Hardness-aware Privileged Features Distillation with Latent Alignment for CVR Prediction

Huining Yuan*
ByteDance Inc.
Beijing, China
yuanhuining0@gmail.com

Wenpeng Zhang*†
ByteDance Inc.
Beijing, China
zhangwenpeng0@gmail.com

Zijie Hao
ByteDance Inc.
Beijing, China
haozijie1993@163.com

Zengde Deng
ByteDance Inc.
Beijing, China
dengzengde@gmail.com

Abstract

In computational advertising, predicting the post-click conversion rate (CVR) using deep neural networks (DNNs) benefits a lot from privileged features, which can be collected for offline training but are unavailable during online serving. To utilize these privileged signals, privileged features distillation (PFD) methods incorporate a teacher model with privileged features to guide the CVR model. However, existing PFD approaches fail to put more emphasis on poorly predicted instances where the teacher's guidance is most crucial, and thus suffer from overconfidence on "easy" instances. In this work, we propose Hardness-aware Privileged Features Distillation (HA-PFD) for enhancing CVR prediction in real-world advertising recommender systems. We specifically design focal-style distillation losses that adaptively adjust the weight of each instance based on its "hardness". This method prioritizes poorly predicted instances during the distillation process, resulting in improved ranking performance and better model calibration. Additionally, we incorporate latent-level distillation into the PFD framework for the first time, which facilitates the student's representation learning through a straightforward layer alignment approach. We also propose a method for selecting privileged features based on their relevance to the conversion label. We conduct extensive offline experiments on large-scale, real-world datasets and online experiments on Douyin, a short video platform with billions of live users. In the offline evaluation, HA-PFD exhibits competitive performance and superior model calibration compared to existing state-of-the-art methods. In the online experiments, HA-PFD significantly improves advertiser value and conversions. Now we have deployed HA-PFD as the main online serving model on our short video platform.

*Both authors contributed equally to this research.
†Corresponding author.

KDD '25, Toronto, ON, Canada.

ACM ISBN 979-8-4007-1454-2/25/08
https://doi.org/10.1145/3711896.3737231

CCS Concepts

• **Information systems** → **Users and interactive retrieval**; **Retrieval models and ranking**; • **Computing methodologies** → **Neural networks**.

Keywords

Privileged Features; Knowledge Distillation; Recommender Systems; Computational Advertising

ACM Reference Format:
Huining Yuan, Wenpeng Zhang, Zijie Hao, and Zengde Deng. 2025. Hardness-aware Privileged Features Distillation with Latent Alignment for CVR Prediction. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2 (KDD '25), August 3–7, 2025, Toronto, ON, Canada.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3711896.3737231

1 Introduction

In recommender systems, features play a fundamental role in predicting post-click conversion rate (CVR) with deep neural networks (DNNs) [7, 9, 13]. Accurate and informative features lay a solid foundation for CVR prediction, and elevate the upper bound for model performance. However, some of the most powerful features are often unavailable for CVR models during online serving and exist only in offline training. For example, in an advertising system, actions like "click" and "add to cart" are strong indicators of a potential "purchase," yet these actions are unknown when the ad recommendation is initially generated and sent to the user. Such features can only be collected afterward for the offline dataset and are referred to as privileged features [41, 43].

To exploit privileged information for CVR prediction, a typical method in the industry involves using techniques from multi-task learning (MTL) [40, 48]. This approach entails designing an auxiliary prediction task for each privileged feature, learning encoders (auxiliary towers) to output auxiliary embeddings that facilitate the main task of CVR prediction. However, as the number of privileged features increases, this method inevitably poses a considerable computational burden.

Recent studies have introduced knowledge distillation (KD) methods [10, 16] to model privileged features, leveraging advances in transfer learning and model compression to address the drawbacks of auxiliary towers. KD typically involves two separate models: the student and the teacher. The teacher is provided with more

training data or larger model capacity. The student then mimics the teacher's predictions (logit-based) [16, 49] or latent features (latent-based) [6, 15, 31, 36, 45], distilling "knowledge" from the teacher to the student. In the case of privileged features distillation (PFD), the teacher and student are given different input features: the teacher receives both privileged and regular features, while the student receives only regular features [11, 41, 43]. During offline training, the knowledge of privileged features is distilled from the teacher to the student, enhancing the student's performance in CVR prediction. During online serving, only the student is used for CVR prediction, which avoids the demand for privileged features.

In this work, we further advance PFD by focusing on CVR prediction in real-world advertising systems. Despite notable advances of PFD in general recommender systems, we identify three critical challenges for effectively performing PFD in advertising systems.

(1) **Hardness awareness.** Existing PFD methods assign equal weights to all instances during distillation. However, for easily predicted instances, the student can already make accurate predictions without teacher supervision; performing distillation on these "easy" instances is redundant. Conversely, it is on the "hard" instances that the teacher's insights are most crucial for improving student's performance. Thus, it is desirable to design an adaptive weighting distillation method that adjusts the weight of each instance based on its "hardness", ensuring that the student's capacity is concentrated on learning CVR prediction for hard instances during knowledge distillation.
(2) **Model calibration.** CVR prediction is a severely imbalanced binary classification problem with usually less than 20% positive instances. In this case, a model can easily suffer from miscalibration or overconfidence on easily predicted negative instances, misrepresenting the predicted CVR [12, 26, 28]. Performing PFD can magnify this phenomenon, as the overconfidence in the teacher is propagated to the subsequent student. In contrast to general recommendation systems, where the primary concern is the ranking of different instances [50], an advertising system depends on the predicted CVR scores for accurate bidding and auctioning of advertisements [44]. Therefore, it is crucial to develop a PFD method that effectively performs knowledge transfer while maintaining model calibration for unbiased CVR scores.
(3) **Feature selection.** In contrast to the MTL approach, effective PFD often requires more careful feature engineering. The privileged features should be relevant to the label, but not overly so [43]. If the teacher can easily map privileged features to the label without making full use of the regular features, the student may struggle to learn novel information with only the regular features provided.

To address these challenges, we propose Hardness-aware Privileged Features Distillation (HA-PFD) for enhancing CVR prediction in real-world advertising systems. We specifically propose two focal-style [24, 28] distillation losses that adaptively adjust the weight of each instance based on its "hardness". This method prioritizes poorly predicted instances during the distillation process, resulting in improved ranking performance and better model calibration. Additionally, we incorporate latent-level distillation into the PFD framework for the first time, which facilitates the student's representation learning through a straightforward layer alignment approach. For implementation, we carefully select a set of 10 privileged features based on their relevance to the conversion label, facilitating effective knowledge distillation.

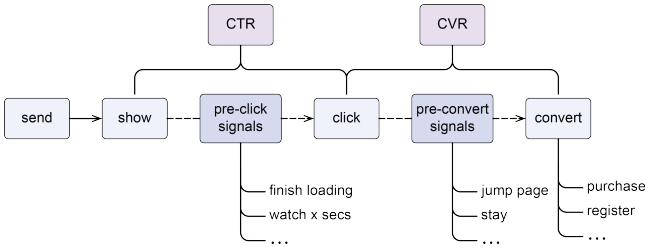

Figure 1: The lifespan of an online advertisement in short video feeds.

We apply HA-PFD for CVR prediction in a real-world advertising system of a short video platform with billions of users, conducting extensive offline evaluation on large-scale real-world datasets and online experiments on Douyin, i.e. Chinese version of TikTok, to serve live users. In the offline results, HA-PFD demonstrates competitive ranking performance compared to state-of-the-art KD methods and the common MTL approach, while significantly improving model calibration. In online experiments serving live users, HA-PFD outperforms our baseline MTL model, increasing advertiser value significantly by 3.739%, number of conversions by 1.426%, and reducing prediction error from -8.634% to -0.983%. Based on these results, we have now deployed HA-PFD as the main online serving model on our short video platform.

2 Preliminary

We begin with a brief overview of the lifecycle of an online advertisement in short video feeds. As illustrated in Figure 1, the process starts with the advertising system generating an ad to deliver a certain promotion to a user. Once the user views the ad, he or she decides whether or not to click on it. If the user clicks on the ad, the system displays more content to the user. Finally, if the user performs the intended actions, such as "registering", "subscribing", or "making a purchase", the ad impression instance is considered converted.

During this process, the probability of a click and a conversion is referred to as the click-through rate (CTR) and the post-click conversion rate (CVR), respectively. In this work, we primarily focus on CVR:

$$CTR = p(\text{click} \mid \text{show})$$
$$CVR = p(\text{convert} \mid \text{click})$$

The prediction of CVR is inherently a binary classification problem, where the model predicts the probability of conversion for each ad instance for a specific user request. This is typically achieved using the classical binary cross-entropy (CE) loss between the ground-truth label and the model-predicted CVR score:

$$\mathcal{L}_{task}(\theta) = -y \log p(x;\theta) - (1-y)\log(1-p(x;\theta)) \quad (1)$$

Here, y denotes the conversion label, $p(x;\theta)$ denotes the predicted CVR probability, and θ denotes the model parameters.

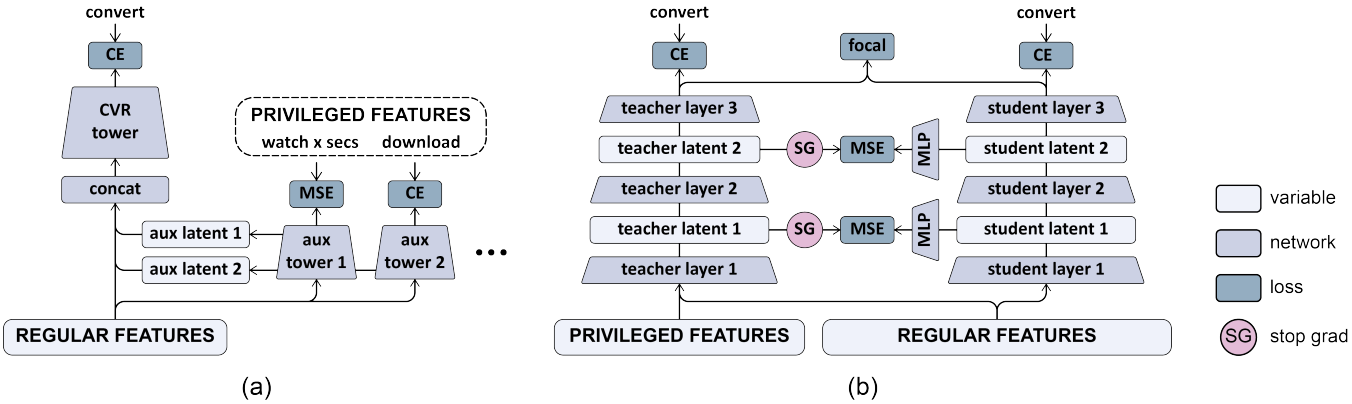

Figure 2: Two approaches for modeling privileged features: MTL VS PFD.

Notably, before clicking and converting, users may perform other actions such as "watching for x seconds" or "jumping to a page," which we term as pre-click and pre-convert events. Naturally, for CVR models, all events occurring after the ad is sent are privileged signals that cannot be used as model input during online serving [41]. However, these privileged signals can be highly relevant to the conversion label.

To enhance the performance of the CVR model, a set of privileged signals is often crafted as additional features for offline training. A widely adopted paradigm for modeling and utilizing privileged features is the auxiliary tower approach, a typical MTL application [40, 48]. As depicted in Figure 2(a), the auxiliary tower approach involves treating each privileged feature as a label for a separate prediction task and building additional towers alongside the main CVR tower to learn these auxiliary tasks. Typically, cross-entropy loss is used for discrete privileged features (e.g., "jump page"), and L2 loss (MSE) is used for continuous privileged features (e.g., "watch x seconds"). These auxiliary towers receive the same regular features as the main tower. Once trained, they act as encoders on the regular features, and their latent features from the output layer are used as additional embeddings for the main tower. Consequently, the main tower receives a concatenation of regular features and these extra embeddings as input for CVR prediction.

However, there are two main drawbacks to the auxiliary tower approach: 1) Each privileged feature introduces an additional DNN, leading to a high computational cost. This can pose a significant obstacle for deployment in efficiency-strict systems like advertising platforms. 2) The relationship between each auxiliary task and the main CVR prediction task is unclear and difficult to quantify, making the effectiveness of each auxiliary embedding questionable.

In contrast, the PFD approach offers a streamlined solution for utilizing privileged features. In this method, a teacher model is provided with privileged features as input to learn novel mappings from features to the conversion label. The student model then learns this mapping by mimicking the teacher's predictions or latent features with an additional loss for knowledge distillation. Compared to the MTL approach, PFD alleviates the burden of managing numerous auxiliary towers and allows for the adaptive utilization of each privileged feature driven by the distillation loss.

3 Method

We now introduce the proposed HA-PFD for effectively distilling privileged features in real-world advertising systems. As demonstrated in Figure 2(b), both the student and the teacher utilize a simple architecture of a three-layer DNN. Concretely, we combine logit-level distillation with latent-level distillation. For the logit level, we propose two versions of distillation losses inspired by the focal loss [24, 28], enabling harness-aware distillation. For the latent level, we adopt a simple layer alignment strategy, where the student's latent features are aligned with the teacher layer by layer, improving the student's representation learning to facilitate CVR prediction.

3.1 Logit-level distillation with focal-style distillation loss

The typical PFD approach uses logit-level distillation based on the original KD loss proposed by Hinton et al. [16], which is a simple cross-entropy loss between the teacher's logits and the student's logits, in addition to the task loss $\mathcal{L}_{task}$:

$$\begin{aligned}\mathcal{L}^{KD}_{logit}(\theta_s) = &- p_t(x,z;\theta_t)\log p_s(x;\theta_s) \\ &- \big(1-p_t(x,z;\theta_t)\big)\log\big(1-p_s(x;\theta_s)\big)\end{aligned} \tag{2}$$

Here, x and z represent the regular and privileged features, respectively. p_t and p_s denote the predicted CVR probabilities from the teacher and the student, and θ_t and θ_s are the model parameters of the teacher and the student. $\mathcal{L}^{KD}_{logit}$ is analogous to $\mathcal{L}_{task}$, except that it uses the teacher's predictions instead of the ground-truth conversion labels for supervision. Typically, student learns from both losses simultaneously.

Although the original KD loss has proven effective for transferring knowledge and enhancing the student's ranking ability, it has a significant limitation for CVR prediction in advertising systems: it assigns equal weights to all instances during distillation, disregarding the importance and difficulty of each instance. This limitation can negatively impact the improvement of student performance. Intuitively, if the CVR of an instance is easily predictable, the student can likely learn to make accurate CVR predictions independently, rendering distillation on such instances redundant. Conversely, for

hard instances where the student performs poorly, the teacher's insights from privileged features are crucial for performance improvement. Therefore, adaptively increasing the weight of these hard instances would be beneficial for improving the student's ranking performance.

Furthermore, this limitation can lead to model miscalibration. In theory, CE is a calibrated loss that converges the model prediction score to the actual CVR through direct optimization of the likelihood of the correct label, and it is intuitive to apply CE for task loss and KD loss. However, CVR prediction is a highly imbalanced binary classification problem. With fewer than 20% converted ads, over 80% of all training instances are negative. Minimizing CE in this scenario pushes the model to make overconfident negative predictions on easily predicted negative instances in order to lower the loss, resulting in miscalibrated CVR predictions that are lower than the actual CVRs [12, 28]. The KD loss $\mathcal{L}_{logit}^{KD}$ exacerbates this issue because the student is simultaneously supervised by both the ground-truth labels and the teacher's predictions. In contrast to pure ranking problems in general recommender systems, where the focus is on the relative ranking of different instances, advertising systems require accurate and well-calibrated CVR predictions for effective bidding and auctioning [44]. Therefore, PFD in advertising systems should maintain model calibration during knowledge distillation.

To mitigate these negative impacts, we propose integrating focal-style distillation losses for hardness-aware distillation. Focal loss has been successfully applied for adaptive weighting of instances and model calibration in classification tasks [28, 34]. In our PFD problem, we incorporate focal loss into logit-level distillation, which allows us to focus the student capacity on hard instances and alleviate overconfidence on easy instances at the same time:

$$\begin{aligned}\mathcal{L}_{logit}^{Focal\ V1}(\theta_s;\gamma) = \\ &- p_t(x,z;\theta_t)\big(1-p_s(x;\theta_s)\big)^{\gamma}\log p_s(x;\theta_s) \\ &- \big(1-p_t(x,z;\theta_t)\big)p_s(x;\theta_s)^{\gamma}\log\big(1-p_s(x;\theta_s)\big)\end{aligned} \tag{3}$$

Compared to the original KD loss, focal loss introduces two additional multiplication factors, $(1-p_s)^{\gamma}$ and p_s^{γ}. These factors are designed to down-weight well-predicted instances and up-weight poorly predicted ones, thereby adjusting the weight of the loss according to the instance's hardness. For a positive instance with a high predicted CVR from the teacher, if the student also confidently predicts that the instance will convert, the focal loss is down-weighted by a small $(1-p_s)^{\gamma}$, reducing the instance's impact on the student's training. Conversely, if the student has high confidence that the instance will not convert, the factor $(1-p_s)^{\gamma}$ becomes large, up-weighting the loss. The hyperparameter γ controls the balance between the two terms, with a larger γ value leading to more aggressive down-weighting of well-predicted instances. In this way, focal loss effectively emphasizes the instances that the student finds challenging, thus distilling the knowledge that truly matters. Moreover, such adaptive weighting of instances alleviates the student's overconfidence on easy instances, resulting in improved model calibration.

The logit-level distillation loss proposed in Equation (3) is a straightforward application of focal loss for PFD. In this version of the focal distillation loss, the hardness factors are determined by the student itself. In other words, the student adaptively learns from the teacher based on the alignment between its predictions and the teacher's predictions. Additionally, we propose another version of focal distillation loss in which the hardness factors are determined by the teacher:

$$\begin{aligned}\mathcal{L}_{logit}^{Focal\ V2}(\theta_s;\gamma) = &-\big(y\big(1-p_t(x,z;\theta_t)\big)^{\gamma}\log p_s(x;\theta_s) \\ &-(1-y)\big)p_t(x,z;\theta_t)^{\gamma}\log\big(1-p_s(x;\theta_s)\big)\end{aligned} \tag{4}$$

In this version of focal loss, the student learns directly from the label, with the hardness factors determined by the teacher. In this way, the teacher functions as an attention module, guiding the student on the difficulty of each instance and indicating how much focus the student should allocate to it. Empirically, we find that $\mathcal{L}_{logit}^{Focal\ V2}$ results in slightly better ranking performance than $\mathcal{L}_{logit}^{Focal\ V1}$ and demonstrates similar model calibration across most of our data settings.

3.2 Latent-level distillation with layer alignment

The latent features of a neural network provide much richer and denser information than the condensed output logits. This richness is widely leveraged in state-of-the-art KD methods for designing effective latent-level distillation [6, 29, 46, 47], where the student is encouraged to mimic the teacher's latent features rather than being directly supervised by potentially miscalibrated teacher logits. However, current works on PFD have yet to incorporate this latent information for better distillation.

In this work, we introduce latent-level distillation to the PFD framework for the first time, enhancing the student's representation learning to facilitate better CVR prediction in the output layer. Given our simple three-layer architecture for both the student and the teacher, we consider two layers of latent features for distillation. We adopt a straightforward strategy that aligns each of the student's latent features with the corresponding latent features from the teacher. To achieve this, we map the student's latent variables with a DNN and then supervise the output to match that of the teacher using a simple L2 loss:

$$\mathcal{L}_{feature}(\theta_s,\theta_g) = \sum_{i=1}^{2}\Big\| g^{(i)}\big(f_s^{(i)}(x;\theta_s);\theta_g\big) - f_t^{(i)}(x,z;\theta_t)\Big\|_2^2 \tag{5}$$

Here, $f_s^{(i)}$ and $f_t^{(i)}$ denote the student and teacher's latent features at layer i, respectively. $g^{(i)}$ is the DNN mapping from the student's latent feature to the teacher's latent feature, with θ_g being its parameters.

This simple approach is similar to FitNet [31], where the student's latent features are directly regularized to match those of the teacher. The difference is that we further bridge the alignment with an additional DNN mapping. The intuition behind this approach is straightforward: the L2 alignment encourages the student to match the superior latent features demonstrated by the teacher. The intermediate DNN mapping provides the student with extra flexibility, ensuring that the student is not required to copy the teacher exactly but rather to preserve the same information. In this way, the student learns to perform better feature extraction from the given

Table 1: Privileged features

Feature	Type	Value	Description
Page_jump	discrete	yes -> 1, no -> 0	Jump to the next page by clicking or finishing playing.
Done_loading	discrete	yes -> 1, no -> 0	Finish loading of the ad.
Play_2s	discrete	yes -> 1, no -> 0	Play 2s of the ad.
Button_click	discrete	yes -> 1, no -> 0	Click on the click button of the ad.
Page_click	discrete	yes -> 1, no -> 0	Click on anywhere in the page.
Swipe_click	discrete	yes -> 1, no -> 0	Swipe left to the next page.
Interact	discrete	yes -> 1, no -> 0	Tag or like or share.
Duration	continuous	[0, 18) -> 1, [18, +∞] -> 0	The total length of time that the user stayed on the ad.
Click_leave_time	continuous	[0, 8) -> 1, [8, +∞] -> 0	The length of time between "click" and "move on".
In_platform_time	continuous	[0, 5000) -> 1, [5000, +∞] -> 0	The total length of time the user's current session in the platform.

regular features and develops improved latent representations for subsequent layers to make CVR predictions.

Overall, the total distillation loss of HA-PFD is the summation of the latent-level loss and either one of the focal distillation losses, with a hyperparameter λ to control the balance between the two, leading to two versions of HA-PFD:

$$\mathcal{L}_{HA-PFD}(\theta_s, \theta_g; \gamma) = \mathcal{L}_{feature}(\theta_s, \theta_g) + \lambda \cdot \mathcal{L}_{logit}^{Focal}(\theta_s; \gamma) \quad (6)$$

The novelty of HA-PFD lies in two aspects: 1) the focal-style distillation losses that adjust the weight of each instance in a hardness-aware manner, concentrating the student on instances that are worth distilling; 2) the incorporation of latent-level distillation to facilitate better representation for the student. In practice, the student and the teacher are trained simultaneously in an online manner for CVR prediction [17], with the distillation loss trained alongside the task losses of the student and the teacher, yielding the total loss function:

$$\begin{aligned}\mathcal{L}_{total}(\theta_t, \theta_s, \theta_g; \gamma) = \mathcal{L}_{task}^{teacher}(\theta_t) + \mathcal{L}_{task}^{student}(\theta_s) \\ + \mathcal{L}_{HA-PFD}(\theta_s, \theta_g; \gamma)\end{aligned} \quad (7)$$

A pseudocode of HA-PFD can be found in Algorithm 1 in the Appendix.

3.3 Feature selection approach

In contrast to MTL and conventional KD for model compression, PFD often requires careful feature selection. The key to selecting or crafting privileged features is to strike a balance between relevance to the conversion label and the difficulty of learning from regular features. As demonstrated in [43], the performance of the student initially increases but then decreases as the predictive power of the privileged features increases. This suggests that if the privileged features are irrelevant to the label, the teacher cannot provide useful information to the student beyond what the regular features already offer. Conversely, if the privileged features are too relevant to the label, the teacher can easily "cheat" in CVR prediction by simply mapping the privileged features to the label, offering little novel information for the student to learn from. In this work, we carefully designed a set of 10 privileged features based on their relevance to the conversion label, facilitating effective knowledge distillation.

First, we discretize all continuous privileged signals into binary discrete features using scalar thresholds. These thresholds are determined using a simple correlation-based heuristic. Specifically, for the two data sets divided by a given threshold, we calculate the Pearson correlation between the privileged feature and the conversion label, and then optimize the threshold by maximizing the absolute difference between the two correlations. For a total set of N ranked instances, the index of the threshold point is given by:

$$\begin{aligned} t &= \arg\max_{1<k<N} |r(1, k-1) - r(k, N)| \\ \text{s.t.} \quad r(m,n) &= \frac{\sum_{i=m}^{n}(z_i - \bar{z})(y_i - \bar{y})}{\sqrt{\sum_{i=m}^{n}(z_i - \bar{z})^2}\sqrt{\sum_{i=m}^{n}(y_i - \bar{y})^2}} \end{aligned} \quad (8)$$

Here, z_i denotes the (privileged) feature value of instance i, $\bar{z}$ and $\bar{y}$ denote the mean of the feature and conversion label. $r(m, n)$ is the Pearson correlation between the feature and the label, calculated on the instances from the interval $[m, n]$. The final threshold is z_t. All instances with $z \geq z_t$ are assigned a discrete feature value of 1, and instances with $z < z_t$ are assigned a feature value of 0.

After discretizing the continuous features, we rank all features by their mutual information with the conversion label:

$$I(z, y) = \sum_{z} \sum_{y \in \{0,1\}} p(z, y) \log \frac{p(z, y)}{p(z)p(y)} \quad (9)$$

Here, we note that the thresholds and mutual information mentioned above are calculated using logged online data of our short video platform from January to March 2024.

Finally, we manually remove the top 2 most relevant features with exceptionally high mutual information and select the top 12 from the remaining features as the privileged features. Among these 12 features, we combine "tag", "like", and "share" into a single feature using logical "or" operations due to the large sparsity of each. This leaves us with 10 final features, including 7 discrete features and 3 (discretized) continuous features. Empirically, we find that adding more privileged features to the teacher does not bring an additional increase to the teacher's performance, so we stick with the chosen 10. A detailed description of these chosen features is provided in Table 1.

Table 2: Offline comparison of different components of HA-PFD in training (September -> October).

Method	AUC	Teacher AUC	Label avg	Pred avg	Teacher pred avg	ECE	MCE
No modeling	0.92826	-	0.12042	0.10020 (-0.02022)	-	0.05429	0.39666
MTL	0.92957	-		0.09995 (-0.02047)	-	0.05544	0.38522
KD	0.92954	0.97082		0.09669 (-0.02373)	0.09170 (-0.02872)	0.05893	0.41704
Focal V1	0.92919	0.97031		0.11713 (-0.00329)	0.09216 (-0.02826)	0.03743	0.18124
Focal V2	0.92932	0.97088		0.11665 (-0.00377)	0.09172 (-0.02870)	0.03769	0.19011
Layer align	0.92968	0.97063		0.10009 (-0.02033)	0.09166 (-0.02876)	0.05386	0.39618
HA-PFD V1	0.93011	0.97091		0.11671 (-0.00371)	0.09139 (-0.02903)	0.03717	0.18075
HA-PFD V2	**0.93022**	0.97080		0.11614 (-0.00428)	0.09148 (-0.02894)	0.03782	0.18957

Table 3: Offline comparison of different components of HA-PFD in testing (November -> December).

Method	AUC	Teacher AUC	Label avg	Pred avg	Teacher pred avg	ECE	MCE
No modeling	0.88363	-	0.13168	0.10501 (-0.02667)	-	0.07688	0.42863
MTL	0.88273	-		0.11144 (-0.02024)	-	0.08093	0.42832
KD	0.88694	0.96587		0.11646 (-0.01522)	0.10687 (-0.02481)	0.08335	0.44229
Focal V1	0.88597	0.96503		0.13277 (0.00109)	0.10800 (-0.02368)	0.05375	0.21560
Focal V2	0.88673	0.96537		0.13291 (0.00123)	0.10788 (-0.0238)	0.05634	0.21957
Layer align	0.88671	0.96555		0.13202 (0.00034)	0.10420 (-0.02748)	0.07947	0.43014
HA-PFD V1	0.88739	0.96688		0.13021 (-0.00147)	0.10414 (-0.02754)	0.05296	0.22182
HA-PFD V2	**0.88772**	0.96640		0.13106 (-0.00062)	0.10792 (-0.02376)	0.05646	0.22248

4 Experiments

To demonstrate the effectiveness of HA-PFD, we conduct extensive offline evaluation on large-scale real-world datasets, where our method exhibits state-of-the-art ranking performance and superior model calibration. Additionally, we perform online evaluation in which our model is deployed to serve live users on Douyin, a short video platform with billions of users.

4.1 Metrics and baselines

We compare our HA-PFD with MTL [40], the original KD [16], and other state-of-the-art KD methods, including ReviewKD [6], adversarial KD [8], and similarity-preserving KD [36]. The original KD represents the classical, yet powerful, logit-level distillation methods. The other methods represent state-of-the-art latent-level distillation methods with different design principles. ReviewKD focuses on aligning the student's latent feature information with the teacher's on an instance-by-instance basis, similar to our layer alignment; similarity-preserving KD aims to preserve the relations of the latent features across different instances, distilling knowledge in a list-wise manner; and adversarial KD leverages adversarial learning to make the student's latent features "look like" the teacher's.

For all KD models, we use the same three-layer DNN architecture for the student and the teacher. The DNN has hidden dimensions of 256 and 128 for the first and second layers, respectively, with SmeLU activation functions [32]. The output layer uses a sigmoid activation function for CVR prediction. We use the Adam optimizer [20] with a learning rate of 0.025 and a batch size of 512. The hyperparameter γ in the focal loss is tuned to 1.0 for optimal model calibration, and λ in $\mathcal{L}_{HA-PFD}$ is tuned to 10.0 for the best ranking performance. For MTL, we use the same DNN architecture, optimizer, and learning rate for the main tower and the auxiliary towers. We use the same set of privileged features for all models.

For offline evaluation, we take the area under the curve (AUC) of the receiver operating characteristic (ROC) curve [2] as the primary metric for the ranking performance of the models. We also report the average model prediction score and average of labels, as well as expected calibration error (ECE) and max calibration error (MCE)[12] for evaluating model calibration.

For online evaluation, we report the advertising cost, the number of sends, shows, clicks, and conversions, as well as the advertiser value, the average predicted CVR, and the posterior ground-truth CVR. Briefly, the predicted CVR is used in the bidding and charging of advertisements. In the bidding process, the ads are ranked by their effective cost per mille (ECPM), representing how much it costs the advertiser for 1000 impressions. Given the predicted CTR, CVR, and the advertiser's bid for a certain ad, ECPM is calculated as:

$$ECPM = p_{CTR} \times p_{CVR} \times bid \times 1000 \tag{10}$$

The ads with the highest ECPMs win the auction and are sent to the users. The advertising system charges the advertiser $\frac{\text{ECPM}}{1000}$ for every ad sent. The sum of these charges constitutes the advertising cost. The advertiser value is then calculated by summing the bids of the ads that actually converted. Overall, for online evaluation, the advertising cost represents the revenue of the advertising system for providing the advertising services, while the advertiser value represents the return the advertiser receives from paying for the

Table 4: Offline comparison of different KD methods in training (September -> October).

Method	AUC	Teacher AUC	Label avg	Pred avg	Teacher pred avg	ECE	MCE
Review	0.92972	0.97004	0.12042	0.09997 (-0.02045)	0.09179 (-0.02863)	0.05474	0.40023
Similarity	0.92925	0.97026		0.10016 (-0.02026)	0.09206 (-0.02836)	0.05318	0.38836
Adversarial	0.92897	0.96995		0.10029 (-0.02013)	0.09223 (-0.02819)	0.05832	0.41021
HA-PFD V2	**0.93022**	0.97080		0.11614 (-0.00428)	0.09148 (-0.02894)	0.03782	0.18957

Table 5: Offline comparison of different KD methods in testing (November -> December).

Method	AUC	Teacher AUC	Label avg	Pred avg	Teacher pred avg	ECE	MCE
Review	0.88698	0.96624	0.13168	0.13031 (-0.00137)	0.10426 (-0.02742)	0.08027	0.42880
Similarity	0.88515	0.96507		0.12653 (-0.00515)	0.10348 (-0.02820)	0.07804	0.42396
Adversarial	0.88422	0.96521		0.12928 (-0.00240)	0.10554 (-0.02614)	0.08160	0.43331
HA-PFD V2	**0.88772**	0.96640		0.13106 (-0.00062)	0.10792 (-0.02376)	0.05646	0.22248

advertisements. For the CVR model, the primary objective is to maximize the advertiser value while maintaining the advertising cost.

4.2 Offline evaluation

We perform offline evaluation on logged data from our short video platform from 2024. The dataset contains 1.68 billion instances with an average CVR of 0.12309. We use data from January to October as the training set and data from November to December as the testing set, resulting in a training set of 1.39 billion instances and a testing set of 0.29 billion instances. Here, we report the training metrics for September and October and the testing metrics for November and December. However, it is important to note that in practice, CVR models are trained online, meaning the model is constantly updated with recent data [17]. Therefore, the training metrics are more indicative of the model's online performance, and we report the testing results for better insights.

We start by verifying the design of HA-PFD. We compare the performance of HA-PFD with the baseline model without privileged feature modeling, the MTL model, the original KD, and different components of HA-PFD, including latent-level distillation with layer alignment and the two versions of logit-level distillation with focal loss. The results are shown in Table 2 and Table 3. Compared to the no-modeling baseline, all other models show an improvement in training AUC, demonstrating the potential of boosting CVR prediction by utilizing privileged features. Surprisingly, however, the MTL model experiences a considerable drop in testing AUC, indicating the potential instability of the MTL approach.

Notably, the no-modeling baseline suffers from severe model miscalibration. With an ECE of 0.05429 and a whopping MCE of 0.39666, the average CVR prediction is misaligned with the label average by a large margin. This is due to the imbalanced nature of the CVR prediction problem, causing the CVR model to easily overfit the CE task loss on negative instances while overlooking the positive instances, deviating prediction score. While MTL does improve the ranking ability of the model, it does not address the miscalibration issue. Specifically, although KD achieves competitive AUC, it exacerbates model miscalibration, indicating that the improved ranking performance is primarily on easily predicted negative instances. On the other hand, although the two versions of focal loss do not perform as well as MTL and KD in terms of AUC, they demonstrate significant improvement in model calibration with a considerable drop in MCE, indicating better concentration on positive instances and reduced overconfidence on negative instances.

The latent-level distillation with layer alignment also shows a significant lift in training AUC, indicating the effectiveness of latent-level distillation. Finally, we combine latent-level distillation with focal logit-level distillation, yielding the HA-PFD model. This allows us to further boost the training AUC while maintaining model calibration and stable performance. Notably, HA-PFD V2 shows the best performance in both training and testing AUC, indicating the effectiveness of our proposed method.

We further compare HA-PFD with other state-of-the-art distillation methods, including ReviewKD, similarity-preserving KD, and adversarial KD. The results are shown in Table 4 and Table 5. ReviewKD is methodologically similar to our layer alignment, wherein each student layer learns from all previous layers of the teacher rather than only the corresponding layer. While it offers slightly better distillation performance than layer alignment, we choose the latter for its simplicity. Overall, HA-PFD V2 outperforms all other KD methods in both AUC and model calibration by combining logit-level distillation with focal loss and latent-level distillation with layer alignment, demonstrating the state-of-the-art performance of our method. Additional offline results are provided in the Appendix.

4.3 Online evaluation

For online evaluation, we deployed our HA-PFD method to the CVR model of online ads on Douyin, a large-scale short video platform with billions of users. We conducted three sets of online A/B experiments from March 2024 to April 2024, with different amounts of user traffic, comparing our model with the former online MTL

Table 6: Online results.

Exp.	Model	Click	Convert	Cost	Advertiser Value	CVR	Predicted CVR
1	MTL	1042103	115048	-	-	0.1104	0.1010 (-8.514%)
	Layer align	1002979 (+3.754%)	116145 (+0.954%)	-0.594% (p=18.59%)	+1.908% (p=26.72%)	0.1158	0.1045 (-9.758%)
2	MTL	597825	71260	-	-	0.1192	0.1039 (-12.836%)
	Focal V1	589290 (-1.428%)	73013 (+2.460%)	+2.537% (p=0.14%)	+2.723% (p=9.74%)	0.1239	0.1258 (+1.533%)
3	MTL	796340	90394	-	-	0.1135	0.1037 (-8.634%)
	HA-PFD V2	751122 (-5.678%)	91683 (+1.426%)	+1.500% (p=1.23%)	+3.739% (p=4.89%)	0.1221	0.1209 (-0.983%)

Table 7: Online experimental settings.

Exp.	Date	Duration	Traffic
1	2024/03/15 - 2024/03/23	7days 22hours	17.54%
2	2024/04/04 - 2024/04/11	6days 23hours	10.00%
3	2024/04/21 - 2024/04/29	7days 7hours	12.50%

model. For fair comparison, each model in an experiment is assigned an equal percentage of user traffic and uses the same CTR prediction model and the same bidding system. While the baseline in these experiments is the same MTL model, note that these experiments are conducted on different dates, with different durations, and with different amounts of user traffic. As the online advertising environment is highly dynamic, these differences inevitably lead to variance in the results. The experimental settings are given in Table 7. We report the main online metrics as well as their relative differences in Table 6. For the cost and advertiser value, we only report the relative values and the statistical significance (p-value) due to confidentiality restrictions. For the predicted CVR, we mark the relative error to the posterior CVR.

From the results of Experiment 1, layer alignment achieves a 0.954% lift in the number of conversions and a 1.908% lift in advertiser value compared to the MTL baseline, demonstrating the effectiveness of latent-level distillation. However, latent-level distillation does not address the problem of miscalibration in the model prediction score. Consequently, the predicted CVR of both models is considerably lower than the actual CVR. On the other hand, the advertising cost is slightly harmed, limiting the potential for further increases in advertiser value. In Experiment 2, Focal V1 significantly increased the predicted CVR, thereby improving model calibration performance. The increase in the predicted CVR score naturally lifts the number of sends and shows, and in turn, the advertising cost. Subsequently, the model achieves a 2.460% lift in the number of conversions and a 2.723% lift in advertiser value with a p-value of 9.74%.

Finally, in Experiment 3, we combine latent-level distillation with focal logit-level distillation, forming our HA-PFD V2. Notably, the model maintains a calibrated predicted CVR score with the introduction of focal loss. Specifically, the model achieves a 1.426% lift in the number of conversions and a 3.739% lift with a sub-5% p-value in advertiser value compared to the MTL baseline, yielding a significant improvement in advertising performance. Additionally, the posterior CVR of our experimental model is higher than the baseline in all three sets of experiments, indicating the superiority of PFD compared to MTL.

Overall, the HA-PFD model achieves a significant lift in advertiser value and greatly improves model calibration by aligning the model prediction with the ground-truth CVR, demonstrating the novelty of our model over MTL. The results of Experiment 3 naturally meet the launch criteria of our short video platform. Consequently, we are able to deploy HA-PFD as the new baseline model for online serving on our short video platform.

5 Related work

Knowledge distillation methods have primarily been developed for model compression [16] and domain transfer [42] by supervising model learning with a powerful teacher. Current KD works can roughly be categorized as logit-based and latent-based [10].

Logit-based distillation, also known as response-based distillation, involves the student directly copying the teacher's predictions [1, 4, 16, 27, 49]. This idea was first proposed by Hinton et al. [16] for image classification, where the teacher's predictions are used as soft targets or "dark knowledge" to guide the student. Decoupled KD [49] proposes to separate the target-class and non-target-class and tune the weights of each part separately during the distillation process. However, logit-based methods fail to incorporate latent-level supervision for better knowledge transfer.

Latent-based methods leverage intermediate features between teacher layers for better representation learning [3, 5, 14, 18, 19, 22, 25, 37, 39]. FitNet [31] first introduced this idea, providing "hints" for the student by directly matching the corresponding latent features using L2 loss. A number of subsequent works have designed feature maps in the latent space for efficient knowledge transfer in the same layer-to-layer manner. AT [46] designs an attention map as the embodiment of teacher knowledge. FSP [45] performs distillation through the Gramian matrix that represents a flow of the solution process. In ReviewKD [6], each student layer learns from its corresponding teacher layer as well as all previous teacher layers. In adversarial KD [8], adversarial learning is used to make student features "indistinguishable" from the teacher's.

Another line of work extends latent-based distillation to a pairwise or list-wise manner, focusing on relationships between different instances [30, 35, 36]. In Similarity-preserving KD [36], the pairwise similarities of instances in a batch are preserved by the student. In CRD [35], the relationship between instances is learned through a contrastive-based objective that maximizes the mutual information between the teacher and student representations.

In recent studies, KD has been applied to modeling privileged features in recommender systems, yielding privileged features distillation (PFD) [11, 21, 38, 41, 43]. Xu et al. [41] first introduced the concept of PFD by incorporating KD for privileged feature modeling in Taobao recommendations. Gui [11] later proposed a listwise distillation loss for preserving the teacher's ranking ability without compromising model calibration. Yang et al. [43] conducted extensive experiments on PFD and provided valuable insights on the design and tuning of PFD. In particular, they pointed out that the student's performance decreases as the privileged features become overly informative of the ground-truth label. This conclusion plays a vital role in our feature engineering for HA-PFD.

6 Discussion

In this work, we delve into distilling privileged signals for enhancing CVR prediction in real-world advertising systems. We identify three critical and unique challenges when implementing PFD techniques for online advertising scenarios. In response to these challenges, we propose HA-PFD, which includes two focal-style distillation losses for hardness-aware knowledge transfer that boost student performance on hard instances and improve model calibration simultaneously. We also introduce latent-level distillation to PFD for the first time to facilitate the student's representation learning. Both offline and online experiments are conducted to demonstrate the effectiveness of HA-PFD.

Here, we discuss the limitations of HA-PFD and potential directions for future work. Primarily, HA-PFD achieves hardness awareness and model calibration through logit-level distillation with focal-style losses, which introduces an additional hyperparameter γ. Our empirical results indicate that the value of γ significantly impacts the final predicted CVR scores. Given that the advertising system operates in a highly dynamic environment, the optimal value of γ may vary over time. Therefore, future work should focus on developing a more adaptive focal loss that can automatically adjust the value of γ based on the changing data distribution. Additionally, the latent-level distillation in HA-PFD is performed using a simple layer-aligning approach. We believe that more effective methods can be developed for better knowledge transfer in PFD. Furthermore, we are interested in exploring PFD with heterogeneous teacher and student models to determine if more complex teacher models can facilitate improved student performance.

Overall, HA-PFD serves as a stepping stone for better leveraging privileged information in machine learning for industrial systems, opening up new possibilities for applying PFD to various scenarios beyond general recommendation systems. We believe that the insights and techniques developed in this work will inspire future research in the field of privileged features distillation and general knowledge distillation.

Acknowledgments

The authors would like to thank Yixiang Mu for his support and valuable discussions.

References

[1] Umar Asif, Jianbin Tang, and Stefan Harrer. 2020. Ensemble knowledge distillation for learning improved and efficient networks. In *ECAI 2020*. IOS Press, 953–960.
[2] Andrew P Bradley. 1997. The use of the area under the ROC curve in the evaluation of machine learning algorithms. *Pattern recognition* 30, 7 (1997), 1145–1159.
[3] Defang Chen, Jian-Ping Mei, Can Wang, Yan Feng, and Chun Chen. 2020. Online knowledge distillation with diverse peers. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 34. 3430–3437.
[4] Defang Chen, Jian-Ping Mei, Hailin Zhang, Can Wang, Yan Feng, and Chun Chen. 2022. Knowledge distillation with the reused teacher classifier. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 11933–11942.
[5] Defang Chen, Jian-Ping Mei, Yuan Zhang, Can Wang, Zhe Wang, Yan Feng, and Chun Chen. 2021. Cross-layer distillation with semantic calibration. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 35. 7028–7036.
[6] Pengguang Chen, Shu Liu, Hengshuang Zhao, and Jiaya Jia. 2021. Distilling knowledge via knowledge review. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 5008–5017.
[7] 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 1st workshop on deep learning for recommender systems*. 7–10.
[8] Inseop Chung, SeongUk Park, Jangho Kim, and Nojun Kwak. 2020. Feature-map-level online adversarial knowledge distillation. In *International Conference on Machine Learning*. PMLR, 2006–2015.
[9] Paul Covington, Jay Adams, and Emre Sargin. 2016. Deep neural networks for youtube recommendations. In *Proceedings of the 10th ACM conference on recommender systems*. 191–198.
[10] Jianping Gou, Baosheng Yu, Stephen J Maybank, and Dacheng Tao. 2021. Knowledge distillation: A survey. *International Journal of Computer Vision* 129, 6 (2021), 1789–1819.
[11] Xiaoqiang Gui, Yueyao Cheng, Xiang-Rong Sheng, Yunfeng Zhao, Guoxian Yu, Shuguang Han, Yuning Jiang, Jian Xu, and Bo Zheng. 2024. Calibration-compatible Listwise Distillation of Privileged Features for CTR Prediction. In *Proceedings of the 17th ACM International Conference on Web Search and Data Mining*. 247–256.
[12] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. 2017. On calibration of modern neural networks. In *International conference on machine learning*. PMLR, 1321–1330.
[13] Huifeng Guo, Ruiming Tang, Yunming Ye, Zhenguo Li, and Xiuqiang He. 2017. DeepFM: A Factorization-Machine based Neural Network for CTR Prediction. arXiv:1703.04247 [cs.IR] https://arxiv.org/abs/1703.04247
[14] Qiushan Guo, Xinjiang Wang, Yichao Wu, Zhipeng Yu, Ding Liang, Xiaolin Hu, and Ping Luo. 2020. Online knowledge distillation via collaborative learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 11020–11029.
[15] Byeongho Heo, Jeesoo Kim, Sangdoo Yun, Hyojin Park, Nojun Kwak, and Jin Young Choi. 2019. A comprehensive overhaul of feature distillation. In *Proceedings of the IEEE/CVF international conference on computer vision*. 1921–1930.
[16] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the Knowledge in a Neural Network. arXiv:1503.02531 [stat.ML] https://arxiv.org/abs/1503.02531
[17] Steven CH Hoi, Doyen Sahoo, Jing Lu, and Peilin Zhao. 2021. Online learning: A comprehensive survey. *Neurocomputing* 459 (2021), 249–289.
[18] Mingi Ji, Byeongho Heo, and Sungrae Park. 2021. Show, attend and distill: Knowledge distillation via attention-based feature matching. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 35. 7945–7952.
[19] Zijian Kang, Peizhen Zhang, Xiangyu Zhang, Jian Sun, and Nanning Zheng. 2021. Instance-conditional knowledge distillation for object detection. *Advances in Neural Information Processing Systems* 34 (2021), 16468–16480.
[20] Diederik P. Kingma and Jimmy Ba. 2017. Adam: A Method for Stochastic Optimization. arXiv:1412.6980 [cs.LG] https://arxiv.org/abs/1412.6980
[21] Ang Li, Jian Hu, Ke Ding, Xiaolu Zhang, Jun Zhou, Yong He, and Xu Min. 2023. Uncertainty-based Heterogeneous Privileged Knowledge Distillation for Recommendation System. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2471–2475.
[22] Shaojie Li, Mingbao Lin, Yan Wang, Yongjian Wu, Yonghong Tian, Ling Shao, and Rongrong Ji. 2022. Distilling a powerful student model via online knowledge distillation. *IEEE transactions on neural networks and learning systems* 34, 11 (2022), 8743–8752.
[23] Wei-Hong Li and Hakan Bilen. 2020. Knowledge distillation for multi-task learning. In *European Conference on Computer Vision*. Springer, 163–176.

[24] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. 2018. Focal Loss for Dense Object Detection. arXiv:1708.02002 [cs.CV] https://arxiv.org/abs/1708.02002
[25] Dongyang Liu, Meina Kan, Shiguang Shan, and Xilin Chen. 2023. Function-Consistent Feature Distillation. arXiv:2304.11832 [cs.CV] https://arxiv.org/abs/2304.11832
[26] Matthias Minderer, Josip Djolonga, Rob Romijnders, Frances Hubis, Xiaohua Zhai, Neil Houlsby, Dustin Tran, and Mario Lucic. 2021. Revisiting the calibration of modern neural networks. *Advances in Neural Information Processing Systems* 34 (2021), 15682–15694.
[27] Seyed Iman Mirzadeh, Mehrdad Farajtabar, Ang Li, Nir Levine, Akihiro Matsukawa, and Hassan Ghasemzadeh. 2020. Improved knowledge distillation via teacher assistant. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 34. 5191–5198.
[28] Jishnu Mukhoti, Viveka Kulharia, Amartya Sanyal, Stuart Golodetz, Philip Torr, and Puneet Dokania. 2020. Calibrating deep neural networks using focal loss. *Advances in Neural Information Processing Systems* 33 (2020), 15288–15299.
[29] Wonpyo Park, Dongju Kim, Yan Lu, and Minsu Cho. 2019. Relational knowledge distillation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 3967–3976.
[30] Baoyun Peng, Xiao Jin, Jiaheng Liu, Dongsheng Li, Yichao Wu, Yu Liu, Shunfeng Zhou, and Zhaoning Zhang. 2019. Correlation congruence for knowledge distillation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 5007–5016.
[31] Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. 2015. FitNets: Hints for Thin Deep Nets. arXiv:1412.6550 [cs.LG] https://arxiv.org/abs/1412.6550
[32] Gil I. Shamir and Dong Lin. 2022. Real World Large Scale Recommendation Systems Reproducibility and Smooth Activations. arXiv:2202.06499 [cs.LG] https://arxiv.org/abs/2202.06499
[33] Xiang-Rong Sheng, Jingyue Gao, Yueyao Cheng, Siran Yang, Shuguang Han, Hongbo Deng, Yuning Jiang, Jian Xu, and Bo Zheng. 2023. Joint optimization of ranking and calibration with contextualized hybrid model. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 4813–4822.
[34] Linwei Tao, Minjing Dong, and Chang Xu. 2023. Dual focal loss for calibration. In *International Conference on Machine Learning*. PMLR, 33833–33849.
[35] Yonglong Tian, Dilip Krishnan, and Phillip Isola. 2022. Contrastive Representation Distillation. arXiv:1910.10699 [cs.LG] https://arxiv.org/abs/1910.10699
[36] Frederick Tung and Greg Mori. 2019. Similarity-preserving knowledge distillation. In *Proceedings of the IEEE/CVF international conference on computer vision*. 1365–1374.
[37] Can Wang, Defang Chen, Jian-Ping Mei, Yuan Zhang, Yan Feng, and Chun Chen. 2022. SemCKD: Semantic calibration for cross-layer knowledge distillation. *IEEE Transactions on Knowledge and Data Engineering* 35, 6 (2022), 6305–6319.
[38] Chenyang Wang, Zhefan Wang, Yankai Liu, Yang Ge, Weizhi Ma, Min Zhang, Yiqun Liu, Junlan Feng, Chao Deng, and Shaoping Ma. 2022. Target interest distillation for multi-interest recommendation. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*. 2007–2016.
[39] Guo-Hua Wang, Yifan Ge, and Jianxin Wu. 2021. Distilling knowledge by mimicking features. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 44, 11 (2021), 8183–8195.
[40] Yuhao Wang, Ha Tsz Lam, Yi Wong, Ziru Liu, Xiangyu Zhao, Yichao Wang, Bo Chen, Huifeng Guo, and Ruiming Tang. 2023. Multi-Task Deep Recommender Systems: A Survey. arXiv:2302.03525 [cs.IR] https://arxiv.org/abs/2302.03525
[41] Chen Xu, Quan Li, Junfeng Ge, Jinyang Gao, Xiaoyong Yang, Changhua Pei, Fei Sun, Jian Wu, Hanxiao Sun, and Wenwu Ou. 2020. Privileged features distillation at taobao recommendations. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2590–2598.
[42] Chenxiao Yang, Junwei Pan, Xiaofeng Gao, Tingyu Jiang, Dapeng Liu, and Guihai Chen. 2022. Cross-task knowledge distillation in multi-task recommendation. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 36. 4318–4326.
[43] Shuo Yang, Sujay Sanghavi, Holakou Rahmanian, Jan Bakus, and Vishwanathan SVN. 2022. Toward understanding privileged features distillation in learning-to-rank. *Advances in Neural Information Processing Systems* 35 (2022), 26658–26670.
[44] Yanwu Yang and Panyu Zhai. 2022. Click-through rate prediction in online advertising: A literature review. *Information Processing & Management* 59, 2 (2022), 102853.
[45] Junho Yim, Donggyu Joo, Jihoon Bae, and Junmo Kim. 2017. A gift from knowledge distillation: Fast optimization, network minimization and transfer learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 4133–4141.
[46] Sergey Zagoruyko and Nikos Komodakis. 2017. Paying More Attention to Attention: Improving the Performance of Convolutional Neural Networks via Attention Transfer. arXiv:1612.03928 [cs.CV] https://arxiv.org/abs/1612.03928
[47] Ying Zhang, Tao Xiang, Timothy M Hospedales, and Huchuan Lu. 2018. Deep mutual learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 4320–4328.
[48] Yu Zhang and Qiang Yang. 2021. A survey on multi-task learning. *IEEE transactions on knowledge and data engineering* 34, 12 (2021), 5586–5609.
[49] Borui Zhao, Quan Cui, Renjie Song, Yiyu Qiu, and Jiajun Liang. 2022. Decoupled knowledge distillation. In *Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition*. 11953–11962.
[50] 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 *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*. 1059–1068.

A Implementation details

We implement our HA-PFD, as well as all competing baseline models, using TensorFlow 1.15. All models are updated using the Adam optimizer, with a learning rate of 0.0025, a weight decay of 0.001, and a gradient clip of 600. All model parameters are randomly initialized from a truncated normal distribution with a mean of 0 and a standard deviation of 0.1. The total number of regular features for all models is 647. All features, including both regular and privileged, are first transformed into 16-dimensional dense embeddings through hash mapping before being fed into the neural networks as input. For all offline evaluation, we perform the training with 3 seeds and report the mean values of the metrics. All model training is conducted on NVIDIA A100 GPUs, with distributed data parallel training. All model inferences for online serving are performed on NVIDIA A40 GPUs.

For ReviewKD, the weight of the feature-level distillation loss is set to 10, the same as our layer aligning strategy. However, for similarity-preserving KD, the weight of the feature-level distillation loss is tuned to 100. For adversarial KD, we build two discriminators, each as a three-layer DNN with hidden dimensions of 128 and 64 for the two layers of teacher latent features, respectively. The discriminators are trained to predict 1 for the teacher latent features and 0 for the student latent features, and are updated with the same Adam optimizer as the other networks. The weight of the discriminator losses is set to 1, and the discriminators are first warmed up for 5 million batches before performing PFD. The student is then trained to optimize its two layers of latent features for high discriminator scores, with the weight of the losses set to 10. For all PFD methods, the teacher and student share the same set of feature embeddings.

B Additional offline results

We perform additional offline evaluation that compare the performance of combining ReviewKD, similarity-preserving KD, and adversarial KD with the original KD, as well as our Focal V1 and Focal V2. The training and testing results are demonstrated in Table 8 and Table 9.

Evidently, ReviewKD and our layer alignment yield the best AUC performance when combined with the original KD, worsening model miscalibration. This again demonstrates the effectiveness of knowledge transfer through KD and the negative impact of PFD without hardness awareness. In contrast, when combined with our proposed two versions of focal-style distillation loss, the model shows a shift of learning concentration from easily predicted negative instances to positive instances, resulting in superior model calibration.

We also note that ReviewKD and layer alignment achieve similar performance in our offline experiments. This is due to the similarity in the design of both methods. In ReviewKD, the student latent

Table 8: Additional offline evaluation in training (Sepetember -> October).

Method	AUC	Teacher AUC	Label avg	Pred avg	Teacher pred avg	ECE	MCE
KD + Layer align	**0.93031**	0.97077	0.12042	0.09655 (-0.02387)	0.09145 (-0.02897)	0.06015	0.41611
KD + Review	0.93029	0.97100		0.09638 (-0.02404)	0.09131 (-0.02911)	0.06100	0.41468
Focal V1 + Review	0.93014	0.97048		0.11689 (-0.00353)	0.09168 (-0.02874)	0.03744	0.18649
Focal V2 + Review	0.93018	0.97036		0.11575 (-0.00467)	0.09172 (-0.0287)	0.03760	0.18808
KD + Similarity	0.92951	0.97054		0.09661 (-0.02381)	0.09151 (-0.02891)	0.05894	0.41503
Focal V1 + Similarity	0.92947	0.97022		0.11713 (-0.00329)	0.09193 (-0.02849)	0.03804	0.18709
Focal V2 + Similarity	0.92959	0.97040		0.11584 (-0.00458)	0.09181 (-0.02861)	0.03833	0.19068
KD + Adversarial	0.92917	0.96939		0.09852 (-0.02190)	0.09213 (-0.02829)	0.06577	0.42364
Focal V1 + Adversarial	0.92899	0.96947		0.10884 (-0.01158)	0.09221 (-0.02821)	0.03965	0.19475
Focal V2 + Adversarial	0.92901	0.96866		0.10884 (-0.01158)	0.09223 (-0.14861)	0.03991	0.19811
HA-PFD V2	0.93022	0.97080		0.11614 (-0.00428)	0.09148 (-0.02894)	0.03782	0.18957

Table 9: Additional offline evaluation in testing (November -> December).

Method	AUC	Teacher AUC	Label avg	Pred avg	Teacher pred avg	ECE	MCE
KD + Layer align	0.88741	0.96521	0.13168	0.11096 (-0.02072)	0.10849 (-0.02319)	0.08247	0.44338
KD + Review	0.88734	0.96603		0.11279 (-0.01889)	0.11016 (-0.02152)	0.08491	0.45009
Focal V1 + Review	0.88720	0.96614		0.12958 (-0.00210)	0.10978 (-0.02190)	0.05258	0.22299
Focal V2 + Review	0.88729	0.96587		0.13064 (-0.00104)	0.10877 (-0.02291)	0.05601	0.22245
KD + Similarity	0.88621	0.96496		0.10760 (-0.02408)	0.10268 (-0.02900)	0.08287	0.43971
Focal V1 + Similarity	0.88617	0.96459		0.13152 (-0.00016)	0.10569 (-0.02599)	0.05239	0.22103
Focal V2 + Similarity	0.88629	0.96501		0.13260 (0.00092)	0.10491 (-0.02677)	0.05488	0.22254
KD + Adversarial	0.88555	0.96544		0.11146 (-0.02022)	0.10720 (-0.02448)	0.08669	0.45793
Focal V1 + Adversarial	0.88415	0.96503		0.13479 (0.00311)	0.10829 (-0.02339)	0.05816	0.22935
Focal V2 + Adversarial	0.88448	0.96510		0.13010 (-0.00158)	0.10689 (-0.02479)	0.05940	0.23290
HA-PFD V2	**0.88772**	0.96640		0.13106 (-0.00062)	0.10792 (-0.02376)	0.05646	0.22248

features are not only aligned with the corresponding teacher latent features but also with all latents from previous layers. However, we adopt a simple three-layer DNN architecture for our CVR model, leaving us with only two layers of latent features. This means that ReviewKD only introduces an additional feature-level distillation loss that aligns student latent 2 with teacher latent 1, compared to layer alignment. Empirically, such a design does not bring significant improvement in AUC. Therefore, we use layer alignment for HA-PFD out of simplicity.

C Scalability in complex network settings

Latent-level knowledge distillation methods have shown reasonable scalability in related works [45]. As our layer alignment is very straightforward, it is expected to show stable scalability in more complex network settings. Here, we perform additional offline experiments of HA-PFD with deeper networks, wider networks, and heterogeneous network structures. Specifically, we consider 3 settings and compare their performance with the original HA-PFD V2:

(1) **Deeper**: increase the number of layers to 5.

(2) **Wider**: double the latent dimensions.

(3) **Wider student**: double the latent dimensions of the student.

As demonstrated in Table 10, it is evident that HA-PFD scales up to a more complex architecture with improved ranking performance. Interestingly, widening only the student yields similar performance to widening both the student and the teacher. This indicates that the bottleneck for improving student ranking performance is not the amount of knowledge held by the teacher, but rather the capacity of the student itself.

Table 10: Scalability of different network settings in training (September -> October).

Method	AUC	Teacher AUC
HA-PFD V2	0.93022	0.97080
Deeper	0.93053	0.97129
Wider	0.93076	0.97127
Wider student	0.93072	0.97121

D Comparison with post-hoc calibration

In industrial settings, post-hoc calibration is commonly a separate post-process module for CVR models. They are orthogonal to our HA-PFD and can be combined for better calibration. Therefore, in this work, we focus mainly on the calibration ability of the model itself. Nevertheless, we perform additional offline experiments with Platt scaling [12] for further insights.

As demonstrated in Table 11, the MCE of MTL is terrible due to severe underestimation of positive instances. While Platt scaling does improve ECE, the MCE barely improved due to the parameters of Platt scaling overfitting the majority of negative instances. On the other hand, HA-PFD significantly improves MCE by concentrating the student on more-difficult positive instances during knowledge distillation. This improvement has a positive impact on the ranking and bidding of advertisements.

Table 11: Comparison with post-hoc calibration methods in training (September -> October).

Method	AUC	ECE	MCE
HA-PFD V2	0.93022	0.037823	0.18957
MTL	0.92957	0.05544	0.38522
MTL + Platt scaling	0.92953	0.031341	0.36283
KD + Platt scaling on teacher	0.92951	0.04529	0.40399

E Discussion on MTL with KD

There are recent works that combine MTL with KD. While doing so does provide exciting directions for modeling and utilizing privileged features, our HA-PFD and KD for MTL differ in their primary goals. PFD aims to utilize privileged signals to distill knowledge that enhances the main task of CVR prediction, while KD for MTL aims to form a representation that can facilitate all tasks in MTL. Regardless, we perform additional offline experiments on combining MTL and KD for better insights. Here, we consider 2 settings:

(1) **KD + MTL**: naive combination of PFD and MTL (model privileged features with both methods).
(2) **KD for MTL**: proposed in [23].

As demonstrated in Table 12, KD + MTL slightly surpasses MTL, while the multi-task-oriented KD for MTL does not perform comparably on the main task.

Table 12: Comparison with MTL with KD in training (September -> October).

Method	AUC	Teacher AUC
HA-PFD V2	0.93022	0.97080
MTL	0.92957	-
KD + MTL	0.92961	0.97079
KD for MTL	0.92941	-

F Relationship with other loss functions

Recent works that propose losses to perform simultaneous LTR and calibration, like JRC [33], do not offer solutions for utilizing privileged features, and orthogonal to our work. Moreover, this line of work heavily relies on CE loss for model calibration, which performs poorly in severe imbalance problems like CVR prediction [12]. Regardless, we perform additional offline experiments with JRC for further insights. As shown in Table 13, while JRC does improve ranking performance through an LTR loss, it does not offer better calibration in imbalance scenarios.

Table 13: Comparison with other loss functions in training (September -> October).

Method	AUC	ECE	MCE
HA-PFD V2	0.93022	0.037823	0.18957
MTL	0.92957	0.05544	0.38522
MTL + JRC	0.92984	0.05492	0.38387

G Pseudocode for HA-PFD

Algorithm 1 Hardness-aware Privileged Features Distillation

Require: Teacher model θ_t, Student model θ_s, Latent mapping θ_g, Dataset $\mathcal{D}$, Optimizer opt

Initialize model parameters θ_s, θ_t, and θ_g randomly
Create data loader: $train\_loader$
Set number of epochs N, batch size B, learning rate η
for epoch $\leftarrow$ 1 to N **do**
  $f_\theta.train()$ ▷ Set model to training mode
  for batch (x, z, y) in $train\_loader$ **do**
    $opt.zero\_grad()$ ▷ Reset gradients
    $p_t, f_t \leftarrow p_t(x, z; \theta_t), f_t(x, z; \theta_t)$ ▷ Teacher prediction and latent variables
    $p_s, f_s \leftarrow p_s(x; \theta_s), f_s(x; \theta_s)$ ▷ Student prediction and latent variables
    $g \leftarrow g(f_s; \theta_g)$ ▷ Map student latent variables
    $teacher\_task\_loss \leftarrow \mathcal{L}_{task}(p_t, y)$ ▷ Teacher task loss
    $student\_task\_loss \leftarrow \mathcal{L}_{task}(p_s, y)$ ▷ Student task loss
    $distill\_loss \leftarrow \mathcal{L}_{HA-PFD}(p_t, f_t, p_s, g)$ ▷ Distillation loss of HA-PFD
    $loss \leftarrow teacher\_task\_loss + student\_task\_loss + distill\_loss$ ▷ Total loss
    $loss.backward()$ ▷ Backward pass
    $opt.step()$ ▷ Update parameters
  end for
end for
return trained student model θ_s