Task-Augmented Knowledge Distillation for Multi-Task Personalized Recommendation

Presenter : Chenxiao Yang

May 31st, 2021

Background | Multi-Task Learning (MTL)

Heterogenous User Feedback as Prediction Tasks: Click, Like, Add to Cart, etc.

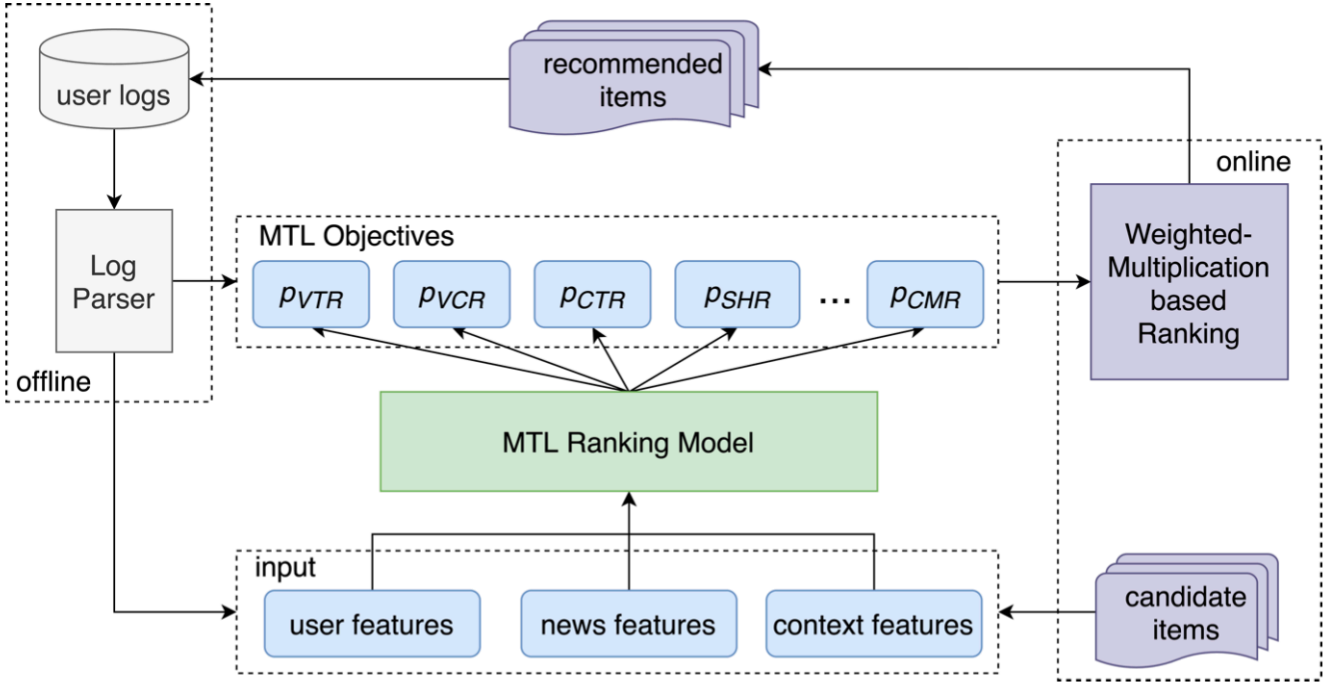
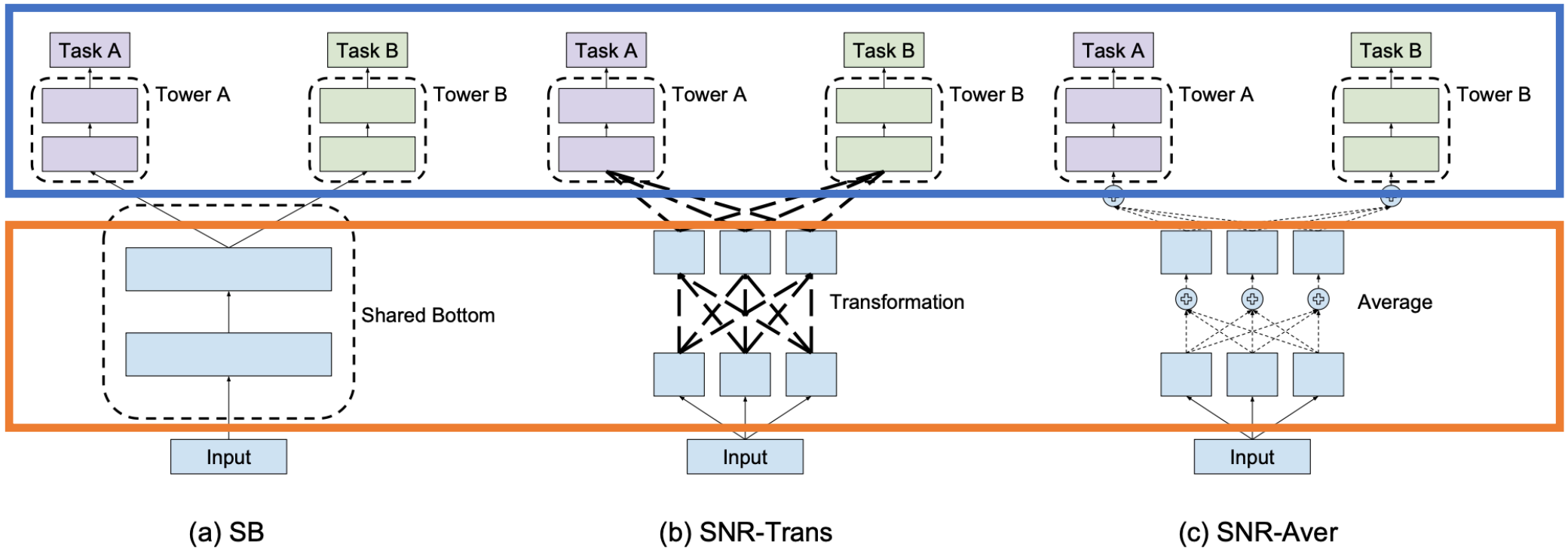


Fig. Illustration of Multi-Task Recommender Systems (from PLE [1] and PFD [3])

Background | Multi-Task Learning (MTL)



High-Level Task-Specific Layers

Low-Level Shared-Bottom Layers

Fig. A Generic Paradigm for Multi-Task Learning Models (from SNR [2])

NO Knowledge Transfer in or above **Task-Specific Layers**

Implicit Knowledge Transfer in **Shared-Bottom Layers** by Branching (b), Gating (c), etc.

Background | Multi-Task Learning (MTL)

LIMITATIONS:

1. Bottleneck on the **ability of knowledge transfer**
2. A lack of **interpretability**

   (will be discussed)
3. Unable to effectively solve **task conflicts**

   (will be discussed)

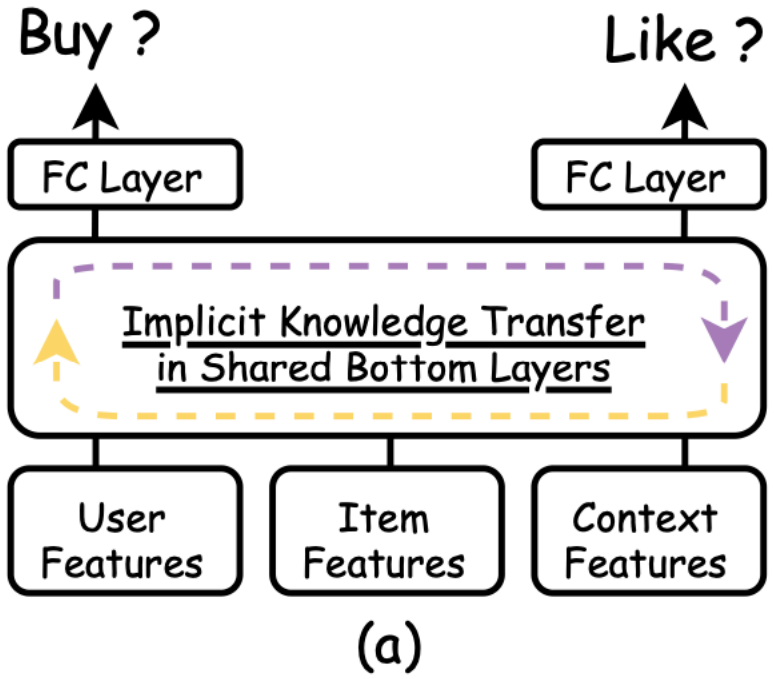


Fig. Common MTL framework

Background | Knowledge Distillation (KD)

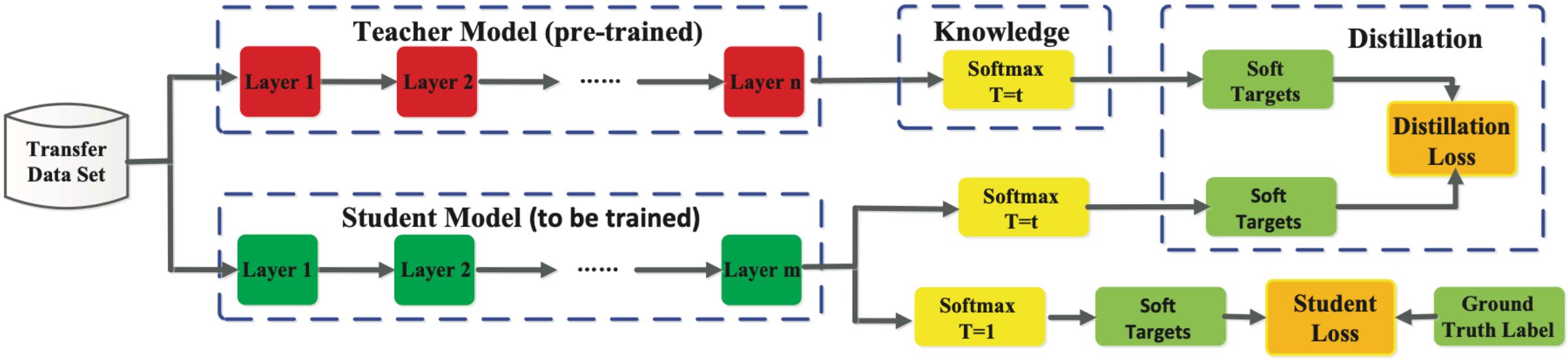


Fig. Generic Response-Based Knowledge Distillation (from a KD survey)

Hard targets	0	1	0	0
	cow	dog	cat	car
Soft targets	10^{-6}	0.9	0.1	10^{-9}

More Informative!

$$L_D(p(z_t,T),p(z_s,T)) = \sum -p_i(z_{ti},T)\log(p_i(z_{si},T)) \ ,$$

$$L_S(y,p(z_s,T)) = \mathcal{L}_{CE}(y,p(z_s,T)) \ ,$$

$$L(x,W) = \alpha * L_D(p(z_t,T),p(z_s,T)) + (1-\alpha) * L_S(y,p(z_s,T)) \ ,$$

Background | Knowledge Distillation (KD)

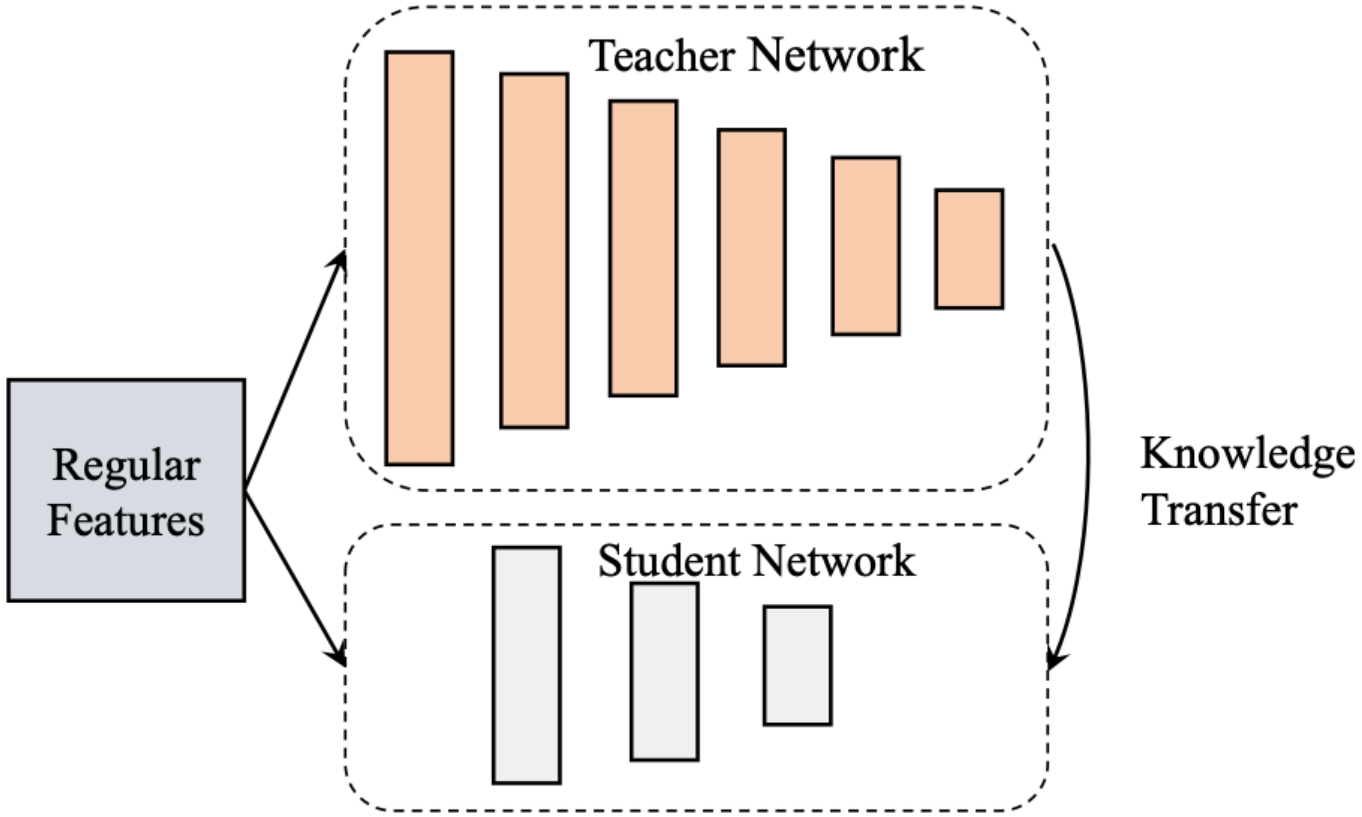


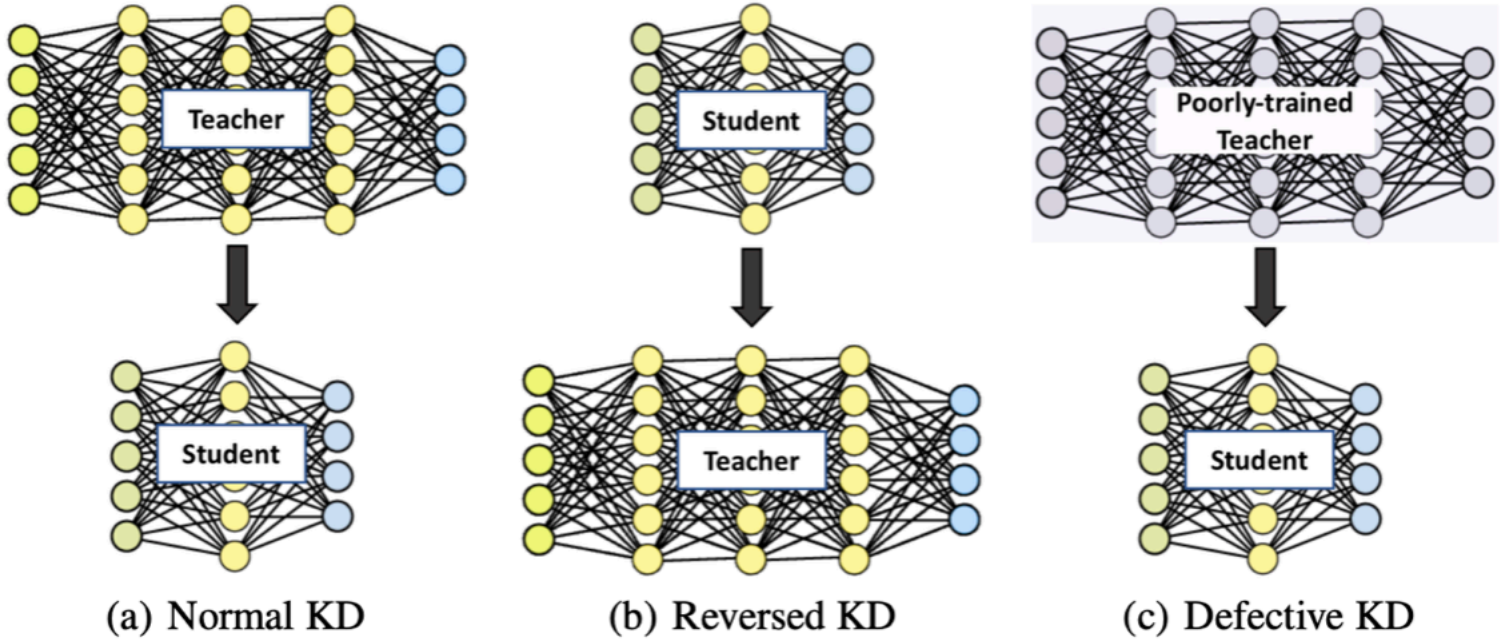
Fig. Model Compression (from PFD [3])

Model Compression:
(Conventional Usage)

Cumbersome Teacher Model
->
Light-weight Student Model

Background | Knowledge Distillation (KD)

KD as Intelligent Label Smoothing Regularization for **Knowledge Transfer** (Our usage)



Benefits of Knowledge Distillation:

1. Label smoothing
2. Example re-weighting
3. Prior knowledge of optimal output layer geometry

Fig. Exploratory Experiments (from [4])

Method | Motivation

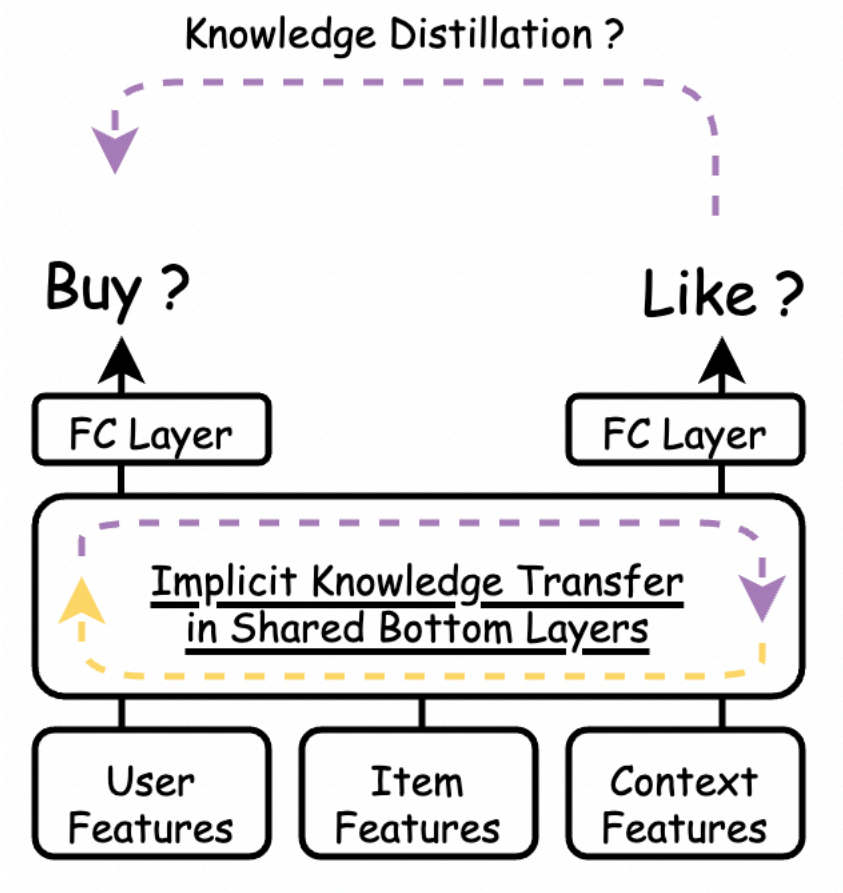


Fig. Intuitive Solution

Motivation:

Predictions of another task contains sample relation unseen to the main-task (which we want to transfer).

Intuitive Solution:

Directly Using Outputs of Another Task as Soft Labels

Problem:

TASK CONFLICTS !

Method | Motivation

Q: What kind of knowledge do we expect to transfer or elsewise to avoid?

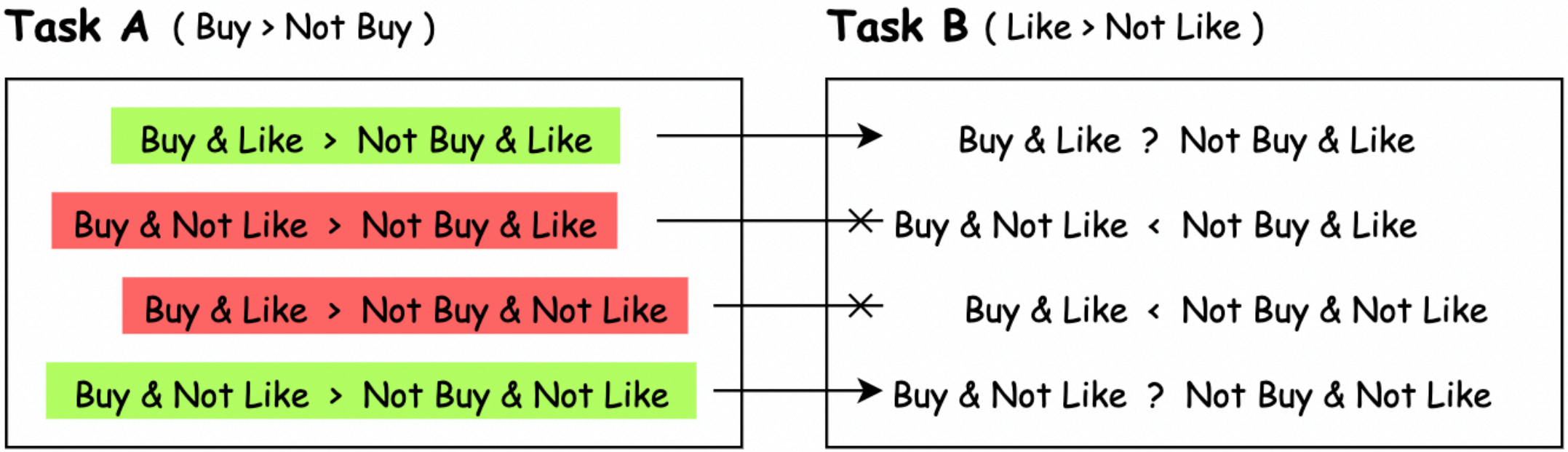


Fig. Knowledge Disentanglement

Solution: To Disentangle Positive and Negative Knowledge

ONLY transfer Positive Knowledge to Avoid Task Conflicts in Essence

Method | Motivation

UNCONFLICTED Fine-Grained Ranking

= Coarse-Grained Ranking + Auxiliary Ranking (Positive Knowledge)

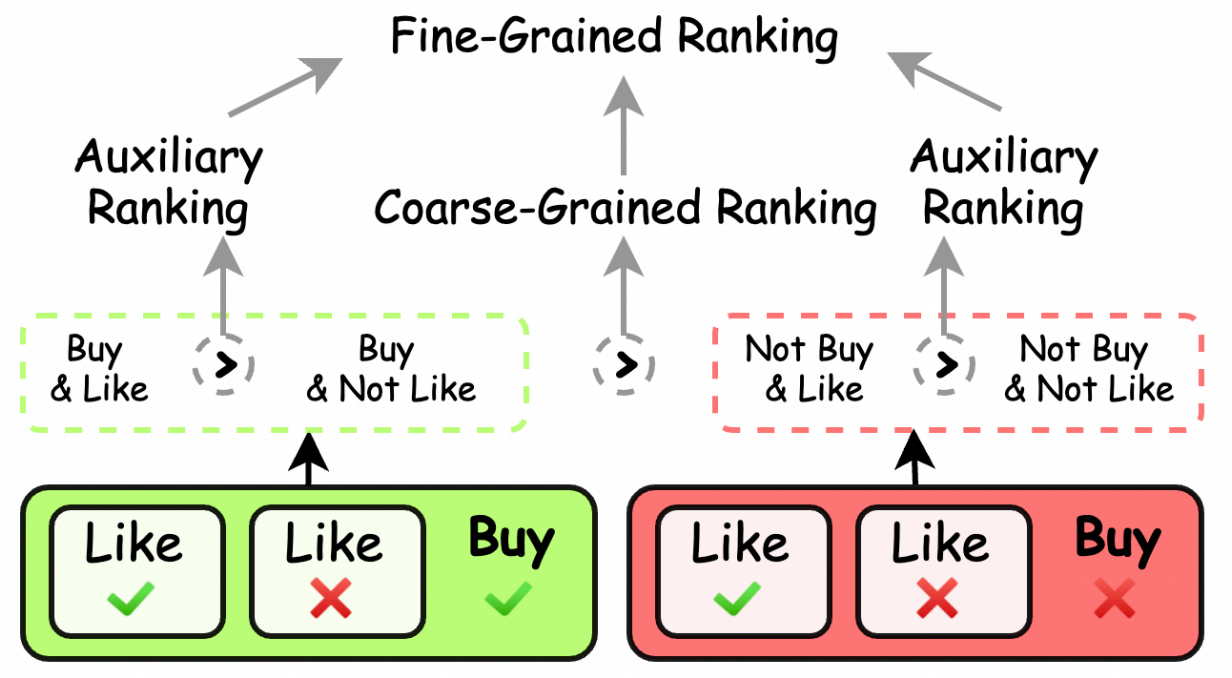


Fig. Unconflicted Fine-Grained Ranking Knowledge

Coarse-Grained Ranking

1. Only contain knowledge about the main task
2. Conflicted across tasks

v.s.

Fine-Grained Ranking

1. Contain across-task knowledge
2. Unconflicted with the main task

Method | Motivation

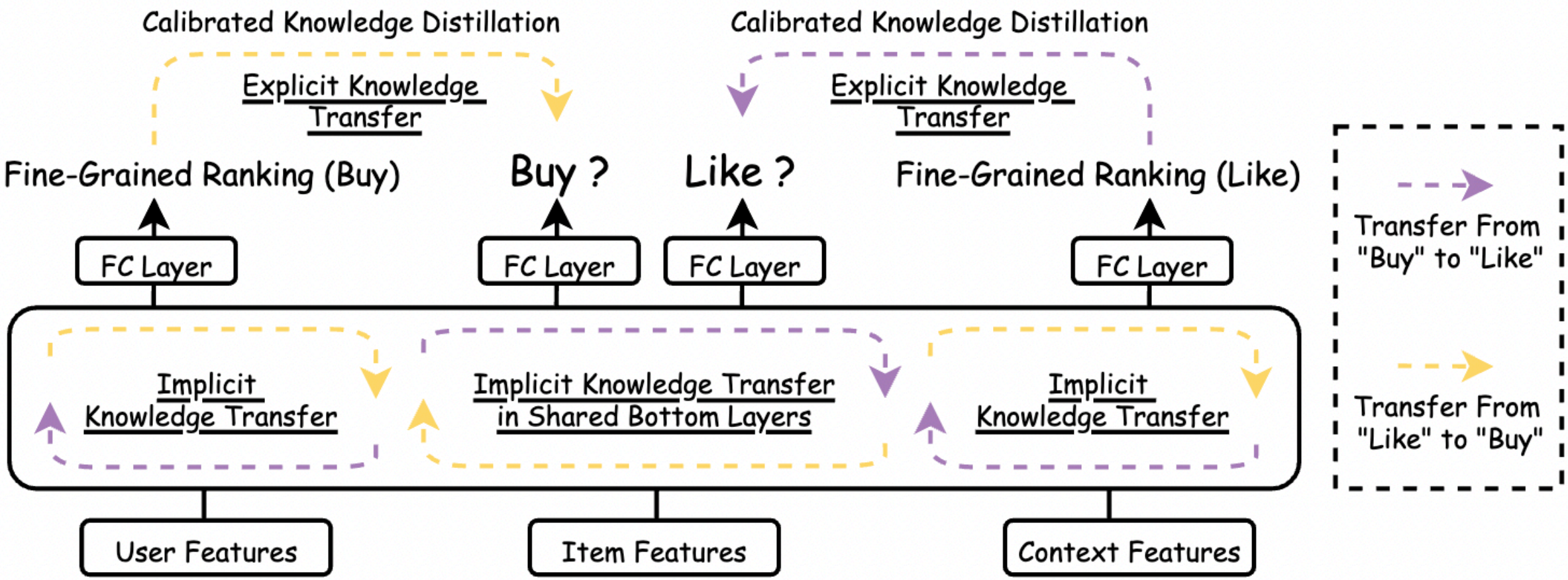


Fig. Simple Framework for TAKD based on Aforementioned Motivation

Method | Task Augmentation

Binary Cross Entropy Loss (Regression-Based): learning to fit the label

$$\mathcal{L}^A = CE(y^A, \hat{y}^A) = \sum_{\mathbf{x}_i \in \mathcal{D}} -y_i^A \ln \hat{y}_i^A - (1 - y_i^A) \ln(1 - \hat{y}_i^A)$$

$$\mathcal{L}^B = CE(y^B, \hat{y}^B) = \sum_{\mathbf{x}_i \in \mathcal{D}} -y_i^B \ln \hat{y}_i^B - (1 - y_i^B) \ln(1 - \hat{y}_i^B),$$

BPR Pair-wise Loss (Ranking-Based) [5]:

learning a bipartite ranking relation (positive sample > negative sample)

$$\mathcal{L}^{BPR} = \sum_{x_i \in \mathcal{D}^+} \sum_{x_j \in \mathcal{D}^-} -\log \sigma(r_i - r_j),$$

- Sigmoid(r) is strictly NOT a probability for BPR loss

Method | Task Augmentation

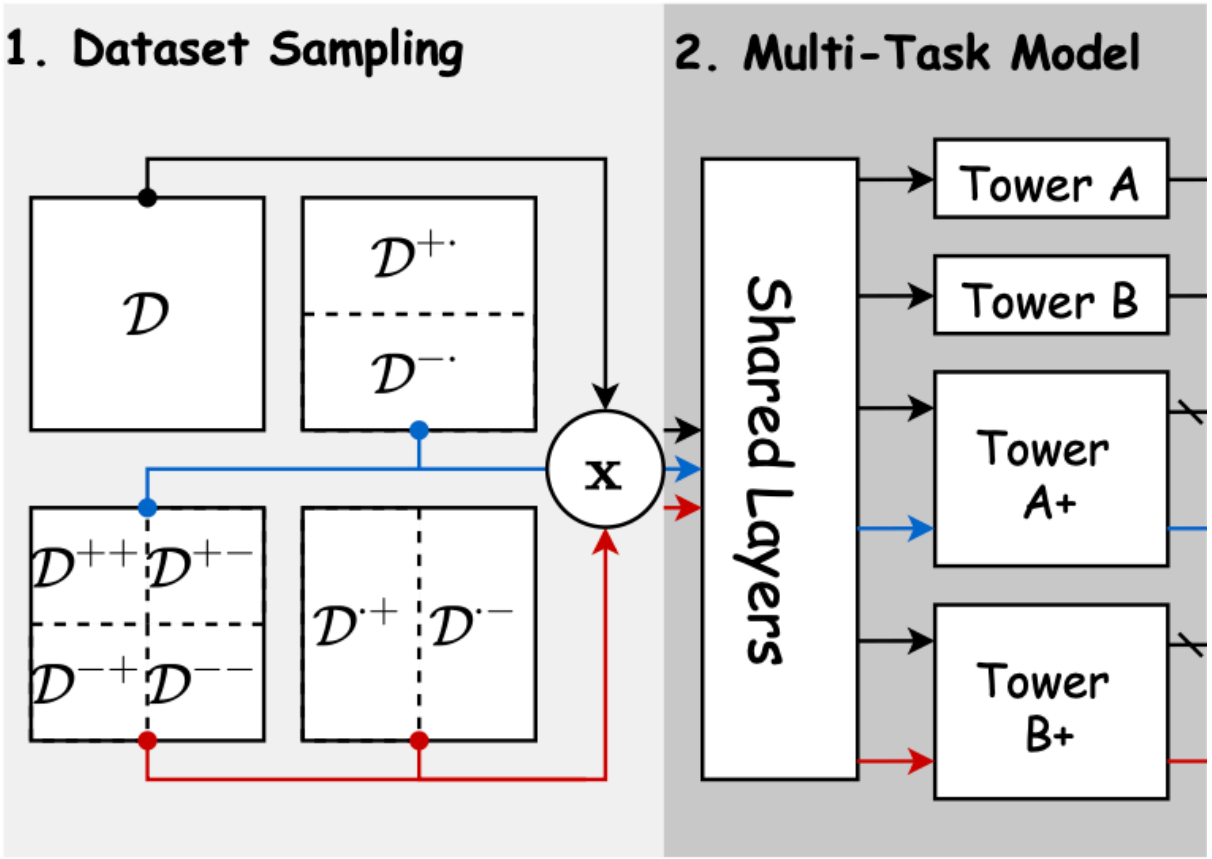


Fig. Training Set Split

Split Training Dataset According to Permutations of Tasks' Labels

$$\mathcal{D}^{+-} = \{\mathbf{x}_i \in \mathcal{D} | y_i^A = 1, y_i^B = 0\},$$
$$\mathcal{D}^{-+} = \{\mathbf{x}_i \in \mathcal{D} | y_i^A = 0, y_i^B = 1\},$$
$$\mathcal{D}^{-\cdot} = \mathcal{D}^{--} \cup \mathcal{D}^{-+},\ \mathcal{D}^{+\cdot} = \mathcal{D}^{+-} \cup \mathcal{D}^{++},$$
$$\mathcal{D}^{\cdot-} = \mathcal{D}^{--} \cup \mathcal{D}^{+-},\ \mathcal{D}^{\cdot+} = \mathcal{D}^{-+} \cup \mathcal{D}^{++}.$$

Method | Task Augmentation

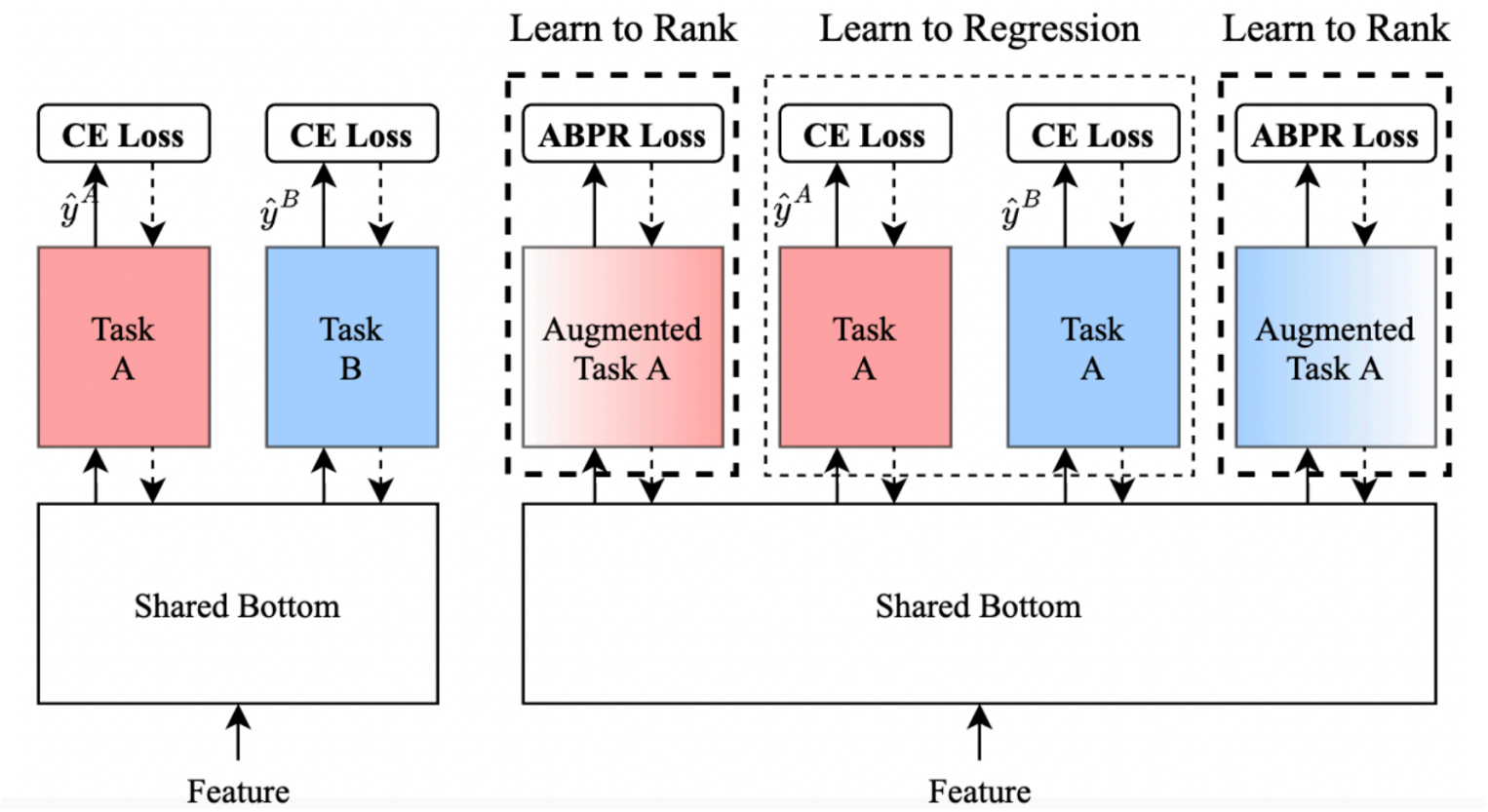


Fig. Task Augmentation

Augmented Ranking-Based Tasks

Aim to learn fine-grained ranking ++ > +- > -+ > - - with the following optimization target :

$$
\begin{aligned}
&\ln p(\Theta|\succ) \\
=&\ln p(\mathbf{x}_{++} \succ \mathbf{x}_{+-}|\Theta) \cdot p(\mathbf{x}_{-+} \succ \mathbf{x}_{--}|\Theta) \cdot p(\Theta) \\
=&\sum_{(\mathbf{x}_{++},\mathbf{x}_{+-},\mathbf{x}_{-+},\mathbf{x}_{--})} \ln \sigma(\hat{r}_{++\succ+-}) + \ln \sigma(\hat{r}_{-+\succ--}) - \\
&- Reg(\Theta),
\end{aligned}
$$

Method | Task Augmentation

Combined with Original BPR Loss:

$$\mathcal{L}^{A+} = \sum_{(\mathbf{x}_{++},\mathbf{x}_{+-},\mathbf{x}_{-+},\mathbf{x}_{--})} -\beta_1^A \ln \sigma(\hat{r}_{++\succ+-}) - \beta_2^A \ln \sigma(\hat{r}_{-+\succ--}) + \sum_{(\mathbf{x}_{+\cdot},\mathbf{x}_{-\cdot})} -\ln \sigma(\hat{r}_{+\cdot\succ-}).$$

Auxiliary Rankings

Coarse-Grained Ranking

$$\mathcal{L}^{B+} = \sum_{(\mathbf{x}_{++},\mathbf{x}_{+-},\mathbf{x}_{-+},\mathbf{x}_{--})} -\beta_1^B \ln \sigma(\hat{r}_{++\succ-+}) - \beta_2^B \ln \sigma(\hat{r}_{+-\succ--}) + \sum_{(\mathbf{x}_{\cdot+},\mathbf{x}_{\cdot-})} -\ln \sigma(\hat{r}_{\cdot+\succ\cdot-}).$$

Method | Calibrated Knowledge Distillation

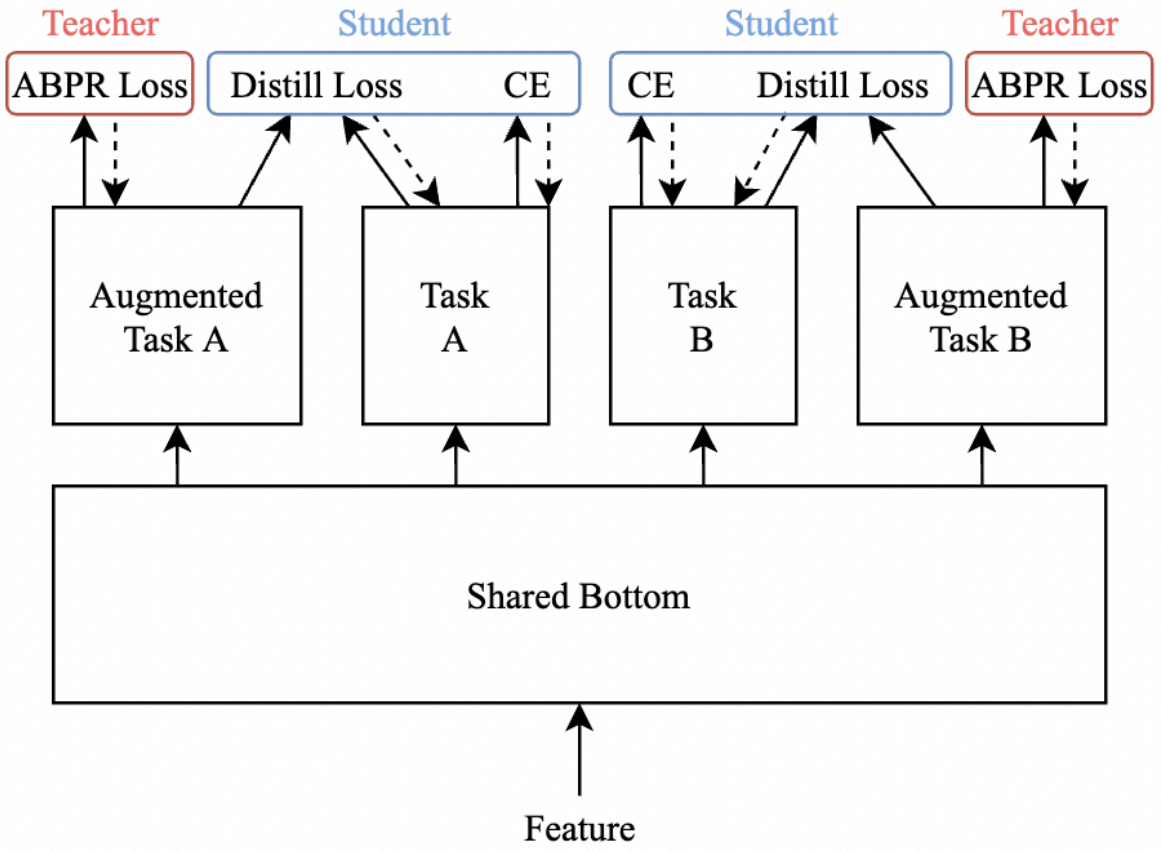


Fig. Directly Using Teacher Outputs

Directly Using Outputs of Teacher for Distillation

Distillation Loss:

$$\mathcal{L}^{A-KD} = CE(\sigma(\hat{r}^{A+}/\tau), \sigma(\hat{r}^{A}/\tau)),$$
$$\mathcal{L}^{B-KD} = CE(\sigma(\hat{r}^{B+}/\tau), \sigma(\hat{r}^{B}/\tau)),$$

Student Loss:

$$\mathcal{L}^{A-Stu} = (1-\alpha^{A})\mathcal{L}^{A} + \alpha^{A}\mathcal{L}^{A-KD},$$
$$\mathcal{L}^{B-Stu} = (1-\alpha^{B})\mathcal{L}^{B} + \alpha^{B}\mathcal{L}^{B-KD},$$

Method | Calibrated Knowledge Distillation

Challenge : Sigmoid(r) is strictly **NOT** a **probability** for BPR loss. **Calibration** is required !

$$\mathcal{L}^{BPR} = \sum_{x_i \in \mathcal{D}^+} \sum_{x_j \in \mathcal{D}^-} -\log \sigma(r_i - r_j),$$

A Simple Solution： Platt Scaling $\tilde{r}^{A+} = P^A \cdot \hat{r}^{A+} + Q^A, \;\; \tilde{y}^{A+} = \dfrac{1}{1+\exp \tilde{r}^{A+}}$

Calibration Loss $\mathcal{L}^{Cal} = \mathcal{L}^{A-Cal} + \mathcal{L}^{B-Cal} = CE(y^A, \tilde{y}^{A+}) + CE(y^B, \tilde{y}^{B+}).$

Method | Training Procedure

Algorithm 1: Training Algorithm for TAKD

Input: Training dataset $\mathcal{D}$, learning rate γ_1 and γ_2, initial parameters Θ and Ω, task weights w_1, w_2, w_3, w_4.

Construct set $\mathcal{D}^{++}, \mathcal{D}^{+-}, \mathcal{D}^{-+}, \mathcal{D}^{--}, \mathcal{D}^{+\cdot}, \mathcal{D}^{-\cdot}, \mathcal{D}^{\cdot+}, \mathcal{D}^{\cdot-}$;

while *Not converged* **do**

  Sample $\mathbf{x}$ uniformly at random from $\mathcal{D}$;

  Sample $\mathbf{x}_{++}, \mathbf{x}_{+-}, \mathbf{x}_{-+}, \mathbf{x}_{--}$ uniformly at random from $\mathcal{D}^{++}, \mathcal{D}^{+-}, \mathcal{D}^{-+}, \mathcal{D}^{--}$ respectively;

  Sample $\mathbf{x}_{+\cdot}, \mathbf{x}_{-\cdot}, \mathbf{x}_{\cdot+}, \mathbf{x}_{\cdot-}$ uniformly at random from $\mathcal{D}^{+\cdot}, \mathcal{D}^{-\cdot}, \mathcal{D}^{\cdot+}, \mathcal{D}^{\cdot-}$ respectively;

  Model parameter Θ optimization:

  Calculate $\mathcal{L}^{A+}(\mathbf{x}_{+\cdot}, \mathbf{x}_{-\cdot}, \mathbf{x}_{++}, \mathbf{x}_{+-}, \mathbf{x}_{-+}, \mathbf{x}_{--}; \Theta)$;

  Calculate $\mathcal{L}^{B+}(\mathbf{x}_{\cdot+}, \mathbf{x}_{\cdot-}, \mathbf{x}_{++}, \mathbf{x}_{+-}, \mathbf{x}_{-+}, \mathbf{x}_{--}; \Theta)$;

  Calculate $\mathcal{L}^{A-Stu}(\mathbf{x}; \Theta), \mathcal{L}^{B-Stu}(\mathbf{x}; \Theta)$;

  $\mathcal{L}^{Model} \leftarrow w_1\mathcal{L}^{A+} + w_2\mathcal{L}^{B+} + w_3\mathcal{L}^{A-Stu} + w_4\mathcal{L}^{B-Stu}$;

  $\Theta \leftarrow \Theta - \gamma_1 \nabla_\Theta \mathcal{L}^{Model}$;

  Calibration parameter Ω optimization:

  Calculate $\mathcal{L}^{Cal}(\mathbf{x}; \Omega)$;

  $\Omega \leftarrow \Omega - \gamma_2 \nabla_\Omega \mathcal{L}^{Cal}$;

end

Two Sets of Parameters:

Theta -> Model (Backbone) Parameters

Omega -> Calibration Parameters

End-to-End Training

Bi-Level Training Procedure:

1. Sampling
2. Updating Model Parameter
3. Updating Calibration Parameter

Method | Correcting Errors

Early Stage of Training — Teacher is NOT well-trained

frequent errors in soft labels may distract the training process of student, causing **slow convergence**

Later Stage of Training — Teacher is well-trained

it is still likely teacher model would occasionally provide mistaken predictions which is called **genetic errors** in another literature

Error Correction Mechanism (Align with the hard label)

$$r^{Teacher}(\mathbf{x}) \leftarrow \mathbb{1}[y] \cdot Max\left\{\mathbb{1}[y] \cdot r^{Teacher}(\mathbf{x}), m\right\}$$

m: Error correction margin $\mathbb{1}[y]$ = 1 if (y == 1) else -1

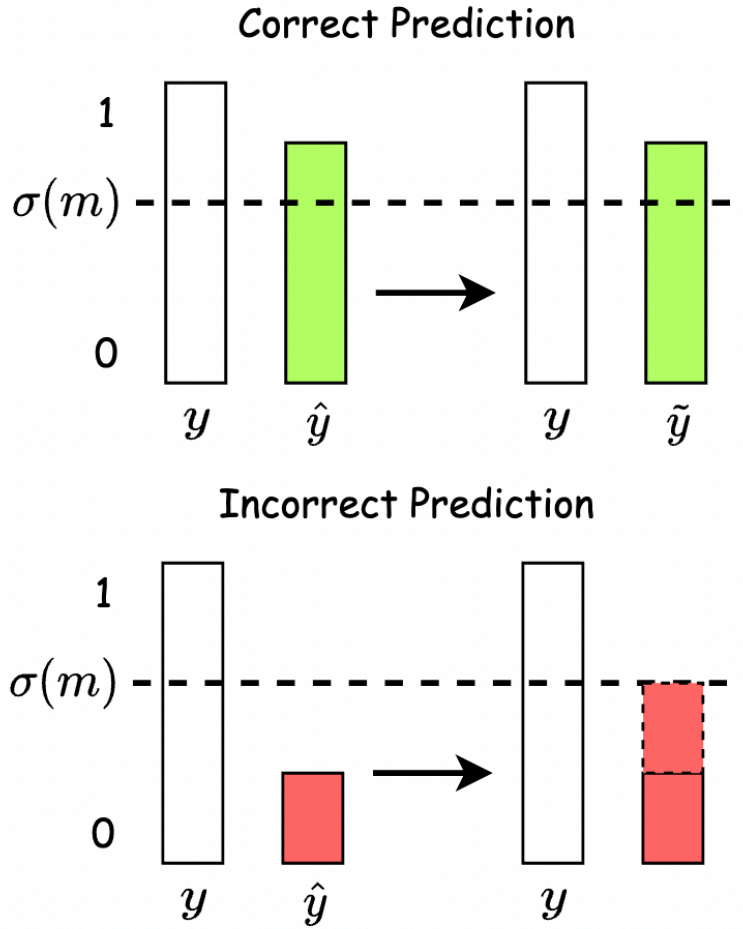


Fig. Error Correction Mechanism

Method | Framework Overview

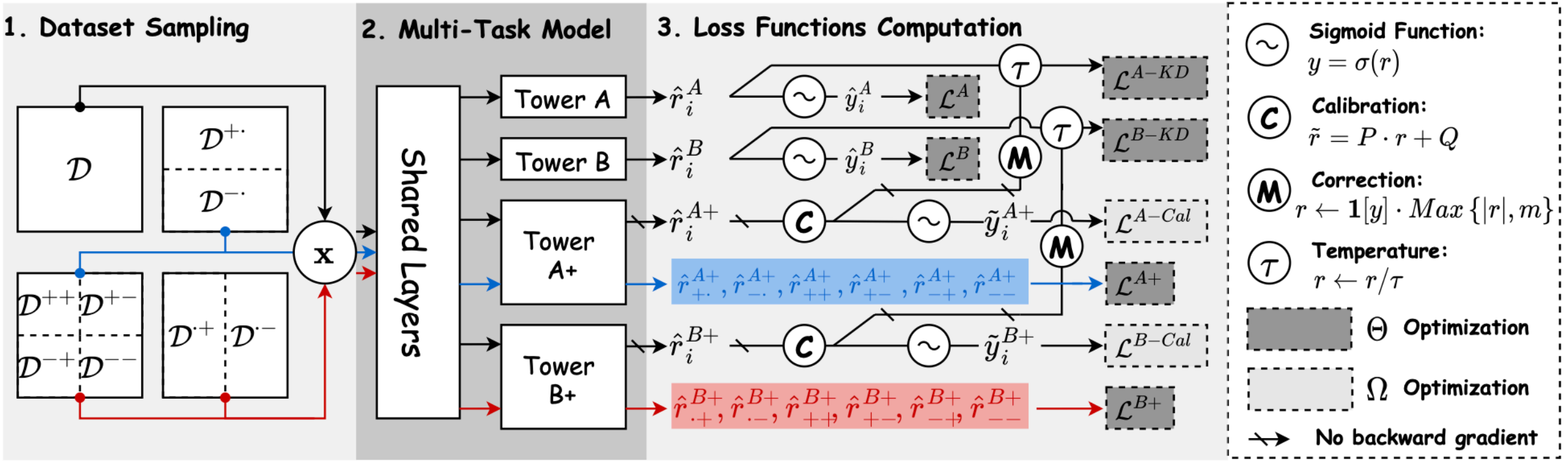


Fig. TAKD framework overview

Method | Adaptions

Example 1: A - Finish Watching, B - Like

Finish & Like > Finish & not Like > not Finish & Like > not Finish & not Like

Example 2: A - CVR, B - CTR

Click & Conversion > Click & not Conversion > not Click

Example 3: A - Click, B - not Interested

Click > not Click > not Interested

Example 4: A, B, C (priority A > B > C)

A & B & C > A & B & notC > A & notB & C > A & notB & notC >
notA & B & C > notA & B & notC > notA & notB & C > notA & notB & notC

Experiments | Settings

Dataset

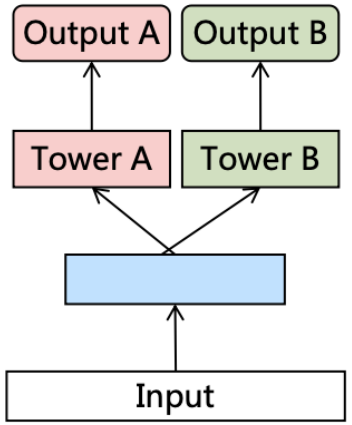
Table 1: Statistics of three datasets.

Datasets	#Samples	#Fields	#Features	Positive Ratio of Task A	Positive Ratio of Task B
TikTok	19622340	9	4691483	37.994%	1.101%
Production	9381820	10	447002	9.975%	1.510%

TikTok: Task A — Finish Watching Task B — Like

Production: Task A — pCTR Task B — Not Interested

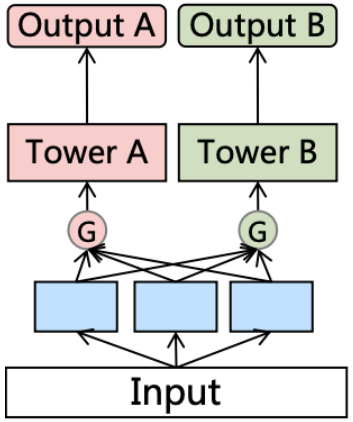
Baselines



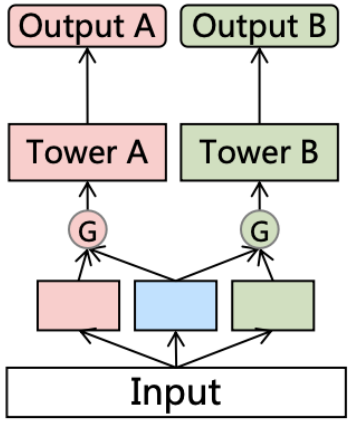
Shared-Bottom

$$\begin{bmatrix} \tilde{x}_1^i \\ \tilde{x}_2^i \end{bmatrix} = \begin{bmatrix} \alpha_{11} & \alpha_{12} \\ \alpha_{21} & \alpha_{22} \end{bmatrix} \begin{bmatrix} x_1^i \\ x_2^i \end{bmatrix}$$

Cross-Stitch



MMOE



PLE

Experiments | Settings

Metrics

AUC : Evaluating Bipartite Ranking Performance

$$\text{AUC} = \frac{1}{N^+N^-} \sum_{\mathbf{x}_i \in D^+} \sum_{\mathbf{x}_j \in D^-} (\mathcal{I}(p(\mathbf{x}_i) > p(\mathbf{x}_j)))$$

Multi-AUC : Evaluating Multipartite Ranking Performance

$$\text{Multi-AUC} = \frac{2}{c(c-1)} \sum_{j=1}^{c} \sum_{k>j}^{c} p(j \cup k) \cdot AUC(k, j),$$

Experiments | Performance Comparison

TABLE II
EXPERIMENT RESULTS OF OUR MODELS AND COMPETITORS ON TIKTOK DATASET.

Methods	TaskA-Student		TaskB-Student		TaskA-Teacher		TaskB-Teacher	
	AUC	Multi-AUC	AUC	Multi-AUC	AUC	Multi-AUC	AUC	Multi-AUC
Single-Model	0.7456	0.6335	0.9491	0.7966	0.7453	0.7140	0.9481	0.8297
Shared-Bottom	$0.7375_{(-0.0081)}$	$0.6344_{(+0.0009)}$	$0.9489_{(-0.0002)}$	$\underline{0.8101}_{(+0.0135)}$	-	-	-	-
Cross-Stitch	$0.7468_{(+0.0012)}$	$0.6445_{(+0.0110)}$	$0.9488_{(-0.0003)}$	$0.7985_{(+0.0019)}$	-	-	-	-
MMOE	$0.7479_{(+0.0023)}$	$\underline{0.6474}_{(+0.0139)}$	$0.9490_{(-0.0001)}$	$0.7980_{(+0.0014)}$	-	-	-	-
PLE	$\underline{0.7485}_{(+0.0029)}$	$0.6464_{(+0.0129)}$	$\underline{0.9495}_{(+0.0004)}$	$0.7983_{(+0.0017)}$	-	-	-	-
TAUG	$0.7491_{(+0.0035)}$	$0.6743_{(+0.0408)}$	$0.9498_{(+0.0007)}$	$0.8081_{(+0.0115)}$	$0.7485_{(+0.0032)}$	$\mathbf{0.7408}_{(+0.0268)}$	$0.9501_{(+0.0020)}$	$\mathbf{0.8335}_{(+0.0038)}$
TAKD	$\mathbf{0.7494}_{(+0.0038)}$	$\mathbf{0.7411}_{(+0.1076)}$	$\mathbf{0.9513}_{(+0.0022)}$	$\mathbf{0.8341}_{(+0.0375)}$	$\mathbf{0.7487}_{(+0.0034)}$	$0.7403_{(+0.0263)}$	$\mathbf{0.9502}_{(+0.0021)}$	$0.8324_{(+0.0027)}$

TABLE III
EXPERIMENT RESULTS OF OUR MODELS AND COMPETITORS ON PRODUCTION DATASET.

Methods	TaskA-Student		TaskB-Student		TaskA-Teacher		TaskB-Teacher	
	AUC	Multi-AUC	AUC	Multi-AUC	AUC	Multi-AUC	AUC	Multi-AUC
Single-Model	0.7597	0.6024	0.7528	0.6270	0.7604	0.6705	0.7535	0.6708
Shared-Bottom	$0.7587_{(-0.0010)}$	$0.6145_{(+0.0121)}$	$0.7540_{(+0.0012)}$	$0.6378_{(+0.0108)}$	-	-	-	-
Cross-Stitch	$0.7600_{(+0.0003)}$	$0.6195_{(+0.0171)}$	$0.7582_{(+0.0054)}$	$0.6360_{(+0.0090)}$	-	-	-	-
MMOE	$0.7605_{(+0.0008)}$	$0.6226_{(+0.0202)}$	$0.7619_{(+0.0091)}$	$\underline{0.6431}_{(+0.0161)}$	-	-	-	-
PLE	$\underline{0.7607}_{(+0.0010)}$	$\underline{0.6240}_{(+0.0216)}$	$\underline{0.7625}_{(+0.0097)}$	$0.6394_{(+0.0124)}$	-	-	-	-
TAUG	$0.7612_{(+0.0015)}$	$0.6394_{(+0.0370)}$	$\mathbf{0.7646}_{(+0.0118)}$	$0.6432_{(+0.0162)}$	$0.7608_{(+0.0004)}$	$0.6768_{(+0.0063)}$	$\mathbf{0.7625}_{(+0.0090)}$	$0.6853_{(+0.0145)}$
TAKD	$\mathbf{0.7618}_{(+0.0021)}$	$\mathbf{0.6861}_{(+0.0837)}$	$0.7644_{(+0.0116)}$	$\mathbf{0.6879}_{(+0.0609)}$	$\mathbf{0.7609}_{(+0.0005)}$	$\mathbf{0.6850}_{(+0.0145)}$	$0.7618_{(+0.0083)}$	$\mathbf{0.6910}_{(+0.0202)}$

Experiments | Ablation Study

TABLE IV
ABLATION ANALYSIS FOR TASK A ON TIKTOK DATASET.

Variants	AUC	Multi-AUC
w/o AuxiliaryRank	$0.7488_{(-0.0006)}$	$0.6510_{(-0.0901)}$
w/o Calibration	$0.7478_{(-0.0016)}$	$0.7396_{(-0.0015)}$
w/o Correction	$0.7486_{(-0.0008)}$	$0.7399_{(-0.0012)}$
KD	$0.7489_{(-0.0005)}$	$0.6901_{(-0.0510)}$
Baseline	**0.7494**	**0.7411**

TABLE V
ABLATION ANALYSIS FOR TASK B ON TIKTOK DATASET.

Variants	AUC	Multi-AUC
w/o AuxiliaryRank	$0.9501_{(-0.0012)}$	$0.8005_{(-0.0336)}$
w/o Calibration	$0.9504_{(-0.0009)}$	$0.8312_{(-0.0029)}$
w/o Correction	$0.9508_{(-0.0005)}$	$0.8310_{(-0.0031)}$
KD	$0.9505_{(-0.0008)}$	$0.8014_{(-0.0327)}$
Baseline	**0.9513**	**0.8341**

w/o AuxiliaryRank : removing BPR losses for leaning auxiliary ranking relations

w/o Calibration : directly employing the teacher model outputs

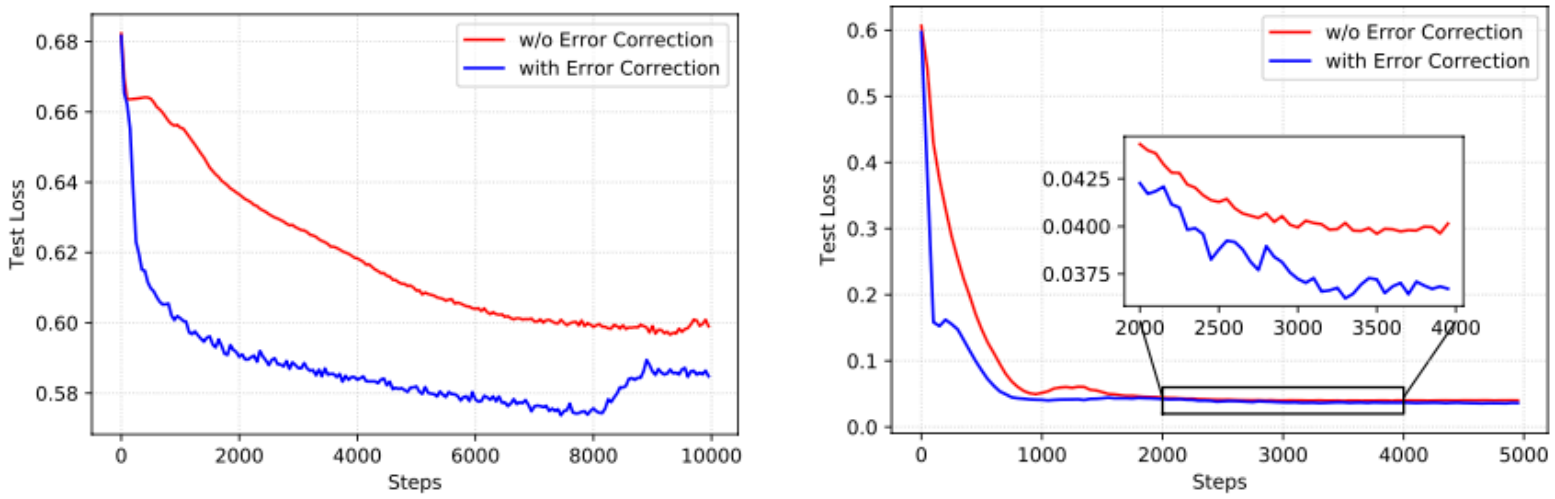
w/o Correction : not applying the error correction mechanism

KD : replacing teachers with regression-based models and using the vanilla knowledge distillation

Baseline : our method

Experiments | Ablation Study

Q: Does **Error Correction Mechanism** Help to Accelerate Convergence and Enhance Knowledge Quality?

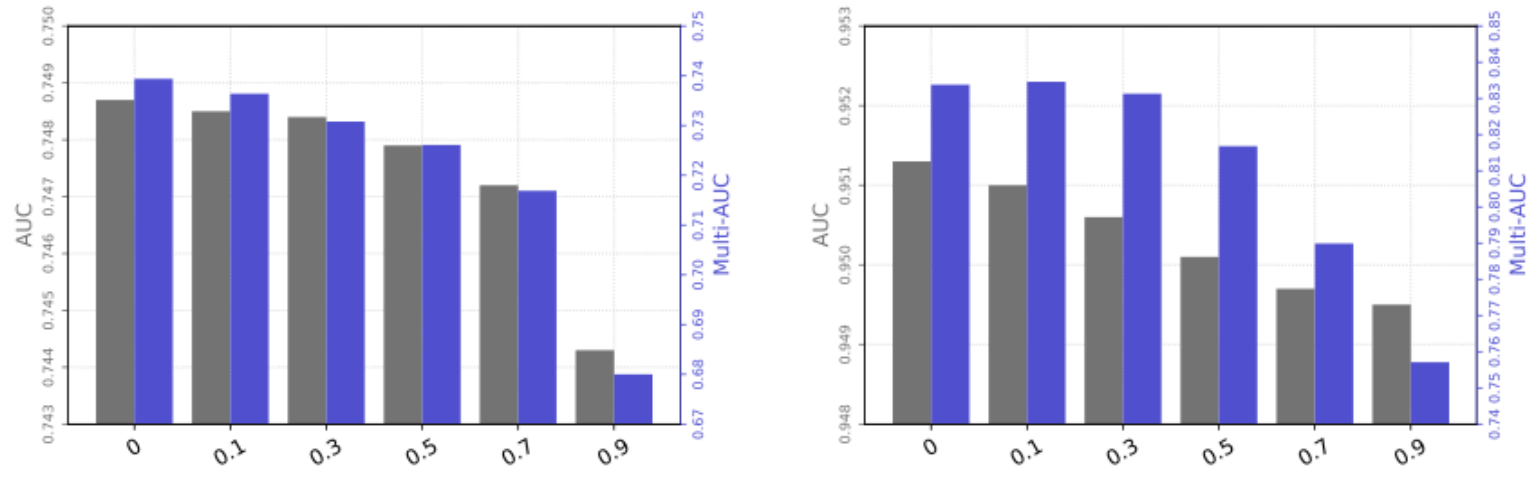


(a) Task A (Finish Watching). (b) Task B (Like).

Fig. 4. Learning curves of TAKD with and without error correction mechanism on TikTok dataset.

Experiments | Ablation Study

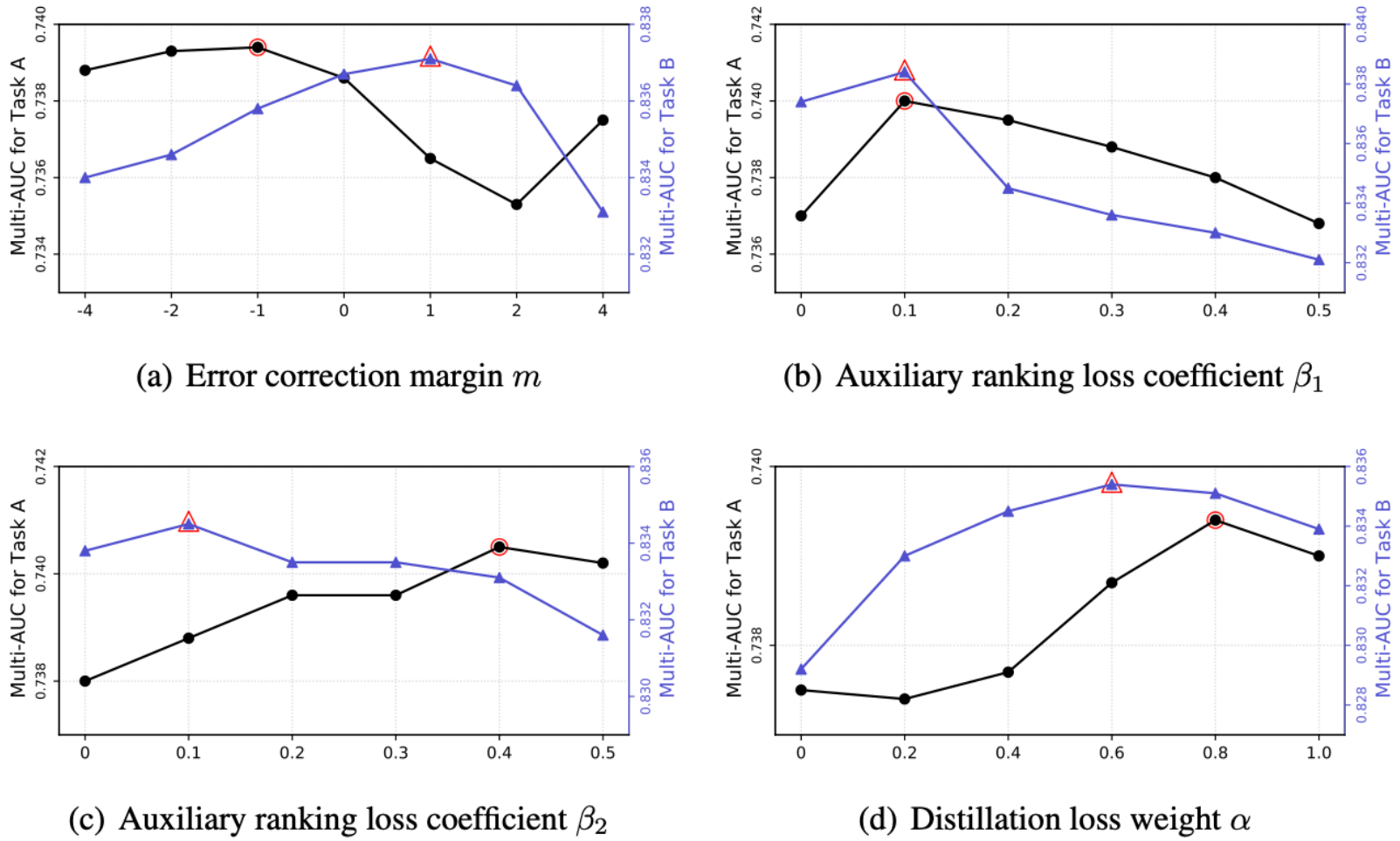
Q: Is Student Really Benefiting from **Auxiliary Ranking Knowledge** from Other Tasks?



(a) Task A (Finish Watching). (b) Task B (Like).

Fig. 5. Impact of corrupted auxiliary ranking information on student model performance for TikTok dataset.

Experiments | Hyper-Parameter Study



(a) Error correction margin m

(b) Auxiliary ranking loss coefficient β_1

(c) Auxiliary ranking loss coefficient β_2

(d) Distillation loss weight α

Figure 5.3 Multi-AUC performance on TikTok dataset for TaskA and Task B w.r.t. different hyper-parameters.

References

[1] H. Tang, J. Liu, M. Zhao, and X. Gong, “Progressive layered extraction (ple): A novel multi-task learning (mtl) model for personalized recom- mendations,” in Fourteenth ACM Conference on Recommender Systems, 2020, pp. 269–278.
[2] J. Ma, Z. Zhao, J. Chen, A. Li, L. Hong, and E. H. Chi, “Snr: Sub- network routing for flexible parameter sharing in multi-task learning,” in Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, no. 01, 2019, pp. 216–223.
[3] C. Xu, Q. Li, J. Ge, J. Gao, X. Yang, C. Pei, F. Sun, J. Wu, H. Sun, and W. Ou, “Privileged features distillation at taobao recommendations,” in Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2020, pp. 2590–2598.
[4] L. Yuan, F. E. Tay, G. Li, T. Wang, and J. Feng, “Revisit knowledge distillation: a teacher-free framework,” arXiv preprint arXiv:1909.11723, 2019.
[5] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme, “Bpr: Bayesian personalized ranking from implicit feedback,” arXiv preprint arXiv:1205.2618, 2012.