol{f}_i^{behav} = \text{Flatten}(\text{MultiHead}(\boldsymbol{d}_{c_1}, \cdots, \boldsymbol{d}_{c_n})). \tag{4}$$

Finally, we concatenate the i-th objective weight embedding $\boldsymbol{\omega_i}$ with $\boldsymbol{f}_i^{inter}$, $\boldsymbol{f}_i^{behav}$ to generate the final item feature embedding $\boldsymbol{f_i}$ as follows:

$$\boldsymbol{f_i} = \text{Concat}(\boldsymbol{f}_i^{inter}, \boldsymbol{\omega_i}, \boldsymbol{f}_i^{behav}). \tag{5}$$

$\boldsymbol{\omega_i} = \{\boldsymbol{\omega}_{i1}, \cdots, \boldsymbol{\omega}_{iK}\}$ where $\boldsymbol{\omega}_{ij}$ represents the j-th objective weight of the i-th item in the list for user u.

Following [15], we use an RNN with GRU unit as the sequential encoder for the recommended item sequence as:

$$\boldsymbol{h_t} = \text{GRU}(\{\boldsymbol{f_1}, \cdots, \boldsymbol{f_{t-1}}\}). \tag{6}$$

$\boldsymbol{h_t}$ is the last output state in GRU. The operation from Eq. (3) to Eq. (6) is regarded as the state encoder of PActor. Finally, we concatenate $\boldsymbol{h_t}$ with the t-th item profile embedding $\boldsymbol{d_t}$, and feed them into a softmax layer to generate the final objective weight distribution embedding $\boldsymbol{\omega}_t \in \mathbb{R}^K$, which is formalized as:

$$\boldsymbol{\omega}_t = \mathrm{Softmax}(\mathrm{ReLU}(\mathrm{Concat}(\boldsymbol{h}_t, \boldsymbol{d}_t)\boldsymbol{W}^a + \boldsymbol{b}^a)). \tag{7}$$

$\boldsymbol{W}^a$ and $\boldsymbol{b}^a$ are the weights and bias. The objective weight embedding $\boldsymbol{\omega}_t$ is regarded as the action of Pareto-oriented RL.

3.3.4 Pareto-oriented Critic (PCritic). PCritic aims to value the current state s_t given an action a_t. The Q value $Q(s_t, a_t)$ reflects the expected return of Pareto-related rewards in the recommended list. We build the Q value as:

$$Q(s_t, a_t) = \mathbb{E}_{s_{t+1}, r_t \sim E}[r_t + \gamma_1 Q(s_{t+1}, a_{t+1})]. \tag{8}$$

γ_1 is a discount factor that measures the influence of position biases in list-wise recommendation. It forces RL module to concern more about the top items. To optimize PAPERec towards Pareto efficiency, we define the reward r_t as the negative L2-norm of weighted sum of all objectives' gradients at the t-th position, simulating the Pareto stationarity optimization introduced in Eq. (2) as follows:

$$r_t = -\|\sum_{j=1}^{K} \boldsymbol{\omega}_{tj} \nabla_\theta \bar{L}_j(\theta)\|_2^2. \tag{9}$$

$\boldsymbol{\omega}_{tj}$ is the personalized objective weight generated by the PActor, and the gradient of the j-th objective-specific loss $\nabla_\theta \bar{L}_j(\theta)$ is generated by the corresponding single-objective model with the t-th item. Theoretically under ideal conditions, through a sufficient RL training of this Pareto-oriented optimization with Eq. (8) and Eq. (9), more r_t will reach their maximization in realistic conditions as in [29]. In this case, $\boldsymbol{\omega}_t$ are learned to either satisfy the KKT conditions (when $r_t = 0$), or give an approximate descent direction that simultaneously optimizes all objectives as stated in Eq. (2). Therefore, the RL module can approximately approach the Pareto stationarity with personalized objective weights.

Specifically, we utilize a similar state encoder in PActor to build PCritic. Besides user profile and contexts, we also add feature fields of the t-th target item profiles $\boldsymbol{d}_t$ and action $\boldsymbol{\omega}_t$ in Eq. (3) to capture item-level feature interaction. Next, we follow Eq. (3) - Eq. (6) as the state encoder of PCritic to build the final hidden state $\boldsymbol{h}'_t$ via Multi-head and GRU encoders. Finally, as in Fig. 2 (c), we concatenate $\boldsymbol{h}'_t$ with both the t-th item profile $\boldsymbol{d}_t$ and the t-th action $\boldsymbol{\omega}_t$ given by PActor to predict the t-th Q value $q(s_t, a_t)$ as:

$$q(s_t, a_t) = \mathrm{ReLU}(\mathrm{Concat}(\boldsymbol{h}'_t, \boldsymbol{d}_t, \boldsymbol{\omega}_t)\boldsymbol{w}^c + b^c). \tag{10}$$

$\boldsymbol{w}^c$ and b^c are the weighs and bias of PCritic.

3.3.5 Optimization Objectives. In RL training, we use DDPG [19] to train both PActor and PCritic with off-policy strategy, and adopt the double strategy [33] with online and target networks. Similar to Zhao et al. [48], we adopt the classical mean squared loss (MSE) to train the PCritic with Q values as:

$$\begin{aligned} L(\psi) &= \mathbb{E}_{s_t, a_t, r_t \sim E}[(y_t - Q_\psi(s_t, a_t))^2], \\ y_t &= r_t + \gamma_1 Q_{\psi'}(s_{t+1}, \mu'(s_{t+1})). \end{aligned} \tag{11}$$

ψ and ψ' are parameters of the online and target networks. ψ is updated during training, while ψ' is the previous RL parameters in experience pool fixed during optimization. y_t^l is the target Q value learned from the current reward r_t and future Q value in $t+1$ generated by the target network ψ'. $\mu'(s_{t+1})$ indicates the deterministic target policy that provides objective weights. $Q_\psi(s_t, a_t)$ is given by the online network ψ via Eq. (8), which will be trained.

The Q value predicted by PCritic is used to update PActor with policy gradient. Specifically, we maximize the expected return to learn the parameter ϕ in PActor as:

$$L(\phi) = -\mathbb{E}_{a \in \pi_\phi}[\log \pi_\phi(s, a) Q_\psi(s, a)]. \tag{12}$$

π_ϕ is the online policy probability given state and action in Eq. (7), and ϕ is the PActor's parameters to be updated. Finally, the overall RL loss L_{RL} is aggregated as follows:

$$L_{RL} = L(\psi) + \beta L(\phi). \tag{13}$$

β is the weight empirically set as 1. We also add some *Gaussian noises* $\boldsymbol{a}_n \sim N(0, \sigma^2)$ to actions, which bring in additional model explorations to the RL training, improving the generalization and robustness of PAPERec. Note that the Pareto-oriented RL could also be easily adapted to point-wise recommendation by considering rewards in future impressions when predicting.

3.4 Implementation Details on List-wise MOR

In this subsection, we introduce the implementation details of the list-wise single-objective models used in PAPERec. Precisely, we consider two essential objectives widely used in industrial recommendation systems, namely the **Click-Through-Rate (CTR)** and the **dwell time (DT)** [44]. CTR is one of the most important metrics to measure recommendation accuracy in practice. However, simply focusing on CTR will inevitably lead to homogeneity, and may also be easily contaminated by clickbait. In contrast, dwell time, which indicates the time a user spends on an item, is regarded as a more quantitative recommendation accuracy metric. Comparing with CTR, dwell time could better reflect users' real preferences on items to alleviate the influence of clickbait. However, dwell time can be easily affected by the type of items (e.g., videos usually have a much longer dwell time than news) and the length of item contents. Moreover, DT is more difficult to be predicted accurately. In real-world scenarios, we usually jointly consider both CTR and DT to complement each other. However, there are often conflicts between CTR and DT, making it difficult to simultaneously optimize these two metrics. Hence, we propose PAPERec to address this MOR issue.

Specifically, we implement two RL-based list-wise models inspired by [42] as the single-objective models for CTR and DT. For consistency, we directly use the same features and state encoder network introduced in Sec. 3.3. The inputs of both CTR- and DT-oriented models contain user profiles, user historical behaviors, recommendation contexts, and target item profiles. We also adopt the same MultiHead and GRU feature extractors from Eq. (3) to Eq. (6) to build the hidden states $\boldsymbol{h}_t^C$ and $\boldsymbol{h}_t^D$ for CTR and DT. We adopt a *double DQN* [33] to optimize both models in list-wise scenario. Taking CTR for example, we use the MSE loss as the optimization objective similar to Eq. (11). For a user u_j at his t-th position in the impressed list, we formalize the loss $L_t^C(u_j)$ as:

$$L_t^C(u_j) = (r_t^C + \gamma_2 Q'_C(s_{t+1}, a_{t+1}) - Q_C(s_t, a_t))^2. \tag{14}$$

The Q value $Q_C(s_t, a_t)$ in CTR is calculated via the hidden states $\boldsymbol{h}_t^C$ and the t-th target item $\boldsymbol{d}_t$, which is noted as:

$$Q_C(s_t, a_t) = \text{ReLU}(\text{Concat}(\boldsymbol{h}_t^C, \boldsymbol{d}_t)\boldsymbol{w}^C + b^C). \tag{15}$$

The reward r_t^C is 1 if d_t is clicked, and otherwise equals 0. Similarly, we also build the MSE loss for DT as follows:

$$L_t^D(u_j) = (r_t^D + \gamma_2 Q'_D(s_{t+1}, a_{t+1}) - Q_D(s_t, a_t))^2. \tag{16}$$

The Q value $Q_D(s_t, a_t)$ in DT is calculated as Eq. (15):

$$Q_D(s_t, a_t) = \text{ReLU}(\text{Concat}(\boldsymbol{h}_t^D, \boldsymbol{d}_t)\boldsymbol{w}^D + b^D). \tag{17}$$

The reward r_t^D is the discretized dwell time of d_t after normalization. Finally, following the scalarization-based loss $L(\theta)$ in Eq. (1), we aggregate $L_t^C(u_j)$ and $L_t^D(u_j)$ with the personalized weights $\boldsymbol{\omega}_t(u_j)$ given by the Pareto-oriented RL in Eq. (7). We have:

$$\begin{gathered} L_{model} = \sum_{u_j \in U} \sum_{(u_j, d_t) \in I} \boldsymbol{\omega}_{t1}(u_j) L_t^C(u_j) + \boldsymbol{\omega}_{t2}(u_j) L_t^D(u_j). \\ \boldsymbol{\omega}_{t1}(u_j) + \boldsymbol{\omega}_{t2}(u_j) = 1, \ \boldsymbol{\omega}_{ti}(u_j) \geq 0, \ \forall u_j \in U. \end{gathered} \tag{18}$$

$(u_j, d_t) \in I$ indicates that the t-th item d_t has been impressed to user u_j. Different user-item pairs have different objective weights $\boldsymbol{\omega}_{ti}(u_j)$, which could satisfy various user preferences on CTR and DT objectives. The overall loss function L is weighted by the Pareto-oriented RL loss L_{RL} in Eq. (13) and the scalarization loss L_{model} in Eq. (18), which is formalized as follows:

$$L = \lambda L_{RL} + (1 - \lambda) L_{model}. \tag{19}$$

The Pareto-oriented RL module and single-objective models are trained alternately, and λ is a hyper-parameter empirically set as 0.5. Note that other recommendation models such as DeepFM and AFN could be easily adopted as single-objective models in PAPERec. It is convenient to add more objectives in PAPERec such as diversity and novelty. We just need to build an additional single-objective model for each new objective, and modify the output of Pareto-oriented RL to fit the number of objectives.

3.5 Online Deployment

3.5.1 Online System. Pareto-efficient multi-objective recommendation aims to simultaneously improve all objectives, while online experiments can provide convincing evaluations to verify the effectiveness in practice. Hence, we deploy PAPERec on a real-world recommendation system named *WeChat Top Stories*. WeChat Top Stories is an integrated recommendation feed in WeChat, which has billion-level daily interactions on million-level heterogeneous items, including articles, news, and videos. The online recommendation system mainly consists of candidate generation (i.e., matching) and ranking modules as [7]. We have deployed PAPERec on ranking for more than three months. PAPERec can converge smoothly and achieve good performances with different training data.

3.5.2 Online Serving. We consider two representative objectives in online for MOR, namely CTR and item-level dwell time (DT). Here, the item-level DT is calculated by the time users spend on each item. PAPERec takes top 200 items pre-retrieved by the matching module as item candidates, and outputs top 10 items (i.e., the final recommended list) for each user request (e.g., enter or refresh the system). Specifically, PAPERec sequentially recommends items and updates RL states from position 1 to 10 to generate the final list. At each position, for all item candidates, PAPERec first calculates two single-objective scores (e.g., the Q values in Eq. (15) and Eq. (17)) for CTR and DT objectives. Next, the Pareto-oriented RL module generates the objective weights given by PActor in Eq. (7). The final score is the weighted addition of all objective scores as in Eq. (18). PAPERec then sorts all item candidates according to the final weighted objective scores. In this case, the objective weights can be viewed as certain reflections of users' objective-level preferences. We implement PAPERec with Tensorflow, using 50 parameter servers and 100 workers (4-core CPU with 8G memory). We spend nearly 4 hours for daily model updating, which is acceptable for industrial systems. It is convenient to adopt PAPERec in other systems.

4 EXPERIMENTS

We propose PAPERec for list-wise multi-objective recommendation, which is widely applied in real-world scenarios. In this section, we conduct extensive offline and online experiments, aiming to answer the following three research questions: (**RQ1**): How does our proposed PAPERec model perform against the state-of-the-art single-objective models and multi-objective models on all objectives (see Sec. 4.4)? (**RQ2**): How does PAPERec perform in online system with various online multi-aspect evaluation metrics (see Sec. 4.5)? (**RQ3**): What do the personalized objective weights learn and reflect? Are they reasonable (see Sec. 4.6)? We focus on CTR and dwell time (DT) in our offline and online evaluations.

4.1 Dataset

Since there are few large-scale open datasets for list-wise multiple-objective recommendation, we build a new dataset LMOR-1.5B from a real-world recommendation system named WeChat Top Stories widely used by millions of people. Precisely, we randomly collect nearly 145 million impressed lists of 12 million users after data masking to protect user's privacy. Each list contains 10 items, and the overall dataset contains 141 million click and 1.5 billion impression instances. These instances cover 7.2 million items of news, articles and videos. Since PAPERec aims to deal with both CTR and dwell time (DT) objectives, the dwell time (i.e., the time a user spends on a clicked item) is also recorded. We split the dataset into a train set and a test set using the chronological order, getting 1.3 billion impressions in train set and 182 million impressions in test set. Table 1 shows the detailed statistics of LMOR-1.5B.

Table 1: Statistics of the LMOR-1.5B dataset.

#user	#item	#click	#instance
11,942,985	7,196,349	141,387,409	1,452,567,072

4.2 Competitors

We implement several widely-used and competitive recommendation models as baselines to compare with PAPERec.

4.2.1 Single-objective Optimization Methods. We first implement some classical single-objective recommendation models to learn user preferences from a single CTR/DT objective as follows:

- **FM [24].** Factorization machine (FM) is a classical method that models second-order feature interactions with latent vectors. It is widely used in practical systems.
- **NFM [14].** NFM uses a neural FM layer before DNN to capture different feature interactions.
- **DeepFM [13].** DeepFM combines FM with DNN in parallel to model feature interactions.
- **AutoInt [31].** AutoInt introduces self-attentive neural network for better feature interactions.
- **AFN [6].** AFN is a recent SOTA feature interaction model which could learn adaptive-order feature interactions via a logarithmic transformation layer.

We focus on CTR and DT objectives in this work and online system. Therefore, for these single-objective methods, besides their original versions that are trained under the CTR-based cross entropy loss, we also implement their DT versions trained under the DT-oriented objective for a comprehensive comparison. Precisely, we replace the CTR-based cross entropy loss with the similar DT-based MSE loss in Sec. 3.4 for model optimization. We utilize *model*-CTR and *model*-DT to represent the corresponding *model* with CTR and DT oriented objectives respectively.

Fairness of comparisons: It should be emphasized that we use the same features and train set that are used in PAPERec for all baselines. Moreover, for more challenging comparisons, we directly use *the outputs of the same list-wise DQN-based single-objective models in Sec. 3.4* as additional features for all baselines. These additional features provide refined list-wise CTR/DT information generated by the same network structure used in PAPERec. Baselines trained solely under CTR-oriented objectives can achieve better DT performances with the help of such refined DT features, and vice versa. In this case, we force the model comparisons to focus more on the Pareto-efficient learning rather than the list-wise DQN-based network structure of our single-objective models. Although it raises the difficulty of getting significant improvements, it also makes the conclusions much more solid.

4.2.2 Multi-objective Optimization Methods. We also implement several competitive multi-objective optimization methods as the second baseline group for both offline and online evaluations.

- **PO-EA-OE.** PO-EA [26] is a previous state-of-the-art multi-objective Pareto-efficient model for general recommendation, which aims to search the Pareto-efficient solutions to aggregate scores from separately pre-trained and fixed single-objective models. PO-EA conducts an evolutionary algorithm to find the appropriate parameters for weighted aggregation, where the objective weights are shared by all users. Inspired by the trial-and-error procedure in reinforcement learning, we further add an online exploration strategy to optimize our evolutionary algorithm with real-world user feedbacks. This enhanced PO-EA version armed with online exploration is noted as PO-EA-OE in experiments. PO-EA-OE is a strong baseline previously deployed in online.
- **POW-RL.** Personalized objective weighting with RL (POW-RL) is another competitive baseline we propose and deploy in online system. Differing from conventional Pareto-efficient MOR models, it brings in objective-level personalization to single-objective model aggregation. Precisely, it conducts reinforcement learning to generate personalized objective weights using double DQN. The list-wise MOR reward is the weighted aggregation of CTR and DT rewards. Note that the objective weights of rewards are shared and fixed, which are empirically pre-defined by posterior online performances and customized online strategies. POW-RL can be viewed as a semi-objective-personalized model.

PO-EA-OE and POW-RL focus on finding appropriate general or personalized balances between CTR and DT oriented objectives. For fair comparisons, we use the same double DQN model and training strategy in PAPERec to optimize their single-objective models as in Sec. 3.4. Note that we do not compare with some Pareto-efficient models such as PE-LTR [20], since it is difficult to directly adopt those models to list-wise MOR tasks. Moreover, PE-LTR takes a lot of effort on Pareto frontier generation and solution selection specially in E-commerce, which does not fit well for our integrated recommendation feed scenario.

4.2.3 Ablation Settings. We further implement different PAPERec versions as an ablation study. PAPERec(CTR) indicates the PAPERec version that only utilizes the scores of CTR-based single-objective model trained under the original PAPERec framework. Similarly, PAPERec(DT) represents the version only considering the scores of DT-based single-objective model. Moreover, we also add a random ensemble model PAPERec(RD) in online evaluation, which randomly generates objective weights for model aggregation.

4.3 Experimental Settings

PAPERec takes top 200 items in each channel as inputs and output top 10 heterogeneous items. The maximum length of user click sequence is 10 in Pareto-oriented RL module and two single-objective models. The dimension of input feature field embeddings of Transformer in $\boldsymbol{U}_p$ and $\boldsymbol{C}_i$ is 8, while the Transformer is 4-head. The dimension of hidden state $\boldsymbol{h}_t$ is 128, and the dimension of item $\boldsymbol{d}_t$ is 32. The discount factors of Pareto-oriented RL γ_1 and single-objective models γ_2 balance the future rewards in list-wise recommendation. We empirically set $\gamma_1 = 0.1$ and $\gamma_2 = 0.3$ according to the overall performances in online and validation set. In training, both Pareto-oriented RL and single-objective models are optimized as Algorithm 1. We use Adam for optimization with the batch size set as 256. We conduct a grid search for parameter selection. All models share the same experimental settings and features also used in PAPERec for fair comparisons.

4.4 CTR and DT Prediction (RQ1)

The offline evaluation aims to verify the effectiveness of PAPERec in multi-objective recommendation. Precisely, we focus on CTR and DT Prediction with the real-world LMOR-1.5B dataset.

4.4.1 Evaluation Protocol. For CTR prediction, we use three classical metrics for offline evaluation, including hit rate@K (HIT@K), mean average precision (MAP) and area under curve (AUC), which are widely utilized in recommendation tasks [6, 13]. HIT@N measures whether clicked items will be ranked in top N items in the list predicted by models. MAP concerns about the ranks of clicked items. AUC calculates the probability that a random positive example scores higher than a random negative example. In LMOR-1.5B,

Table 2: Results of CTR and DT Prediction. The biggest improvements of PAPERec are significant with $p < 0.01$.

Model	CTR-related metrics					DT-related metrics				
	HIT@1	HIT@3	HIT@5	MAP	AUC	WHIT@1	WHIT@3	WHIT@5	WMAP	AUC
FM-CTR	0.2108	0.5084	0.7123	0.5107	0.7828	0.6905	1.6340	2.2705	1.6291	0.5204
NFM-CTR	0.2505	0.5548	0.7511	0.5537	0.7936	0.8204	1.7824	2.3888	1.7679	0.5244
DeepFM-CTR	0.2559	0.5610	0.7594	0.5603	0.8093	0.8341	1.7969	2.4133	1.7858	0.5155
AutoInt-CTR	0.2573	0.5613	0.7555	0.5605	0.8070	0.8507	1.8155	2.4148	1.8006	0.5254
AFN-CTR	0.2601	0.5695	0.7653	0.5656	0.8127	0.8605	1.8445	2.4489	1.8185	0.5245
FM-DT	0.2070	0.5057	0.7113	0.5076	0.7698	0.6876	1.6406	2.2795	1.6317	0.5762
NFM-DT	0.2494	0.5536	0.7522	0.5531	0.7732	0.8387	1.8100	2.4201	1.7928	0.5763
DeepFM-DT	0.2528	0.5565	0.7539	0.5595	0.7974	0.8604	1.8398	2.4431	1.8192	0.5857
AutoInt-DT	0.2555	0.5582	0.7530	0.5580	0.7980	0.8530	1.8184	2.4170	1.8029	0.5734
AFN-DT	0.2570	0.5640	0.7608	0.5616	0.8037	0.8811	1.8704	2.4621	1.8444	0.5883
PO-EA-OE	0.2532	0.5544	0.7523	0.5594	0.8010	0.8510	1.8159	2.4152	1.8016	0.5535
POW-RL	0.2558	0.5577	0.7540	0.5610	0.8020	0.8704	1.8390	2.4298	1.8364	0.5945
PAPERec(CTR)	**0.2650**	**0.5740**	**0.7678**	**0.5704**	**0.8149**	0.8729	1.8550	2.4530	1.8306	0.5243
PAPERec(DT)	0.2534	0.5557	0.7526	0.5560	0.7930	0.8843	1.8654	2.4568	1.8469	**0.6204**
PAPERec	0.2591	0.5649	0.7606	0.5632	0.8042	**0.8896**	**1.8742**	**2.4655**	**1.8866**	0.5951

all clicked items are viewed as positive examples, while all unclicked items are regarded as negative examples.

For DT prediction, we hope that items with higher dwell time should (i) have higher ranks, and (ii) considered more significant in evaluation. Different from clicks in CTR prediction, dwell time is a continuous value. We suppose that the evaluation metrics of DT prediction should take the specific value of dwell time into consideration. Therefore, we evaluate all models with the enhanced versions of HIT@K, MAP and AUC. Following the similar metrics in [20], we propose WHIT@K and WMAP considering the specific value of dwell time as weights, which are formalized as follows:

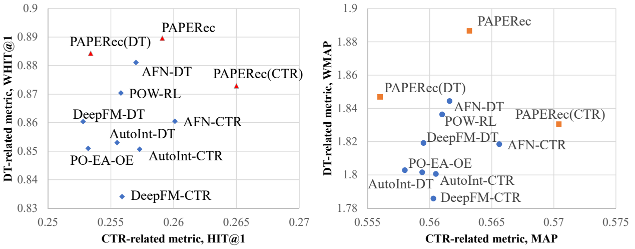
$$\text{WHIT@K} = \frac{1}{N}\sum_{i=1}^{N} dwell\_time_i \times is\_hit(i, K). \tag{20}$$

N indicates the number of clicks, and $dwell\_time_i$ is the DT of the i-th item. $is\_hit(i, K) = 1$ denotes that the i-th item is ranked in top K according to the model's scores, and otherwise equals 0. We also have WMAP enhanced from MAP as follows:

$$\text{WMAP} = \frac{1}{N_l}\sum_{k=1}^{N_l} \text{WAP}_k, \quad \text{WAP}_k = \frac{\sum_{i=1}^{N_r} p@i \cdot dwell\_time_i^k}{C_k}. \tag{21}$$

N_l indicates the number of lists, N_r represents the number of items in a list ($N_r = 10$ in PAPERec), and C_k is the clicked item amount of the k-th list. $p@i$ represents the precision of top i items in the k-th list, and $dwell\_time_i^k$ reflects the dwell time of the i-th item in the k-th list. The AUC of DT prediction also reflects the probability a random higher-DT example scores higher than a random lower-DT example. We conduct a maximum truncation for dwell time to avoid extreme examples. All models follow the same evaluation manner.

4.4.2 Experimental Results. Table 2 shows the results of ten CTR-related and DT-related metrics on list-wise multi-objective recommendation, from which we can observe that:



(a) HIT@1 of CTR / WHIT@1 of DT (b) MAP of CTR / WMAP of DT

Figure 3: Visualization of PAPERec and different baselines.

(1) PAPERec models achieve the best performances on all CTR and DT related metrics. Considering the overall MOR performance, PAPERec models significantly outperform all single-objective optimization and multi-objective optimization methods with the significance level $p < 0.01$. It verifies the effectiveness of the Pareto-oriented RL. In PAPERec, Pareto-oriented RL guides both CTR and DT based single-objective models to reach the approximate Pareto efficient situations with personalized objective weights. In this case, both single-objective models can be trained more sufficiently and wisely to handle with the general characters and conflicts in MOR. Fig. 3 visualizes the CTR and DT related performances of different models, which also verify the advantages of PAPERec intuitively.

(2) Comparing among different PAPERec versions, we find that PAPERec achieves the best overall performance (PAPERec has the best WHIT@N and WMAP of DT and the second-best results for the rest CTR and DT metrics). The advantages of PAPERec over single CTR/DT model versions mainly come from the personalized objective weights via Pareto-efficient training, which can benefit online single-objective model fusion. The advantages of personalized objective weights are also reflected in online evaluation metrics of

CTR, DT and diversity, which will be discussed in Sec. 4.5. We will also give a detailed analysis of the objective-level personalization in Sec. 4.6 from multiple user, item and type aspects. Moreover, we also conduct PAPERec with different discount factors in RL modules, and find that PAPERec with $\gamma_1 = 0.1$ and $\gamma_2 = 0.3$ achieves the best performance. It further confirms the effectiveness of RL-based modeling and future rewards in list-wise MOR.

(3) PAPERec even surprisingly outperforms PAPERec(DT) in DT prediction. Note that dwell time can be viewed as a quantitative CTR metric that is extremely challenging to predict, for the reward of DT is continuous and the DT prediction is much harder than binary click prediction. The improvements of PAPERec in DT may owe to the smart fusion of CTR and DT objectives, since CTR-oriented models can provide a coarse-grained judgment, which might be beneficial for DT prediction.

(4) In baselines, AFN models achieve relatively good overall performances. Nevertheless, PAPERec still outperforms AFN in overall. Multi-objective models such as PO-EA-OE and POW-RL also perform well in balancing CTR and DT. These models are trained under both CTR and DT supervised information and have improvements. However, they still perform worse than PAPERec, which shows that the personalized objective weights of PAPERec are more effective. Simple joint optimization may not improve the overall performance. We should highlight that all single-objective optimization baselines are enhanced with the list-wise single-objective DQN models used in PAPERec (see Sec. 4.2). These challenging comparisons further confirm the effectiveness and robustness of PAPERec in MOR.

4.5 Online Evaluation (RQ2)

The CTR and DT predictions confirm the effectiveness of PAPERec with offline CTR and DT related metrics. However, the chain chemical reaction of multi-objective recommendation should be comprehensively exposed via online evaluation on real-world systems. Hence, we conduct an online evaluation for PAPERec.

4.5.1 Evaluation Protocol. We have deployed PAPERec on a real-world MOR system in WeChat Top Stories as introduced in Sec. 3.5. Precisely, we implement different PAPERec versions and multi-objective methods, which are deployed in ranking with other online modules unchanged. The online base model is a weighted combination with two double DQN based single-objective CTR/DT models, where the objective weights are empirically set and fixed according to previous overall online performances.

For comprehensive comparisons, we concentrate on five online metrics to evaluate models from multiple aspects including CTR, DT and diversity. We have: (a) CTR, which is a classical ranking metric that measures online click-oriented performance. (b) Has-click rate (HCR), which indicates the proportion of users who have clicked any items of the day. (c) Dwell time in system per capita (DT-s), which represents the average time users spend on the online system (including both main feed and item's content page). (d) Dwell time on item per capita (DT-i), which represents the average valid time users spend on items' content pages. (e) Clicked tag number per capita (CTN), which is a diversity metric that is calculated by the number of clicked deduplicated tags in items. We conduct the online evaluation for 7 days with more than 6 million users involved, and report the improvement percentages instead of the specific values.

Table 3: Online evaluation on a real-world system with multiple CTR, DT and diversity related metrics.

Metrics	CTR	HCR	DT-s	DT-i	CTN
PO-EA-OE	+0.14%	-0.05%	+0.41%	+0.88%	+1.10%
POW-RL	+0.58%	-0.01%	+0.85%	+2.11%	+5.39%
PAPERec(RD)	+1.05%	+0.05%	+0.59%	+2.33%	+15.60%
PAPERec(CTR)	**+1.63%**	+0.15%	+0.95%	+2.97%	+13.28%
PAPERec(DT)	+1.32%	+0.11%	+0.86%	+3.00%	**+15.76%**
PAPERec	+1.11%	**+0.23%**	**+1.33%**	**+3.26%**	+15.34%

4.5.2 Experimental Results. Table 3 shows the online evaluation results with multiple metrics. We can find that:

(1) All PAPERec models have consistent improvements over the base model, which achieve the best performances on all types of online metrics. PAPERec achieves the best performances on three metrics including both click and dwell time related metrics, with the significance level $p < 0.05$ for HCR and $p < 0.01$ for DT-s and DT-i. The simultaneous improvement on both CTR (+1.11%) and DT-i (+3.26%) are rare and impressive in online evaluation. It verifies that PAPERec can improve multiple CTR and DT objectives simultaneously in online, and reconfirms the significance of Pareto-oriented RL to model optimization and online fusion.

(2) PAPERec(CTR) achieves the best CTR result. However, compared with other PAPERec versions, PAPERec(CTR) has the worst diversity result in CTN. It implies that too many concentrations on CTR-oriented objectives may inevitably result in homogeneity. Nevertheless, it still outperforms other baselines on CTN, which may benefit from the sufficient Pareto-oriented training. The improvements of PAPERec over its random ensemble version (RD) also proves that the objective weights are essential and sensitive.

(3) Other multi-objective baselines like PO-EA-OE and POW-RL also have improvements on several CTR and DT metrics. It reconfirms the importance of multi-objective fusion in online systems. POW-RL has better performance due to the personalized objective weights learned from RL-based fusion module. However, PAPERec still outperforms these models by a large margin, since its personalized objective weights learned from Pareto-oriented RL can help both single-objective model training and model fusion.

(4) We also find that PAPERec has 0.24% improvements on user stickiness. User stickiness is the core indicator of real-world systems, which is calculated by the proportion of yesterday's users who also utilize our system today. It is extremely difficult to have user stickiness improved by simple model optimization. The user stickiness improvement verifies the online effectiveness of PAPERec indirectly from another aspect.

4.6 Analysis on Objective-level Personalization (RQ3)

In this subsection, we aim to explore what the personalized objective weights have learned and implied, and whether the Pareto-based objective-level personalization is reasonable. We analyze the objective weights at user, item and item type levels.

4.6.1 Evaluation Protocol. The personalized objective weights generated by Pareto-oriented RL are influenced by both users and items. Therefore, we want to know what characters of users and items will affect objective-level personalization and whether they are reasonable. Specifically, for user aspect, we first cluster all impression instances in the test set by users. Next, we rank all users by their average objective weights ω_{CTR} and ω_{DT} generated by Pareto-oriented RL. Third, we select users with the top 10% CTR and DT weights, regarded as the user groups with CTR/DT objective preferences respectively predicted by PAPERec. Similarly, we also build the corresponding item groups with CTR/DT preferences, and filter out users and items that are below certain click thresholds to alleviate bias. Finally, we calculate the average CTR of user groups and the average dwell time per click (DT/c) of item groups to show the validity and effectiveness of our personalized objective weights. Moreover, LMOR-1.5B is an integrated recommendation dataset containing various types of items including video, article and news, for which users may have inherent objective-level preferences. Hence, we also display the proportions of different item types for items with polarized CTR/DT objective weights.

Table 4: Results of CTR and dwell time per click for users and items with polarized CTR/DT objective preferences.

Aspect	Metric	Top 10% CTR weight	Top 10% DT weight
user	CTR	+33.8%	-36.3%
item	DT/c	-6.4%	+3.8%

Table 5: Results of the proportions of different item types for items with polarized CTR/DT objective preferences.

Item type	video	article	news
Top 10% CTR weight	15.4%	71.9%	13.7%
Top 10% DT weight	21.2%	66.2%	12.6%

4.6.2 *Experimental Results.*

Table 4 shows the relative values of CTR and DT/c compared with the average results of all users and items, and Table 5 displays the proportions of video/article/news with different CTR/DT weights, from which we can find that:

(1) In user aspect, the CTR of users having top 10% CTR weight is 33.8% higher than the average CTR of all users, while the users with top 10% DT weight have 36.3% lower CTR. The CTR deviation is reasonable. CTR is calculated by dividing clicks by impressions. A user having a higher CTR indicates that the user is more willing to click items under the same amount of impressions. This type of user should care more about CTR-related objectives. The results imply that PAPERec has successfully found out users who care more about clicks, and gives those users higher CTR objective weights for better objective-level personalization. A similar CTR deviation could also be found in the item groups.

(2) In item aspect, the DT/c of items having top 10% CTR weight is 6.4% lower than the average number, while that having top 10% DT weight is 3.8% higher. DT/c indicates the dwell time per click. An item with a higher DT/c represents that users will spend more time on reading its contents per click, where DT should be more considered. Hence, the DT/c deviation is also reasonable, indicating the successful objective-level personalization at the item level.

(3) From Table 5 we can observe that: videos usually have larger DT weights and focus more on dwell time, while article and news have larger CTR weights. These results are in line with the actual demands in integrated recommendation systems, since video recommendation should pay more attention to DT metrics that can truly reflect user's satisfaction. It reconfirms the effectiveness of PAPERec in learning reasonable objective weights for objective-level personalization at the item type level.

5 CONCLUSION AND FUTURE WORK

In this work, we propose a new Personalized Approximate Pareto-Efficient Recommendation (PAPERec) framework to simultaneously improve all objectives in multi-objective recommendation. We conduct a Pareto-oriented RL to generate the personalized objective weights in scalarization, which can help single-objective models to approximately optimize toward Pareto-efficient situations. The proposed objective-level personalization is beneficial for both model optimization and online fusion. In experiments, PAPERec achieves significant improvements on both offline and online evaluations with CTR/DT related metrics. Analyses on objective-level personalization also verify its effectiveness. Currently, PAPERec has been deployed on WeChat Top Stories, affecting millions of users.

In the future, we will conduct further discussions on the Pareto frontier of PAPERec to flexibly generate different overall objective preferences for different online scenarios. We will also explore and add the prior knowledge of user objective-level preferences to our Pareto-oriented RL module, along with more sophisticated neural structures. Finally, we will also attempt to complete the theoretical part of the personalized approximate Pareto-efficient learning in order to inspire and facilitate the model design.

REFERENCES

[1] Mohammad Mahdi Azadjalal, Parham Moradi, Alireza Abdollahpouri, and Mahdi Jalili. 2017. A trust-aware recommendation method based on Pareto dominance and confidence concepts. *Knowledge-Based Systems* (2017).

[2] Abhijnan Chakraborty, Saptarshi Ghosh, Niloy Ganguly, and Krishna P Gummadi. 2017. Optimizing the recency-relevancy trade-off in online news recommendations. In *Proceedings of WWW*.

[3] Minmin Chen, Alex Beutel, Paul Covington, Sagar Jain, Francois Belletti, and Ed H Chi. 2019. Top-k off-policy correction for a REINFORCE recommender system. In *Proceedings of WSDM*.

[4] Xinshi Chen, Shuang Li, Hui Li, Shaohua Jiang, Yuan Qi, and Le Song. 2019. Generative Adversarial User Model for Reinforcement Learning Based Recommendation System. In *Proceedings of ICML*.

[5] Heng-Tze Cheng, Levent Koc, Jeremiah Harmsen, Tal Shaked, Tushar Chandra, Hrishi Aradhye, Glen Anderson, Greg Corrado, Wei Chai, Mustafa Ispir, et al. 2016. Wide & deep learning for recommender systems. In *Proceedings of the DLRS workshop*.

[6] Weiyu Cheng, Yanyan Shen, and Linpeng Huang. 2020. Adaptive Factorization Network: Learning Adaptive-Order Feature Interactions. In *Proceedings of AAAI*.

[7] Paul Covington, Jay Adams, and Emre Sargin. 2016. Deep neural networks for youtube recommendations. In *Proceedings of RecSys*.

[8] Laizhong Cui, Peng Ou, Xianghua Fu, Zhenkun Wen, and Nan Lu. 2017. A novel multi-objective evolutionary algorithm for recommendation systems. *J. Parallel and Distrib. Comput.* (2017).

[9] Tomaž Curk et al. 2020. Relation chaining in binary positive-only recommender systems. *Expert Systems with Applications* (2020).

[10] Jean-Antoine Désidéri. 2012. Multiple-gradient descent algorithm (MGDA) for multiobjective optimization. *Comptes Rendus Mathematique* (2012).

[11] Yufei Feng, Fuyu Lv, Weichen Shen, Menghan Wang, Fei Sun, Yu Zhu, and Keping Yang. 2019. Deep session interest network for click-through rate prediction. In *Proceedings of IJCAI*.

[12] Scott Fujimoto, David Meger, and Doina Precup. 2019. Off-policy deep reinforcement learning without exploration. In *Proceedings of ICML*.

[13] Huifeng Guo, Ruiming Tang, Yunming Ye, Zhenguo Li, and Xiuqiang He. 2017. DeepFM: a factorization-machine based neural network for CTR prediction. In *Proceedings of IJCAI*.

[14] Xiangnan He and Tat-Seng Chua. 2017. Neural factorization machines for sparse predictive analytics. In *Proceedings of SIGIR*.

[15] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. 2016. Session-based recommendations with recurrent neural networks. In *Proceedings of ICLR*.

[16] Eugene Ie, Vihan Jain, Jing Wang, Sanmit Navrekar, Ritesh Agarwal, Rui Wu, Heng-Tze Cheng, Morgane Lustman, Vince Gatto, Paul Covington, et al. 2019. Reinforcement learning for slate-based recommender systems: A tractable decomposition and practical methodology. In *Proceedings of IJCAI*.

[17] Tamas Jambor and Jun Wang. 2010. Optimizing multiple objectives in collaborative filtering. In *Proceedings of RecSys*.

[18] Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. Matrix factorization techniques for recommender systems. *Computer* (2009).

[19] Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. 2016. Continuous control with deep reinforcement learning. In *Proceedings of ICLR*.

[20] Xiao Lin, Hongjie Chen, Changhua Pei, Fei Sun, Xuanji Xiao, Hanxiao Sun, Yongfeng Zhang, Wenwu Ou, and Peng Jiang. 2019. A pareto-efficient algorithm for multiple objective optimization in e-commerce recommendation. In *Proceedings of RecSys*.

[21] Feng Liu, Huifeng Guo, Xutao Li, Ruiming Tang, Yunming Ye, and Xiuqiang He. 2020. End-to-End Deep Reinforcement Learning based Recommendation with Supervised Embedding. In *Proceedings of WSDM*.

[22] Qi Liu, Ruobing Xie, Lei Chen, Shukai Liu, Ke Tu, Peng Cui, Bo Zhang, and Leyu Lin. 2020. Graph Neural Network for Tag Ranking in Tag-enhanced Video Recommendation. In *Proceedings of CIKM*.

[23] Yong Liu, Yinan Zhang, Qiong Wu, Chunyan Miao, Lizhen Cui, Binqiang Zhao, Yin Zhao, and Lu Guan. 2019. Diversity-Promoting Deep Reinforcement Learning for Interactive Recommendation. *arXiv preprint arXiv:1903.07826* (2019).

[24] Steffen Rendle. 2010. Factorization machines. In *Proceedings of ICDM*.

[25] Marco Tulio Ribeiro, Anisio Lacerda, Adriano Veloso, and Nivio Ziviani. 2012. Pareto-efficient hybridization for multi-objective recommender systems. In *Proceedings of RecSys*.

[26] Marco Tulio Ribeiro, Nivio Ziviani, Edleno Silva De Moura, Itamar Hata, Anisio Lacerda, and Adriano Veloso. 2014. Multiobjective pareto-efficient approaches for recommender systems. *TIST* (2014).

[27] Mario Rodriguez, Christian Posse, and Ethan Zhang. 2012. Multiple objective optimization in recommender systems. In *Proceedings of RecSys*.

[28] Badrul Munir Sarwar, George Karypis, Joseph A Konstan, John Riedl, et al. 2001. Item-based collaborative filtering recommendation algorithms.. In *Proceedings of WWW*.

[29] Ozan Sener and Vladlen Koltun. 2018. Multi-task learning as multi-objective optimization. In *Proceedings of NIPS*.

[30] Ying Shan, T Ryan Hoens, Jian Jiao, Haijing Wang, Dong Yu, and JC Mao. 2016. Deep crossing: Web-scale modeling without manually crafted combinatorial features. In *Proceedings of KDD*.

[31] Weiping Song, Chence Shi, Zhiping Xiao, Zhijian Duan, Yewen Xu, Ming Zhang, and Jian Tang. 2019. Autoint: Automatic feature interaction learning via self-attentive neural networks. In *Proceedings of CIKM*.

[32] Fei Sun, Jun Liu, Jian Wu, Changhua Pei, Xiao Lin, Wenwu Ou, and Peng Jiang. 2019. BERT4Rec: Sequential Recommendation with Bidirectional Encoder Representations from Transformer. In *Proceedings of CIKM*.

[33] Hado Van Hasselt, Arthur Guez, and David Silver. 2016. Deep reinforcement learning with double q-learning. In *Proceedings of AAAI*.

[34] Lu Wang, Wei Zhang, Xiaofeng He, and Hongyuan Zha. 2018. Supervised reinforcement learning with recurrent neural network for dynamic treatment recommendation. In *Proceedings of KDD*.

[35] Shanfeng Wang, Maoguo Gong, Haoliang Li, and Junwei Yang. 2016. Multi-objective optimization for long tail recommendation. *Knowledge-Based Systems* (2016).

[36] Qiong Wu, Yong Liu, Chunyan Miao, Binqiang Zhao, Yin Zhao, and Lu Guan. 2019. PD-GAN: Adversarial Learning for Personalized Diversity-Promoting Recommendation. In *Proceedings of IJCAI*.

[37] Shu Wu, Yuyuan Tang, Yanqiao Zhu, Liang Wang, Xing Xie, and Tieniu Tan. 2019. Session-based Recommendation with Graph Neural Networks. In *Proceedings of AAAI*.

[38] Jun Xiao, Hao Ye, Xiangnan He, Hanwang Zhang, Fei Wu, and Tat-Seng Chua. 2017. Attentional factorization machines: Learning the weight of feature interactions via attention networks. In *Proceedings of IJCAI*.

[39] Lin Xiao, Zhang Min, Zhang Yongfeng, Gu Zhaoquan, Liu Yiqun, and Ma Shaoping. 2017. Fairness-aware group recommendation with pareto-efficiency. In *Proceedings of RecSys*.

[40] Ruobing Xie, Cheng Ling, Yalong Wang, Rui Wang, Feng Xia, and Leyu Lin. 2020. Deep Feedback Network for Recommendation. In *Proceedings of IJCAI*.

[41] Ruobing Xie, Zhijie Qiu, Jun Rao, Yi Liu, Bo Zhang, and Leyu Lin. 2020. Internal and Contextual Attention Network for Cold-start Multi-channel Matching in Recommendation. In *Proceedings of IJCAI*.

[42] Ruobing Xie, Shaoliang Zhang, Rui Wang, Feng Xia, and Leyu Lin. 2021. Hierarchical Reinforcement Learning for Integrated Recommendation. In *Proceedings of AAAI*.

[43] Xin Xin, Alexandros Karatzoglou, Ioannis Arapakis, and Joemon M Jose. 2020. Self-Supervised Reinforcement Learning for Recommender Systems. In *Proceedings of SIGIR*.

[44] Xing Yi, Liangjie Hong, Erheng Zhong, Nanthan Nan Liu, and Suju Rajan. 2014. Beyond clicks: dwell time for personalization. In *Proceedings of RecSys*.

[45] Jing Zhang, Bowen Hao, Bo Chen, Cuiping Li, Hong Chen, and Jimeng Sund. 2019. Hierarchical Reinforcement Learning for Course Recommendation in MOOCs. In *Proceedings of AAAI*.

[46] Weinan Zhang, Tianming Du, and Jun Wang. 2016. Deep learning over multi-field categorical data. In *European conference on information retrieval*.

[47] Dongyang Zhao, Liang Zhang, Bo Zhang, Lizhou Zheng, Yongjun Bao, and Weipeng Yan. 2019. Deep Hierarchical Reinforcement Learning Based Recommendations via Multi-goals Abstraction. In *Proceedings of KDD*.

[48] Xiangyu Zhao, Long Xia, Liang Zhang, Zhuoye Ding, Dawei Yin, and Jiliang Tang. 2018. Deep reinforcement learning for page-wise recommendations. In *Proceedings of RecSys*.

[49] Xiangyu Zhao, Liang Zhang, Zhuoye Ding, Long Xia, Jiliang Tang, and Dawei Yin. 2018. Recommendations with negative feedback via pairwise deep reinforcement learning. In *Proceedings of KDD*.

[50] Guanjie Zheng, Fuzheng Zhang, Zihan Zheng, Yang Xiang, Nicholas Jing Yuan, Xing Xie, and Zhenhui Li. 2018. DRN: A deep reinforcement learning framework for news recommendation. In *Proceedings of WWW*.

[51] Lixin Zou, Long Xia, Zhuoye Ding, Dawei Yin, Jiaxing Song, and Weidong Liu. 2019. Reinforcement Learning to Diversify Top-N Recommendation. In *International Conference on Database Systems for Advanced Applications*.