# Pantheon: Personalized Multi-objective Ensemble Sort via Iterative Pareto Policy Optimization

Jiangxia Cao, Pengbo Xu, Yin Cheng, Kaiwei Guo, Jian Tang, Shijun Wang, Dewei Leng, Shuang Yang, Zhaojie Liu, Yanan Niu, Guorui Zhou, Kun Gai\* Kuaishou Technology, Beijing, China

{caojiangxia, xupengbo03}@kuaishou.com, yin.sjtu@gmail.com, {guokaiwei, tangjian03, wangshijun03, lengdewei, yangshuang08, zhaotianxing, niuyanan, zhouguorui}@kuaishou.com, kun.gai@qq.com

#### ABSTRACT

RecSys engines have significantly advanced our daily-life, such as Kuaishou for short-video/live-streaming, Taobao for onlineshopping, and so on. To provide promising recommendation results, there exist three major stages in the industrial RecSys chain to support our service: (1) The first Retrieval model aims at searching hundreds of item candidates. (2) Next, the Ranking model estimates the multiple aspect probabilities Pxtrs for each retrieved item. (3) At last, the Ensemble Sort stage merges those Pxtrs into one comparable score, and then selects the best dozen items with the highest scores to recommend them. To our knowledge, the wideaccepted industry ensemble sort approach still relies on manual formula-based adjustment, i.e., assigning manual weights for Pxtrs to control its influence to generate the fusion score. Under this framework, the RecSys severely relies on expert knowledge to determine satisfactory weight for each Pxtr, which blocks our system further advancements.

In this paper, we provide our milestone ensemble sort work and the first-hand practical experience, Pantheon, which transforms ensemble sorting from a "human-curated art" to a "machineoptimized science". Compared with formulation-based ensemble sort, our Pantheon has the following advantages: (1) Personalized Joint Training: our Pantheon is jointly trained with the real-time ranking model, which could capture ever-changing user personalized interests accurately. (2) Representation inheritance: instead of the highly compressed Pxtrs, our Pantheon utilizes the finegrained hidden-states as model input, which could benefit from the Ranking model to enhance our model complexity. Meanwhile, to reach a balanced multi-objective ensemble sort, we further devise an iterative Pareto policy optimization (IPPO) strategy to consider the multiple objectives at the same time. To our knowledge, this paper is the first work to replace the entire formulation-based ensemble sort in industry RecSys, which was fully deployed at Kuaishou live-streaming services, serving 400 Million users daily.

## **CCS CONCEPTS**

• Information systems  $\rightarrow$  Recommender systems.

# **KEYWORDS**

Multi-Objective Optimization; Reinforcement Learning;

#### **ACM Reference Format:**

Jiangxia Cao, Pengbo Xu, Yin Cheng, Kaiwei Guo, Jian Tang, Shijun Wang,, Dewei Leng, Shuang Yang, Zhaojie Liu, Yanan Niu, Guorui Zhou, Kun Gai\*. 2025. Pantheon: Personalized Multi-objective Ensemble Sort via Iterative

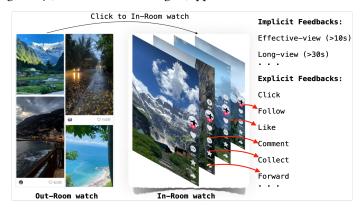


Figure 1: Multiple objectives at Kuaishou.

## 1 INTRODUCTION

"A system is enjoying maximum economic satisfaction when no one can be made better off without making someone else worse off."

––Vilfredo Pareto

Kuaishou, as a leading short-video and live-streaming sharing platform, building a booming environment which attracts and benefits millions of users and creators worldwide. As shown in Figure 1, our Kuaishou offers a highly immersive experience through auto-play short-videos/live-streaming in full-screen mode by simply swiping Out-Room watching or clicking to In-Room watching, and leaving some interactions during surfing our platform [6], e.g., click, long-view, comment, etc. To optimize user experience, our platform relies heavily on the recommendation system (RecSys) to ensure relevant content reaches the right users [19]. However, with tens of millions of short-videos and live-streamings uploaded daily by creators, performing real-time scoring for every user-item pair across the entire space is computationally impossible [7]. To make a trade-off between effectiveness and efficiency, the industrial RecSys usually follows an elaborate cascading chain design paradigm with three major stages to respond user's recommendation request [8, 15].

 User/Item information disentangled Retrieval Model [12, 16, 20]: as shown in Figure 2(a), the retrieval model always follow User/Item information disentanglement two-tower framework.

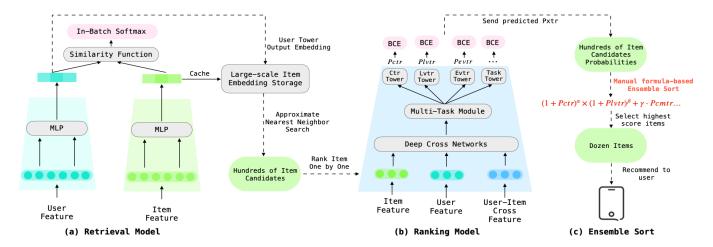


Figure 2: Illustration of RecSys chain: (a) the two-tower retrieval model searches hundreds of item candidates; (b) the ranking model to estimate the multiple objective probabilities; (c) the ensemble sort to fuse those prediction values as one score to select the top items.

Therefore, we could easily find a small group of hundreds of item candidates from billion-scale item pool relies users' profile only.

- Multiple task learning guided Ranking Model [27, 31, 33]: as shown in Figure 2(b), the hybrid ranking model always mixes the user, item and user-item-cross information jointly to estimate multiple aspect interaction probabilities Pxtrs for retrieved items one by one, e.g., ctr of click rate, lvtr of long-view rate.
- Multiple objective optimizing oriented Ensemble Sort [30]: as shown in Figure 2(c), the ensemble sort aims at fuse the predicted Pxtrs into a single comparable score. Thus we could select the best dozen items with the highest scores to recommend them.

As the final stage in the recommendation pipeline, the Ensemble Sort [34] critically shapes both user experience and platform ecology to decide which content will be distributed to our users. Therefore, the ensemble sort stage should be handled carefully handling, to find an optimal balance across multiple objective [11, 13, 32], such as total watch time, total clicks, and other metrics. To the best of our knowledge, although there exists elaborate efforts [2, 5, 21], but most industrial RecSys still employ formula-based ensemble sorting mechanisms to measure final fused scores - typically using hybrid multiplicative or additive scoring formulas (in Figure 2(c)). It is worth noting that there exist some weights (e.g.,  $\alpha$ ,  $\beta$ ,  $\gamma$ ) to control the importance of corresponding Pxtr in the fused score. To find a group of suitable weights to maximize multiple objectives, the RecSys engineers could leverage offline parameter tuning tools [3] (non-personalization, simple models with small data volume) with expert expertise to identify them. However, in this setting, the highly flexible ("freestyle") nature of the formulation inherently caps its performance potential. It often demands excessively complex rule engineering to reach a Pareto-optimal trade-off among user experience metrics - where no metric can be improved without degrading another.

At Kuaishou, our former ensemble sort mechanism is also equipped by a complex formulation, which has been iterated for several years. While delivering significant initial benefits, the limitations of the pre-defined style formula have become increasingly apparent over time. Meanwhile, iteration requires a heavy reliance on manual expertise and extensive A/B testing, resulting in deterioration in efficiency. Such "hand-crafted" ensemble sort has become a bottleneck hindering further advancements in our system, preventing the full potential of massive data [26] and complex models scaling [1]. To this end, we started to explore the neural-network based ensemble sort - towards to find a new way to unleash it from a "human-curated art" to a "machine-optimized science". Considering that all Pxtrs metrics are generated from the ranking model (in Figure 2), could we develop a 'plugin' module within the ranking model to fuse them?

Motivates by this, we propose our first-hand practical milestone work, **Pantheon**, which **successfully replace the traditional formulation-style ensemble sort mechanism in our system**. In our model architecture designing, our Pantheon has the following advantages compared with formulation-based ensemble sort:

- Highly personalized joint training: In the fusion process, our Pantheon jointly trains with the real-time ranking model [4] and assigns un-shared learnable user and item features as additional information to generate the ensemble score, achieving granular personalization through adaptive ensemble.
- Fine-grained representation inheritance: Actually, propagating these few numerical scores Pxtr leads to severe information decay. Here we reuse the high-dimensional task-specific representations from the task tower as our model input [31], which could benefit our ensemble sort from the computationally complex Ranking model, to enhance our model effectiveness.

In the label side, we follow the standard simple additive weighting mechanism to combine the multiple objectives of user experience, e.g.,  $\sum_{Pxtr}^{\{Pctr,Plvtr,...\}} w^{Pxtr} \mathcal{L}^{Pxtr}$ . Supervised by the additive function, our model is encouraged to converge to a locally optimal point theoretically, while the quality of the 'locally convergence' critically depends on the weighting coefficients  $w^{Pxtr}$  of these objectives. To guarantee convergence to Pareto-optimal states [9, 14, 28], we

design an Iterative Pareto Policy Optimization (IPPO) mechanism from a reinforcement learning (RL) perspective [23], to automatically search a group of weights to reach a Pareto frontier.

In summary, our contributions are as follows:

- We propose our neural-network based ensemble sort approach, to our knowledge, Pantheon is the first work to replace the entire formulation-based ensemble sort in industry RecSys.
- We offer insights into both model architecture and training policy, addressing practical challenges encountered in industrial RecSys, which will shed light on other researchers to explore a more robust ensemble sort.
- We conduct extensive offline and online experiments to verify our Pantheon effectiveness, which contributes more than 1% of Clicked User online gains to Kuaishou live-streaming services.

## 2 PRELIMINARY

In this section, we briefly review: (1) The multi-task learning based Ranking model in RecSys to produce Pxtrs; (2) the traditional Pxtr based ensemble sort for multi-objective optimization [10, 18] in RecSys.

# 2.1 Multi-Task Learning in RecSys

In practice, the multi-task learning always conducted at ranking model, which includes the following four major components:

- Training Label: During watching live-streamings, users always leave a large amount of interaction logs, such as click, long-view, effective-view, and others. Thus, each (user, item) view will generate multiple ground-truth signals, e.g., click  $y^{\text{ctr}} \in \{0, 1\}$ , long-view  $y^{\text{lvtr}} \in \{0, 1\}$ , effective-view  $y^{\text{evtr}} \in \{0, 1\}$ , . . . .
- Input Feature: Generally speaking, the hybrid ranking model integrates user, item, and user-item cross-features, which can be roughly divided into four categories: (1) ID features: user ID, item ID, category ID, etc, (2) Statistical features: user watched live-streaming in the last month, live-streamingmetrics [24] total watched times, to describe user activity levels or item popularity. (3) Sequence features: user recent or searched live-streaming sequences, such as DIN [35], SIM [29]. (4) Multi-modal features: LLM-generated item content embedding and Semantic ID, such as LARM [22], QARM [25].
- Model Architecture: For brevity, let denote the above input features as v, the multi-task learning can be formed as:

$$\begin{split} & e^{\text{ctr}}, e^{\text{lvtr}}, \dots = \text{Mixture-of-Expert}(\mathbf{v}), \\ & t^{\text{ctr}} = \text{Tower}^{\text{ctr}}(e^{\text{ctr}}), \quad t^{\text{lvtr}} = \text{Tower}^{\text{lvtr}}(e^{\text{lvtr}}), \quad \dots, \\ & \hat{y}^{\text{ctr}} = \text{Pred}^{\text{ctr}}(t^{\text{ctr}}), \quad \hat{y}^{\text{lvtr}} = \text{Pred}^{\text{lvtr}}(t^{\text{lvtr}}), \quad \dots, \end{split} \tag{1}$$

where the  $\mathbf{t}^{\mathsf{ctr}}, \mathbf{t}^{\mathsf{lvtr}} \in \mathbb{R}^d$  are the tasks' hidden-states, and the  $\hat{y}^{\mathsf{ctr}}, \hat{y}^{\mathsf{lvtr}} \in (0,1)$  are the predicted scores. The  $\mathsf{Pred}(\cdot)$  are single-layer MLP with Sigmoid activated function, the  $\mathsf{Tower}(\cdot)$  are stacked MLP with  $\mathsf{ReLU}$  activated function, and the multitask module  $\mathsf{Mixture-of-Expert}(\cdot)$  is a gate-expert paradigm networks, such as the MMoE [27], PLE [31] or HoME [33]. The  $\hat{y}^{\mathsf{ctr}} \in (0,1)$  and  $\hat{y}^{\mathsf{lvtr}} \in (0,1)$  are the predicted probabilities, i.e., Pxtr.

• Loss Function: According to the predicted Pxtr  $\{\hat{y}^{\text{ctr}}, \hat{y}^{\text{lvtr}}, \ldots\}$  and the ground-truth label  $\{y^{\text{ctr}}, y^{\text{lvtr}}, \ldots\}$ , the ranking model

conduct multiple binary cross-entropy classification loss to optimize different  $\mbox{{\tt Pxtr}}.$ 

$$\mathcal{L}_{\text{ranking}}^{\text{ctr}} = -\left(y^{\text{ctr}}\log\left(\hat{y}^{\text{ctr}}\right) - (1 - y^{\text{ctr}})\log\left(1 - \hat{y}^{\text{ctr}}\right)\right),$$

$$\mathcal{L}_{\text{ranking}}^{\text{lvtr}} = -\left(y^{\text{lvtr}}\log\left(\hat{y}^{\text{lvtr}}\right) - (1 - y^{\text{lvtr}})\log\left(1 - \hat{y}^{\text{lvtr}}\right)\right), \quad (2)$$

$$\mathcal{L}_{\text{ranking}} = \mathcal{L}_{\text{ranking}}^{\text{ctr}} + \mathcal{L}_{\text{ranking}}^{\text{lvtr}} + \dots$$

where the  $\mathcal{L}_{ranking}$  is the final training loss. Since the different objectives assigned different prediction tower, the ranking model could achieve the maximizing accuracy for each objective.

# 2.2 Multi-Objective Optimization in RecSys

After the ranking model, the multi-objective optimization [17] aims at selecting the best items that have the highest score for user experience. Here we review an **offline** parameter searching workflow for the formula-based ensemble sort:

- **Offline Data Collection**: In an offline setting, we first collect millions of data samples, while each samples records the ranking model predictions results Pxtrs and the ground-truth labels i.e.,  $y^{\text{ctr}} \in \{0, 1\}$ , long-view  $y^{\text{lvtr}} \in \{0, 1\}$ , and so on.
- Pre-Defined Formulation Style: Based on these samples, we then copy the pre-defined online base ensemble sort fusion formula style. Here we provide a toy example:

Score = 
$$(1 + \hat{y}^{\text{ctr}})^{\alpha} \times (1 + \hat{y}^{\text{lvtr}})^{\beta} + \gamma \cdot \hat{y}^{\text{evtr}} + \dots$$
 (3)

where the Score is the fusion score of each sample, and  $\alpha$ ,  $\beta$ ,  $\gamma$  are formula parameters.

Hand-Crafted Evaluation Metric: During parameter searching, we need to hand-craft an intermediate evaluation target to reflect our goal, such as ensuring the final aggregated Score accurately represents both click-through and long-play precision.

EvalMetric =2 × GAUC(Score, 
$$y^{\text{ctr}}$$
)  
+ 5 × GAUC(Score,  $y^{\text{lvtr}}$ ) + ... (4)

In contrast to ranking models, here we employ the fused Score for all objectives to calculate the GAUC (see the details in Sec.5.1). The fixed hyper-parameters 2 and 5 are according to the expert knowledge, which should be carefully selected. Actually, these weights reflect the relative business importance with the expert knowledge.

 Parameter Tuning: Based on the pre-defined formulation style and hand-crafted evaluation metric, our goal is to search for a group of appropriate weights which maximize the EvalMetric:

$$\alpha^*, \beta^*, \gamma^* = \operatorname{argmax}_{\alpha, \beta, \gamma} \text{EvalMetric}$$
 (5)

where the  $\alpha^*, \beta^*, \gamma^* \in \mathbb{R}$  are the optimal parameters under the EvalMetric setting. In this way, we typically employ Bayesian-based parameter search tools to accelerate this process, such as TPE-based Optuna and Paradance<sup>1</sup>. However, the searched parameters may **only achieve optimal performance at the single-dimensional EvalMetric** in the offline datasets, not necessarily reflect the real-world online A/B test performance from diverse aspects. Therefore, we typically combine the parameter searching with expert knowledge to ensure online effectiveness with further hand-crafted efforts.

<sup>&</sup>lt;sup>1</sup>https://github.com/optuna/optuna and https://github.com/yinsn/ParaDance. Specifically, the ParaDance is an open-source tool developed by this paper author Yin Cheng.

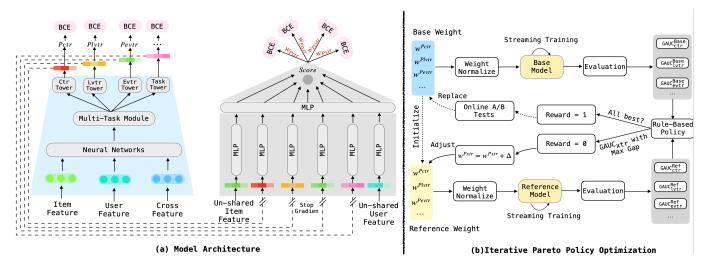


Figure 3: Pantheon model architecture and its iterative Pareto policy optimization strategy.

#### 3 PANTHEON WORKFLOW

In this section, we introduce the details of our model, Pantheon. We first express how our Pantheon performs multi-objective optimization through joint training with the ranking model. Afterwards, we describes the IPPO strategy to show how we find a group of loss weights meets the Pareto-optimal constraints.

#### 3.1 Fusion Score Generation

As shown in Figure 3(a), our Pantheon can be seen as an additional 'plugin' component from ranking model. Here we offer our model architecture and training loss designing insights.

3.1.1 Model Architecture. Benefiting from joint training with the ranking model, our training framework has the advantages of largescale parameters and massive training data naturally. Hence, in model designing, we focus on that: (a) reuse the extracted highdimension characterization from sophisticated ranking model, we concatenate the multiple tasks' representations as our Pantheon inputs, which alleviates the information decay of highly-compressed Pxtr (b) to avoid two-staged formulation-based ensemble sort paradigms, build an end-to-end ensemble model. For the first point, while employing the high-dimensional task-specific representations t (in Eq.(1)) to support our Pantheon, we apply the 'stop-gradient' operator to disentangle our ensemble sort component with the ranking model, which enforces gradient isolation between the ranking model and Pantheon. For the second point, we introduce an additional set of learnable user/item features to serve as personalized inputs for our Pantheon. According to them, our Pantheon inputs can be formed as:

$$P = \{ItemFea, Stop-Gradient(t^{ctr}), Stop-Gradient(t^{lvtr}), Stop-Gradient(t^{cmtr}), \dots, Stop-Gradient(t^{xtr}), UserFea\}.$$
(6)

where the ItemFea/UserFea  $\in \mathbb{R}^d$  are the un-shared additional item/user features,  $\mathbf{t}^{\mathsf{xtr}} \in \mathbb{R}^d$  are the tower output representation in Eq.( 1), and the P denotes the high-dimensional final input of our Pantheon. In our experiments, we find that the un-shared additional

item/user features also significantly accelerated the convergence of Pantheon. According to the fine-grained information, we utilize a simple network to generate the fusion score as:

$$Score = Ensemble\_Encoder(P).$$
 (7)

where the Ensemble\_Encoder( $\cdot$ ) is a MLP-based networks with Sigmoid activated function (in Figure 3(a)), and the Score  $\in$  (0, 1) is the final fused float result.

3.1.2 Training loss function. Compared with original ranking model that utilizes different Pxtr to predict different tasks in Eq.(2), in multi-objective optimization paradigm, we need to apply a single Score to compare different items while maximizes the multiple objectives metrics. Therefore, in our Pantheon, we employ a standard simple additive weighting mechanism to combine multiple objectives information:

$$\begin{split} \mathcal{L}_{\mathsf{Pantheon}}^{\mathsf{ctr}} &= - \big( y^{\mathsf{ctr}} \log \left( \mathsf{Score} \right) - (1 - y^{\mathsf{ctr}}) \log \left( 1 - \mathsf{Score} \right) \big), \\ \mathcal{L}_{\mathsf{Pantheon}}^{\mathsf{lvtr}} &= - \big( y^{\mathsf{lvtr}} \log \left( \mathsf{Score} \right) - (1 - y^{\mathsf{lvtr}}) \log \left( 1 - \mathsf{Score} \right) \big), \\ \mathcal{L}_{\mathsf{Pantheon}} &= w^{\mathsf{ctr}} \mathcal{L}_{\mathsf{Pantheon}}^{\mathsf{ctr}} + w^{\mathsf{lvtr}} \mathcal{L}_{\mathsf{Pantheon}}^{\mathsf{lvtr}} + \dots \end{split}$$

where the  $\mathcal{L}_{Pantheon}$  is the final training loss objective, and the  $w^{\text{ctr}}$ ,  $w^{\text{lvtr}}$  are the **positive weights** to balance different objectives importance. Since different objectives show different positive sample densities, the influence of different tasks varies even same weight. To avoid our model degenerate into a pure click model, we employ such weights to balance the influence of tasks, which can lead to learn the global objectives' interest of each user. Particularly, in our experiments, we found that the weight of the loss is crucial to the final convergence result of the model. In the next section, we will elaborate on our IPPO strategy to show the process of how to iterate these weights.

## 3.2 Iterative Pareto Policy Optimization

In this section, we give our training policy from reinforcement learning perspective, to automatically search a group of weights to reach a Pareto frontier. *3.2.1 Basic Concepts.* Before going on, we first introduce some basic concepts for better understanding:

PROPOSITION 3.1. Pareto Optimality under Positive Weights. Given scalarized loss  $\mathcal{L} = \sum_{i=1}^{n} w^{i} \mathcal{L}^{i}$  with strictly positive weights  $w^{i} > 0$  ( $\forall i$ ), let  $\theta^{*}$  be a local minimizer of  $\mathcal{L}$ . Then  $\theta^{*}$  is locally Pareto optimal for the multi-objective problem  $\min_{\theta}(\mathcal{L}^{1}, \ldots, \mathcal{L}^{n})$ . Formally:

$$\theta^* \in \operatorname*{arg\,min}_{\theta} \mathcal{L} \implies \nexists \theta' \text{ s.t. } \begin{cases} \mathcal{L}^i(\theta') \leq \mathcal{L}^i(\theta^*), \ \forall i \\ \mathcal{L}^k(\theta') < \mathcal{L}^k(\theta^*), \ \exists k \end{cases} \tag{9}$$

*Proof*: Assume the contrary that  $\exists \theta'$  Pareto dominating  $\theta^*$ . Then:

$$\begin{split} \sum_{i=1}^n w^i \mathcal{L}^i(\theta') &< \sum_{i=1}^n w^i \mathcal{L}^i(\theta^*) \quad (\because w^k > 0 \text{ and } \mathcal{L}^k(\theta') < \mathcal{L}^k(\theta^*)) \\ &\Rightarrow \mathcal{L}(\theta') < \mathcal{L}(\theta^*) \end{split}$$

contradicting the local optimality of  $\theta^*$  for  $\mathcal{L}$ .

PROPOSITION 3.2. Invariance Property under Homogeneous Scaling. For any positive constant k > 0, rescaling the total loss  $\mathcal{L} = \sum_{\mathsf{xtr}}^{\{\mathsf{ctr}, \mathsf{lvtr}, \ldots\}} w^{\mathsf{xtr}} \mathcal{L}^{\mathsf{xtr}} \to k \mathcal{L} = \sum_{\mathsf{xtr}}^{\{\mathsf{ctr}, \mathsf{lvtr}, \ldots\}} k w^{\mathsf{xtr}} \mathcal{L}^{\mathsf{xtr}}$  preserves the optimal model parameters and the resulting score distri-

bution. Formally,  $\theta^*$  be the same optimal model parameters:

$$\theta^* = \operatorname{argmin}_{\theta} \mathcal{L} = \operatorname{argmin}_{\theta} k \times \mathcal{L}$$
 (11)

(10)

*Proof*: We first derive the gradient direction as:

$$\nabla_{\theta}(k\mathcal{L}) = k\nabla_{\theta}\mathcal{L} \propto \nabla_{\theta}(\mathcal{L}) = \nabla_{\theta}\mathcal{L} \tag{12}$$

Thus, gradient descent paths for  $\mathcal{L}$  and  $k\mathcal{L}$  are identical when learning rates are scaled by  $\frac{1}{k}$ . Next, at the optimal convergence status, both losses share the same critical point condition:

$$k\nabla_{\theta}\mathcal{L} = 0 \iff \nabla_{\theta}\mathcal{L} = 0$$
 (13)

Thus weight normalization does not change the convergence point.

Definition 3.3. **Relative Importance**. The relative importance between objective i and j is quantified by:

$$\rho_{ij} \triangleq \frac{w^i}{w^j} \tag{14}$$

which determines the locally optimal Pareto frontier location in multiobjective optimization.

Therefore, our goal is to determine a better Pareto-optimal relative positive weights  $\rho$  through weights normalization, which preserves the optimization dynamics while ensuring fair evaluation metric comparison across objectives with the same learning rate.

- 3.2.2 Reinforcement Optimizing Policy. Reinforcement learning (RL) agents are fundamentally designed to maximize cumulative rewards by interacting with environments through sequential actions. Inspired by this paradigm, we propose an Iterative Pareto Policy Optimization (IPPO) mechanism to automatically discover improved Pareto frontiers through RL principles. Notably, our ensemble ranking framework naturally aligns with RL concepts, as illustrated by the following mappings in Figure 3(b):
- **State**: The weights  $w^{xtr} > 0$  indicates the locally Pareto frontier.
- Agent: The trainable model to chase the dynamic user interests.

- Environment: Real-time user-item interaction logs for streaming training and evaluation.
- Reward: Binary 0/1 signal triggered when the reference agent outperforms the base model across all GAUC metrics.
- Action: Replacing base model, or adjusting reference model's weights, e.g.,  $\Delta = \frac{0.1}{N}$ , where *N* is the objectives number.

The core objective is to learn an optimal policy  $\pi(\text{Action}|\text{State})$  that maps states to action distributions, thereby maximizing the expected cumulative reward. Following the self-play idea, our IPPO framework iteratively maintains two policy models: a fixed base model serving as a performance benchmark, and a reference model that explores parameter adjustments to surpass the base model. In practice, we devise a rule-based policy governing actions on the reference model:

- If all evaluation metrics dominate: Update the better Pareto frontier by replacing the base model with the reference model.
- Otherwise: Adjusting the corresponding objective weight with maximum GAUC gap (w<sup>xtr</sup> = w<sup>xtr</sup> + Δ) for subsequent iterations, where Δ is small value, e.g., Δ = <sup>0.1</sup>/<sub>M</sub>.

In this way, we can encourage our model explores a series of 'good' weights to reach a better Pareto frontier automatically.

#### 4 SCORE DISTRIBUTION DISCUSSION

In this section, we discuss and provide our first-hand empirical insights: how the Pantheon output distribution effect the online A/B test. In practice, different weight group of our Pantheon will produce different fusion score distributions (in Eq.(7)), thus they have different sorting abilities. To analyze the sorting ability, we have conducted a lot of experiments and found that they have the following characteristics based on its output distributions:

- Mean value determines exposure number: larger mean value will increase the amount of live-streaming distribution to users.
- Variance value determines exposure position: larger variance value make users watch live-streaming earlier than short-video.

According to our observation, we could add a Mean-Variance calibration between two different models for a fair online A/B test comparison. In our system, we align the models of the experimental Pantheon variants group to the output distribution of the baseline Pantheon. We found that this technique can ensure that the exposure numbers and positions of different models are aligned.

#### 5 EXPERIMENTS

In this section, we answer the following research questions:

- **RQ1**: How does Pantheon bring gains in offline evaluation?
- RQ2: How does Pantheon contributes online improvements?
- RQ3: What is the impact of model architecture and model input on offline performance?
- RQ4: How does Pantheon changes our services ecology?
- RQ5: How does the final fusion score more balanced in its dependence on multiple objectives?

#### 5.1 Data-streaming and Metrices

We conduct extensive experiments at the live-streaming recommendation services at Kuaishou, which is one of the largest recommendation scenario including over 400 million users and several

Table 1: Offline Results in term GAUC in Live-Streaming Recommendation.
---

Method	wtr	ltr	lvtr	ctr	evtr	inlvtr	inevtr
Ranking Model	66.49%	74.73%	69.56%	64.18%	66.07%	60.36%	60.76%
Formulation	57.61%	68.32%	65.81%	61.21%	60.75%	57.40%	56.97%
Pantheon	59.94%	71.62%	67.46%	63.76%	62.99%	58.69%	58.17%
Improvement↑	+2.33%	+3.30%	+1.55%	+2.55%	+2.24%	+1.19%	+1.20%

Table 2: Online Pantheon A/B Testing Performance (%) at Kuaihou.

Scenarios	Exposure	Clicked User	Watch Time	In-Room Watch Time	Gift Count	Follow
Scenario#1	+0.133%	+1.010%	+0.722%	+0.780%	+0.068%	+2.246%
Scenario#2	+1.927%	+1.671%	+1.766%	+1.637%	+0.202%	+2.638%
Scenario#3	+0.244%	+1.039%	+0.257%	+0.299%	+1.437%	+1.297%
Scenario#4	-0.109%	+0.518%	+1.292%	+0.881%	+1.335%	+0.208%

Table 3: Offline Ablation Studies in term of GAUC at Live-streaming recommendation.

Method	wtr	ltr	lvtr	ctr	evtr	inlvtr	inevtr
Formulation	57.61%	68.32%	65.81%	61.21%	60.75%	57.40%	56.97%
Pxtr&MLP	59.20%	70.93%	66.91%	63.23%	62.13%	57.92%	57.81%
Hidden-State&MLP	59.94%	71.62%	67.46%	63.76%	62.99%	58.69%	58.17%
Hidden-State&Transformer	60.56%	72.15%	67.57%	63.89%	63.06%	58.79%	58.31%

billion exposure logs in our data-streaming. For evaluation, we select the the wide-used GAUC as our evaluation metric, since it can comprehensively measure the model's ability among users:

$$GAUC = \sum_{u} w_u AUC_u \quad \text{where } w_u = \frac{\text{exposure}_u}{\text{all exposure}}, \quad (15)$$

where the  $w_u$  denotes the user u's exposure ratio, the AUC $_u$  is the AUC-ROC results of user u.

## 5.2 Offline Comparison (RQ1)

The main experiment results are shown in Table 1, which describes the most importance objectives evaluation results in our system (i.e., wtr/follow, ltr/like, lvtr/long-view, ctr/click, evtr/effective-view, inlvtr/in-room long-view and inevtr/in-room effective-view). Here we first report the Ranking model metrics, since it estimates each objective separately, thus it represents the ceiling performance of our system. Based on the Ranking model output Pxtrs, we next show our online formulation results, which have been served as our ensemble sort stage solution for past several years. From it, we could found that the offline evaluation results are reduced under various objectives, e.g.,  $66.4\% \Rightarrow 57.6\%$  in wtr. It is reasonable to have a such performance degradation since the trade-off between performance and interpretability in multi-objective optimization. For our Pantheon, it obviously shows statistical improvements over the formulation-based ensemble sort in all objectives with a large number of improvements, average +1.62% in different objectives. It is the largest modification in past years at Kuaishou live-streaming, which demonstrates our Pantheon could automatically converge to a better state to balance different objectives' performance under the proposed IPPO technique.

# 5.3 Online A/B Tests (RQ2)

In this section, we deploy our Pantheon to four different Scenarios to replace the previous formulation-based ensemble sort to response real recommendation requests. In our service, the most important online metrics are the clicked user, watching-time and gift count, which reflect the watching live-streaming user group, total amount of time spend on live-streaming and the value of digital gifts. Besides, we also report the exposure metric, which measures the load of live-streaming users watched, to help us make a fair comparison with the base formulation-based ensemble sort approach. Specifically, due to the large scale of our business, the about 0.1% improvement in clicked User is statistically significant enough to our system. According to Table 2, we could find that our Pantheon achieves a very significant improvement of +1.010%, +1.671%, +1.039% and +0.518% in term of the clicked user in four different scenarios, respectively. Meanwhile, our Pantheon also bring significant gain at the watch time and interaction metrics (the biggest gains in past year), which reveals our method could converge to a better Pareto frontier for our system.

## 5.4 Ablation Study (RQ3)

In this section, we construct several model variants to validate our Pantheon framework effectiveness. As shown in Figure 3(a), our Pantheon has two additional optimizing directions to enhance ensemble sort ability, from the model input (in Eq.(6)) and ensemble encoder (in Eq.(7)). For a fair comparison, we conducted three aspects modifications:

 Pxtr v.s. Hidden-State: Does the high-dimensional hidden-state t preserves more information than the highly-compressed Pxtr, thereby improving ensemble sort performance?

Table 4: Exposure Ecology Analysis across User Group.

User Group	Activate	Exposure	In-Room Eff-View
	High	-2.2%	+3.5%
In-Room	Mid	-0.9%	+5.7%
	Low	+3.6%	+9.8%
	High	+0.7%	+4.8%
Out-Room	Mid	+6.4%	+12.9%
	Low	+4.3%	+6.9%

Table 5: Behaviour Pattern Ecology Analysis.

Classification	Behaviours	Improvement
	Long&Inter&Gift	+0.83%
Multiple-Good	Long&Inter	+5.32%
Muniple-Good	Long&Gift	+0.53%
	Inter&Gift	+3.65%
	Long-View	+10.84%
Single-Good	Interaction	+9.13%
	Gift	+1.87%

Table 6: Kendall's  $\tau$  Coefficient Analysis.

Pxtr of	Scer	nario#1	Scenario#2		
Ranking Model	Fomula Pantheon		Fomula	Pantheon	
inlvtr	0.2657	0.2158	0.2278	0.2106	
inetr	0.2230	0.2104	0.2001	0.2009	
lvtr	0.2099	0.1412	0.1898	0.1339	
evtr	0.1645	0.1297	0.1514	0.1197	
ltr	0.1194	0.1011	0.1186	0.1032	
wtr	0.1054	0.1023	0.0997	0.1044	

 MLP v.s. Transformer: Does the advanced model architecture could further enhance ensemble sort performance?

According to Table 3, we can draw the following conclusions: (1) Compared with Formulation, the Pxtr&MLP shows superior performance across these objectives. It demonstrates the effectiveness of our IPPO mechanism to ensure our framework can be convergence to a promising Pareto frontier adaptively. (2) Compared with Pxtr&MLP, the Hidden-State&MLP consistently yields the significant accuracy improvement on all objectives, which reveals that joint training our Pantheon with Ranking model's multi-task module hidden-state is a more powerful way to enhance ensemble sort ability (3) Instead of utilizing the MLP as the Ensemble\_Encoder(·), we further upgrade it to the Transformer architecture to obtain the final score. Compared with Hidden-State&MLP, the Hidden-State&Transformer version shows significant gains over MLP-based Pantheon, which demonstrates a new direction to explore more complex architecture to build a more powerful ensemble sort.

# 5.5 Ecology Changes (RQ4)

In this section, we aim to answer a question: what kind of changes does our Pantheon ensemble sort bring to our service ecology? As shown in Figure 1, live-streaming has two styles (e.g. Out-Room/In-Room form), and different users have different habits when they

surf on our platform. For the diverse users, we first divide them into two user group categories: those who have a preference for watching In-Room and those who usually watch Outside-Room. Regarding the two different user group, we further divide them into three groups according to whether they are active on our platform. Here we show empirical performance in Table 4, which describes the exposure changes rate compared with the formulation-based ensemble sort method, and we have the observations:

- The exposure traffic has shifted from "In-Room users" to "Out-Room users", while it has increased the effective usage scale. This phenomenon shows that our Pantheon optimizes traffic exposure from the top user group, frees up capacity for potential new users, and increases opportunities to distribute to more users.
- For the In-Room user group, our Pantheon utilizes a smaller traffic pool but drives a higher experience, e.g., In-Room Eff-View rate +4.8%/+12.9%/+6.9%. This phenomenon shows that our Pantheon recommends more right live-streamings to our users, building a better environment for our system.

Moreover, to investigate the impact of our Pantheon effects more comprehensively, we conduct another comparison from the user behaviour perspective. From the perspective of live-streaming business, we can classify the user's behavior into three aspects (i.e., watching time, interaction, gift) to identify good user watching.

- Single-Good: only one of the following signals occurred during user watching a live-streaming: long-view (> 60s), interaction (like/comment/forward/follow), or send gift.
- Multiple-Good: occurs two or more signals.

There we show the ecology change results in Table 5, our Pantheon consistently yields significant improvements across the single-good and multiple-good behaviours. We suppose the reason lies in that our IPPO follows the Pareto optimizing strategy, which ensure that optimizing one metric without reducing other metrics.

# 5.6 Objective Dependence Analysis (RQ5)

In this section, we explore the fusion score's dependency among the different objectives. Here we conducted the Kendall's  $\tau$  analysis between the **Ranking model** Pxtr with the fusion scores of formula and our pantheon. Specifically, the Kendall's  $\tau$  coefficient value is in the ranges from [-1,+1], higher score means more related. According to the Table 6, we could find that our Pantheon also maintains the **exact same order of importance among the multiple objective** with the origin formula, while has the minor variations in the absolute magnitude of the coefficient weights. Such phenomenon demonstrates that our Pantheon is capable of effectively capturing a more balanced objectives usage. We attribute the reason to our model utilizing the high-dimensional representations, instead of the highly-compressed numerical scores Pxtr, which could enhance our ensemble sort robustness to reduce the numerical fluctuations caused by top important objectives.

### 6 CONCLUSION

In this paper, we explore a more flexible ensemble sort approach, Pantheon, which fully replaced the wide-used formula-based ensemble sort and successfully deployed at Kuaishou, serving 400 Million users per day. Specifically, compared with traditional formula-based

ensemble sort, our Pantheon has several key insights: (1) our Pantheon is a 'plugin' component that jointly training with Ranking model. Meanwhile, we do not utilize the highly-compressed numerical scores Pxtr as input, but employ the high-dimensional task representations as Pantheon inputs, which maintains more finegrained information. (2) Our Pantheon follows a reinforcement optimizing policy IPPO to search the most appropriate objective weights automatically, which could guarantee our model to find better Pareto frontier without hurt some objectives to improve other objectives. In the future, we will utilize our Pantheon as reward model, to explore industrial-scale generative recommendation.

#### REFERENCES

- [1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. In arXiv.
- [2] Aman Agarwal, Xuanhui Wang, Cheng Li, Michael Bendersky, and Marc Najork. 2019. Addressing trust bias for unbiased learning-to-rank. In Proceedings of the ACM on Web Conference.
- [3] Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. 2019. Optuna: A next-generation hyperparameter optimization framework. In ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD).
- [4] Jiangxia Cao, Shen Wang, Yue Li, Shenghui Wang, Jian Tang, Shiyao Wang, Shuang Yang, Zhaojie Liu, and Guorui Zhou. 2024. Moment&Cross: Next-Generation Real-Time Cross-Domain CTR Prediction for Live-Streaming Recommendation at Kuaishou. (2024).
- [5] Yang Cao, Changhao Zhang, Xiaoshuang Chen, Kaiqiao Zhan, and Ben Wang. 2025. xMTF: A Formula-Free Model for Reinforcement-Learning-Based Multi-Task Fusion in Recommender Systems. In Proceedings of the ACM on Web Conference.
- [6] Jianxin Chang, Chenbin Zhang, Yiqun Hui, Dewei Leng, Yanan Niu, Yang Song, and Kun Gai. 2023. PEPNet: Parameter and Embedding Personalized Network for Infusing with Personalized Prior Information. In ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD).
- [7] Gaode Chen, Ruina Sun, Yuezihan Jiang, Jiangxia Cao, Qi Zhang, Jingjian Lin, Han Li, Kun Gai, and Xinghua Zhang. 2024. A Multi-modal Modeling Framework for Cold-start Short-video Recommendation. In ACM Conference on Recommender Systems (RecSys).
- [8] Paul Covington, Jay Adams, and Emre Sargin. 2016. Deep Neural Networks for YouTube Recommendations. In ACM Conference on Recommender Systems
- [9] Jean-Antoine Désidéri. 2012. Multiple-gradient descent algorithm (MGDA) for multiobjective optimization. Comptes Rendus Mathematique (2012).
- [10] Jörg Fliege and Benar Fux Svaiter. 2000. Steepest descent methods for multicriteria optimization. Mathematical methods of operations research (2000).
- [11] Malay Haldar, Mustafa Abdool, Liwei He, Dillon Davis, Huiji Gao, and Sanjeev Katariya. 2023. Learning To Rank Diversely At Airbnb. In ACM International Conference on Information and Knowledge Management (CIKM).
- [12] Po-Sen Huang, Xiaodong He, Jianfeng Gao, Li Deng, Alex Acero, and Larry Heck. 2013. Learning deep structured semantic models for web search using clickthrough data. In Proceedings of the 22nd ACM international conference on Information & Knowledge Management.
- [13] Alex Kendall, Yarin Gal, and Roberto Cipolla. 2018. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In IEEE/CVF Computer Vision and Pattern Recognition Conference (CVPR).
- [14] Vitaly Kurin, Alessandro De Palma, Ilya Kostrikov, Shimon Whiteson, and Pawan K Mudigonda. 2022. In defense of the unitary scalarization for deep multi-task learning. Advances in Neural Information Processing Systems (NeurIPS) (2022)
- [15] Chao Li, Zhiyuan Liu, Mengmeng Wu, Yuchi Xu, Huan Zhao, Pipei Huang, Guoliang Kang, Qiwei Chen, Wei Li, and Dik Lun Lee. 2019. Multi-interest network with dynamic routing for recommendation at Tmall. In ACM International

- Conference on Information and Knowledge Management (CIKM).
- [16] Wuchao Li, Rui Huang, Haijun Zhao, Chi Liu, Kai Zheng, Qi Liu, Na Mou, Guorui Zhou, Defu Lian, Yang Song, et al. 2024. DimeRec: A Unified Framework for Enhanced Sequential Recommendation via Generative Diffusion Models. In arXiv.
- [17] Chengzhi Lin, Chuyuan Wang, Annan Xie, Wuhong Wang, Ziye Zhang, Can-guang Ruan, Yuancai Huang, and Yongqi Liu. 2025. AlignPxtr: Aligning Predicted Behavior Distributions for Bias-Free Video Recommendations. arXiv (2025).
- [18] Xi Lin, Hui-Ling Zhen, Zhenhua Li, Qing-Fu Zhang, and Sam Kwong. 2019. Pareto multi-task learning. Advances in Neural Information Processing Systems (NeurIPS) (2019)
- [19] Chi Liu, Jiangxia Cao, Rui Huang, Kuo Cai, Weifeng Ding, Qiang Luo, Kun Gai, and Guorui Zhou. 2024. CRM: Retrieval Model with Controllable Condition. arXiv (2024).
- [20] Chi Liu, Jiangxia Cao, Rui Huang, Kai Zheng, Qiang Luo, Kun Gai, and Guorui Zhou. 2024. KuaiFormer: Transformer-Based Retrieval at Kuaishou. (2024).
- [21] Qingyun Liu, Zhe Zhao, Liang Liu, Zhen Zhang, Junjie Shan, Yuening Li, Shuchao Bi, Lichan Hong, and Ed H Chi. 2023. Multitask Ranking System for Immersive Feed and No More Clicks: A Case Study of Short-Form Video Recommendation. In ACM International Conference on Information and Knowledge Management (CIKM).
- [22] Yueyang Liu, Jiangxia Cao, Shen Wang, Shuang Wen, Xiang Chen, Xiangyu Wu, Shuang Yang, Zhaojie Liu, Kun Gai, and Guorui Zhou. 2025. LLM-Alignment Live-Streaming Recommendation. arXiv (2025).
- [23] Ziru Liu, Jiejie Tian, Qingpeng Cai, Xiangyu Zhao, Jingtong Gao, Shuchang Liu, Dayou Chen, Tonghao He, Dong Zheng, Peng Jiang, et al. 2023. Multi-task recommendations with reinforcement learning. In Proceedings of the ACM on Web Conference.
- [24] Yucheng Lu, Jiangxia Cao, Xu Kuan, Wei Cheng, Wei Jiang, Jiaming Zhang, Yang Shuang, Liu Zhaojie, and Liyin Hong. 2025. LiveForesighter: Generating Future Information for Live-Streaming Recommendations at Kuaishou. arXiv (2025).
- [25] Xinchen Luo, Jiangxia Cao, Tianyu Sun, Jinkai Yu, Rui Huang, Wei Yuan, Hezheng Lin, Yichen Zheng, Shiyao Wang, Qigen Hu, et al. 2024. QARM: Quantitative Alignment Multi-Modal Recommendation at Kuaishou. arXiv (2024).
- [26] Xiao Lv, Jiangxia Cao, Shijie Guan, Xiaoyou Zhou, Zhiguang Qi, Yaqiang Zang, Ming Li, Ben Wang, Kun Gai, and Guorui Zhou. 2024. MARM: Unlocking the Future of Recommendation Systems through Memory Augmentation and Scalable Complexity. arXiv (2024).
- [27] Jiaqi Ma, Zhe Zhao, Xinyang Yi, Jilin Chen, Lichan Hong, and Ed H Chi. 2018. Modeling Task Relationships in Multi-task Learning with Multi-gate Mixture-of-Experts. In ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD).
- [28] Pingchuan Ma, Tao Du, and Wojciech Matusik. 2020. Efficient continuous pareto exploration in multi-task learning. In *International Conference on Machine Learn*ing (ICML), PMLR.
- [29] Qi Pi, Guorui Zhou, Yujing Zhang, Zhe Wang, Lejian Ren, Ying Fan, Xiaoqiang Zhu, and Kun Gai. 2020. Search-based User Interest Modeling with Lifelong Sequential Behavior Data for Click-Through Rate Prediction. In ACM International Conference on Information and Knowledge Management (CIKM).
- [30] Ozan Sener and Vladlen Koltun. 2018. Multi-task learning as multi-objective optimization. Advances in Neural Information Processing Systems (NeurIPS) (2018).
- [31] Hongyan Tang, Junning Liu, Ming Zhao, and Xudong Gong. 2020. Progressive Layered Extraction (PLE): A Novel Multi-Task Learning (MTL) Model for Personalized Recommendations. In ACM Conference on Recommender Systems (PROSUL).
- [32] Jie Tang, Huiji Gao, Liwei He, and Sanjeev Katariya. 2024. Multi-objective Learning to Rank by Model Distillation. In ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD).
- [33] Xu Wang, Jiangxia Cao, Zhiyi Fu, Kun Gai, and Guorui Zhou. 2024. HoME: Hierarchy of Multi-Gate Experts for Multi-Task Learning at Kuaishou. (2024).
- [34] Qihua Zhang, Junning Liu, Yuzhuo Dai, Yiyan Qi, Yifan Yuan, Kunlun Zheng, Fan Huang, and Xianfeng Tan. 2022. Multi-task fusion via reinforcement learning for long-term user satisfaction in recommender systems. In ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD).
- [35] Guorui Zhou, Xiaoqiang Zhu, Chenru Song, Ying Fan, Han Zhu, Xiao Ma, Yanghui Yan, Junqi Jin, Han Li, and Kun Gai. 2018. Deep Interest Network for Click-Through Rate Prediction. In ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD).