Adaptive Data and Task Joint Scheduling for Multi-Task Learning

Zeyu Liu[1], Heyan Chai[2], Chaoyang Liu[1,3], Lingzhi Wang[1] and Qing Liao[1,3]✉

[1] Harbin Institute of Technology (Shenzhen), Shenzhen, China
[2]The Chinese University of Hong Kong, HongKong, China
[3] Peng Cheng Laboratory, Shenzhen, China
liuzeyu@stu.hit.edu.cn, chaiheyan@gmail.com, lichy@pcl.ac.cn, {wanglingzhi liaoqing}@hit.edu.cn

Quick Review:

Multi-Task Learning

💡 **Aims to use a deep learning model to learn multiple tasks simultaneously, improving performance across all tasks by sharing knowledge between them.**

⚙️ **Compared to single-task learning, multi-task learning enhances data efficiency and saves researchers time and effort in building multiple independent models.**

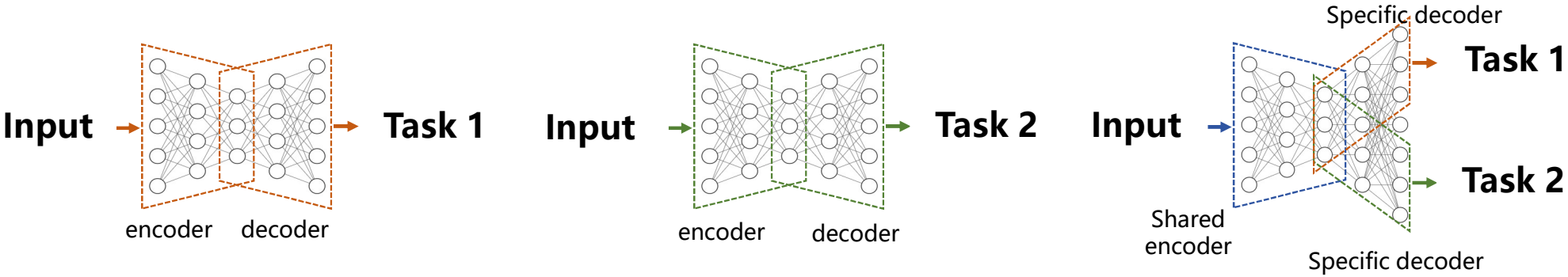


Figure 1-1 Comparison between Single Task Learning and Multi Task Learning

Quick Review:

There exists conflicting in MTL:

- Different optimization objectives
- Inconsistent training progress.

Performance Degradation

Some or even all tasks perform worse than when trained individually.

Task conflict specifically manifests as **Gradient Conflict** between tasks:

a. Task Gradient Direction Conflict

b. Task Gradient Magnitude Imbalance

Figure 1-2 Examples of task gradient conflicts

- Task Gradient **Direction Conflict**: Model may struggle to optimize all tasks simultaneously and converge to a stable optimal solution.
- Task Gradient **Magnitude Imbalance**: Tasks with larger gradients may dominate the training process.

How to Mitigate Task Conflicts

Related Solution:

Design Specially MTL Architecture

Replace the shared bottom network with an expert network, and each task has its own gating network to combine the outputs of multiple expert networks.

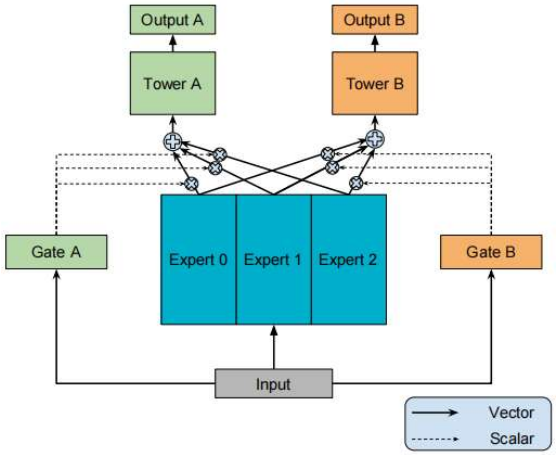


Figure 1-3 MMoE

Design MTL strategy

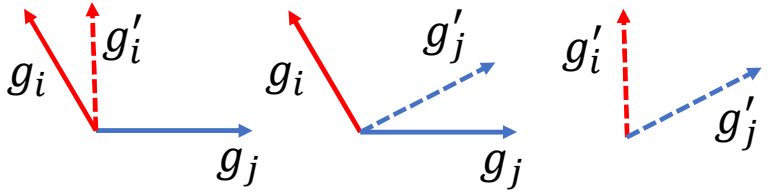


Figure 1-4 PCGrad

Eliminating the conflicting components of task gradients.

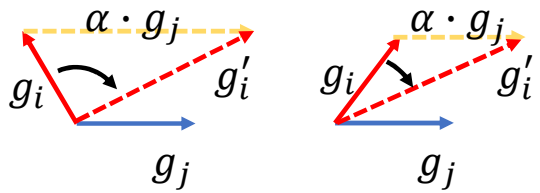


Figure 1-5 GradVac

Pre-adjust task gradients, adaptively tuning and aligning them.

Motivation-Data Level:

The difficulty level of the same data sample varies across different tasks.

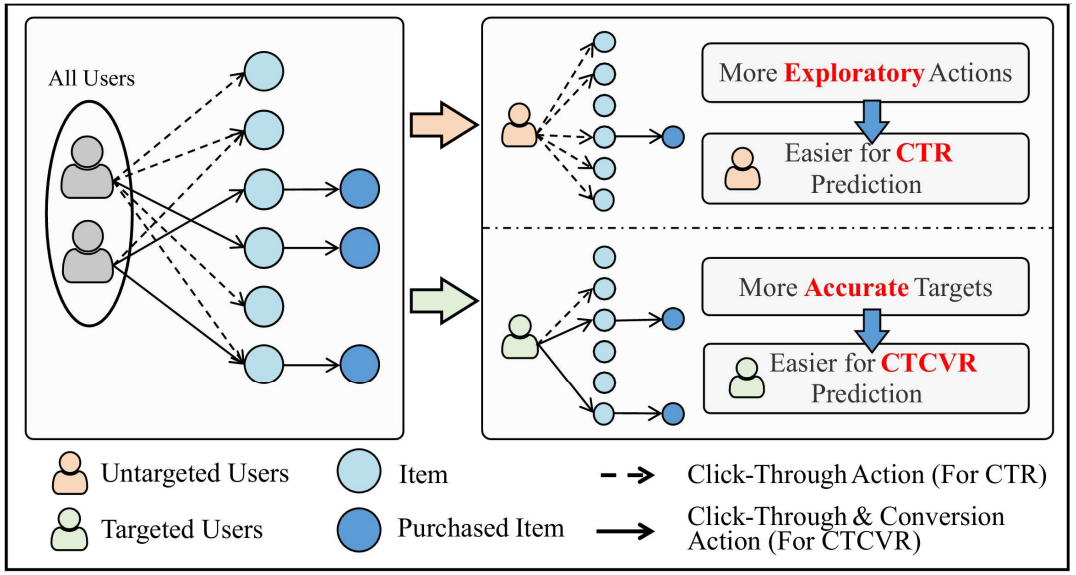
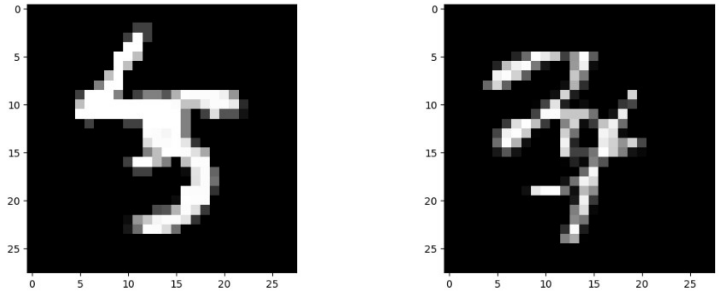


Figure 2-1 Impact of data sample on click-through rate (CTR) and click-through conversion rate (CTCVR) prediction tasks

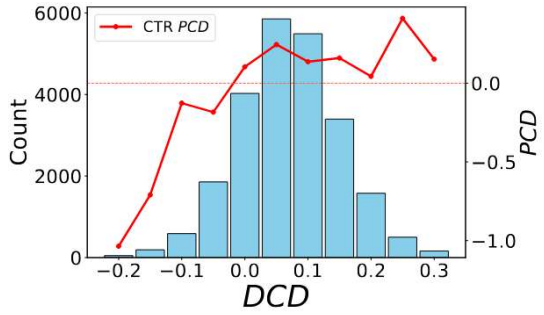


(a) Labeled with [5,5] (b) Labeled with [3,4]

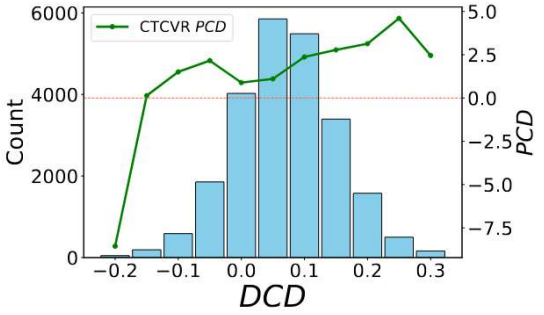
Figure 2-2 Examples in MultiMNIST Dataset

Data affect the updating of model parameters, so the same data have different effects on different tasks.

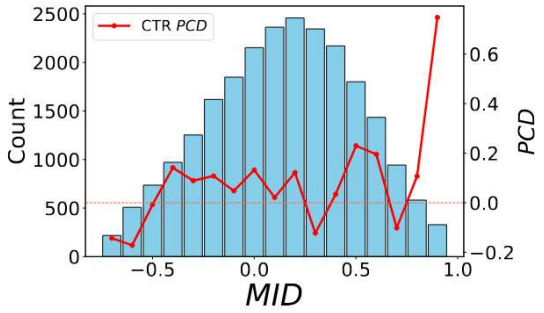
Motivation-Task Level:



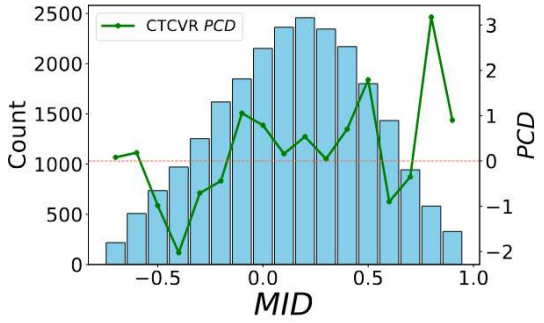
(a) PCD and DCD relationship on CTR task



(b) PCD and DCD relationship on CTCVR task



(c) PCD and MID relationship on CTR task



(d) PCD and MID relationship on CTCVR task

Figure 2-3 The influence of task gradient conflict on different tasks.

Compared with (a) and (b)

Even the same degree of task gradient direction conflict (magnitude imbalance) impact tasks differently.

Compared with (a) and (c)

Gradient direction conflicts and magnitude imbalances inconsistently affect task performance.

Should account for differential impacts of gradient conflicts on tasks to achieve more effective joint learning.

Methodology:

Data and Task Joint Scheduling (DTJS)

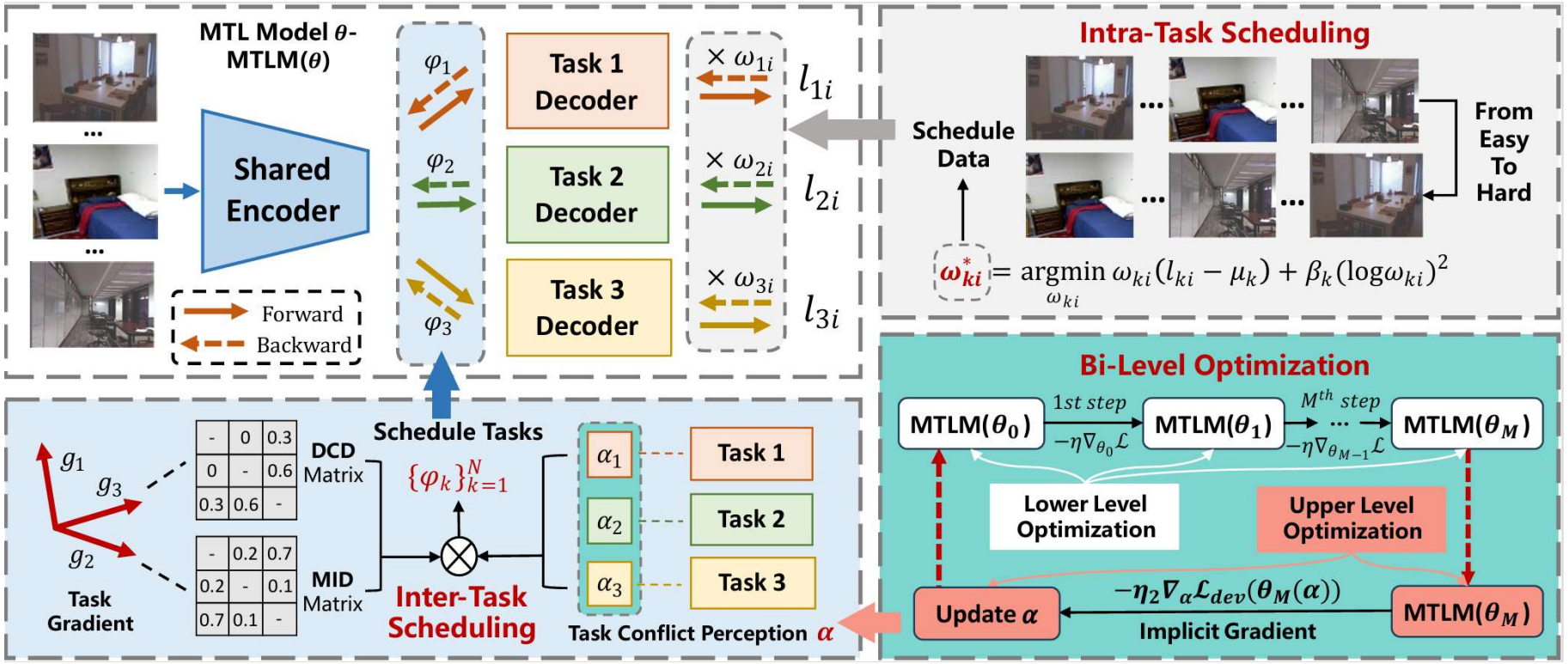


Figure 3.1 **The overall framework of DTJS.** DTJS consists of intra-task scheduling, inter-task scheduling, and bi-level optimization. The MTL model parameter is updated during the lower-level optimization, and the learnable task conflict perception is optimized via upper-level optimization.

Methodology-Intra-Task Scheduling

- **Goals**: **Measure difficulty of same data across tasks and mitigate negative impact of difficult data.**

- **Data Difficulty**: $l_{ki} = l_k(f_k(x_i^k), y_i^k; \theta^k)$ （3-1）
- **Standard of Difficulty**: $\mu_k^{(t)} = \gamma_k \mu_k^{(t-1)} + (1-\gamma_k)\mathcal{L}_k^{(t)}$ （3-2）
- **Data Weight**: $\omega_{ki}^* = \underset{\omega_{ki}}{\operatorname{argmin}}\, \omega_{ki}(l_{ki} - \mu_k) + \beta_k(\log\omega_{ki})^2$ （3-3）

By using Alternative Optimization Strategy, **Data Weight is**:

$$w_{ki} = \begin{cases} e, & \frac{l_{ki}-\mu_k}{2\beta_k} \le -\frac{1}{e}, \\ e^{-\mathcal{W}\left(\frac{l_{ki}-\mu_k}{2\beta k}\right)}, & \frac{l_{ki}-\mu_k}{2\beta_k} > -\frac{1}{e}, \end{cases} \quad （3\text{-}4）$$

$\mathcal{W}$ is Lambert W function

- **Loss function after intra-task scheduling**:

$$l_{k\_intra}(f_k(x_i^k), y_i^k; \theta^k) = \omega_{ki}(l_{ki} - \mu_k) + \beta_k(\log\omega_{ki})^2 \quad （3\text{-}5）$$

Assign lower weights to difficult data samples;
Weights become uniform in later training stages.

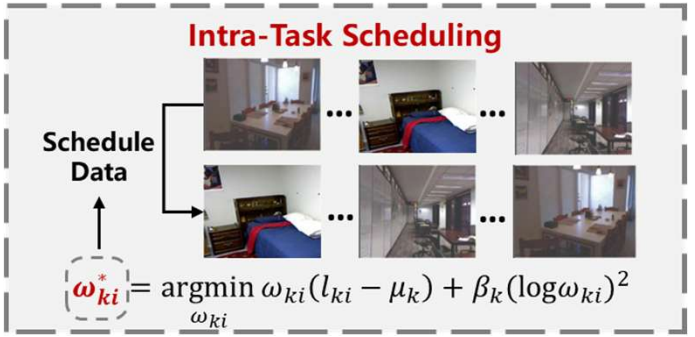


Figure 3-2: Intra-Task Scheudling

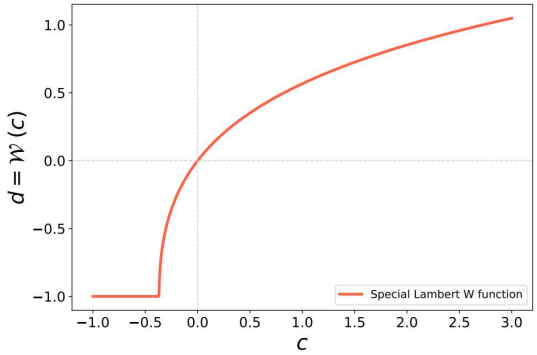


Figure 3-3: Lambert W function

Methodology-Inter-Task Scheduling

- **Goals**: Adaptively schedule tasks based on gradient conflict impact to mitigate negative effects of task conflicts.

Measure the task conflicts degree on Task k

$$d_k = \sum_{j \neq k}^{N} DCD(g_k, g_j) \quad (3\text{-}6)$$

$$m_k = \sum_{j \neq k}^{N} MID(g_k, g_j) \quad (3\text{-}7)$$

$$DCD(g_i, g_j) = \frac{g_i \cdot g_j}{||g_i||\,||g_j||},$$

$$MID(g_i, g_j) = \frac{4||g_i||\,||g_j||}{(||g_i||^2 + ||g_j||^2)} - 1,$$

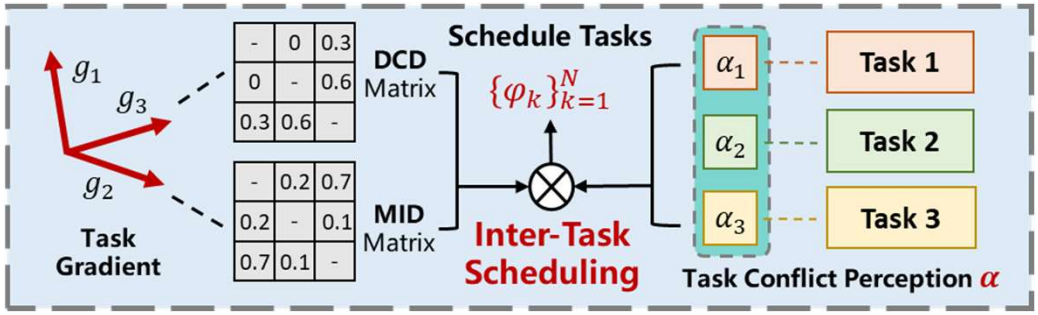


Figure 3-3: Inter-Task Scheudling

Adaptive Task Conflict Perception α

- Enable tasks to adaptively sense conflict impact on d_k, m_k : $\alpha = \{\{a_k, b_k\}_{k \in N}\}$

The results after inter-task scheduling

$$\varphi_k = \sigma(a_k \cdot d_k) + \sigma(b_k \cdot m_k) \quad (3\text{-}8)$$

Higher φ_k indicates task conflicts influence task k less, model will prioritize learning this task.

Loss after inter-task scheduling: $\mathcal{L}(\theta, \alpha; \mathcal{D}_{train}) = \sum_{k=1}^{N} \varphi_k \sum_{i=1}^{n_k} l_{k\_intra}(f_k(x_i^k), y_i^k; \theta^k)$ (3-9)

Methodology-Bi-level Optimization

Why Need Bi-level ? Model parameters θ are optimized through joint scheduling, influenced by learnable task conflict perception α, which is in turn affected by θ.

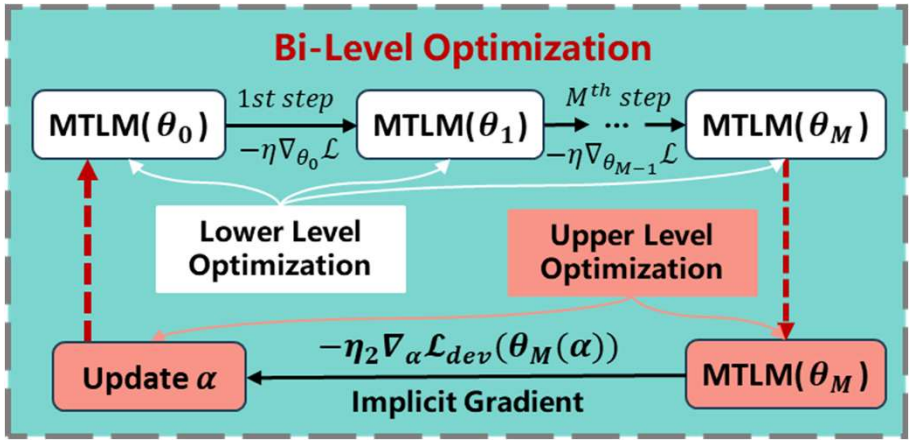


Figure 3-4：Bi-level Optimization

Lower-Level Optimization:

$$\theta_M = \theta_{M-1} - \eta\nabla_{\theta_{M-1}}\mathcal{L}(\theta_{M-1}, \alpha; \mathcal{D}_{train}) \quad (3\text{-}10)$$

Upper-Level Optimization:

To obtain the implicit gradient of the perception parameter α, we separates a small dataset $\mathcal{D}_{dev}$ from the validation set.

The objective of the upper-level optimization is:

$$\mathcal{L}_{dev}(\theta_M(\alpha); \mathcal{D}_{dev}) = \frac{1}{N}\sum_{k=1}^{N}\sum_{i=1}^{m_k} l_{k\_intra}(f_k(x_i^k), y_i^k; \theta^k) \quad (3\text{-}11)$$

Final Results: $\alpha = \alpha - \eta_2\nabla_\alpha\mathcal{L}_{dev}(\theta_M(\alpha); \mathcal{D}_{dev}),$

where:

$$\nabla_\alpha\mathcal{L}_{dev}(\theta_M(\alpha); \mathcal{D}_{dev}) = \nabla_{\theta_M}\mathcal{L}_{dev}(\theta_M(\alpha); \mathcal{D}_{dev})\nabla_\alpha\theta_M$$

$$\nabla_\alpha\theta_M \approx -\sum_{\tau=0}^{M}(I - \eta\nabla^2_{\theta_M}\mathcal{L}(\theta_M, \alpha; \mathcal{D}_{train}))^\tau\eta\nabla_\alpha\nabla_{\theta_M}\mathcal{L}(\theta_M, \alpha; \mathcal{D}_{train})$$

Experiment:

Datasets and Tasks:

- **NYUv2 Dataset:** An indoor scene understanding dataset with three tasks: 13-class semantic segmentation, depth estimation, and surface normal prediction.
- **MultiMNIST Dataset:** A multi-task version of the handwritten digit dataset, with two tasks: classification of the digit on the left and classification of the digit on the right.
- **AliExpress Dataset:** A recommendation system dataset covering four countries (Netherlands, Spain, France, and the United States), with two tasks: click-through rate prediction and click conversion rate prediction.。
- **QM9 Dataset:** A molecular dataset with 11 regression tasks for predicting molecular properties.

Evaluation: We also use the average per-task performance improvement to evaluate the performance of multi-task learning。

$$\triangle m\% = \frac{1}{K}\sum_{k=1}^{K}(-1)^{\delta_k}(M_{m,k} - M_{b,k})/M_{b,k} \quad (3\text{-}12)$$

Experiment:

RQ1: How does DTJS perform compared to SOTA?

TABLE II: PERFORMANCE ON ALIEXPRESS DATASET WITH 2 RECOMMENDATION SYSTEM TASKS IN 4 COUNTRIES

Method	Netherlands, NL		Spain, ES		France, FR		America, US		△m%↑
	CTR↑	CTCVR↑	CTR↑	CTCVR↑	CTR↑	CTCVR↑	CTR↑	CTCVR↑	
STL	0.7222	0.8590	0.7266	0.8855	0.7259	0.8737	0.7061	0.8637	-
MGDA	0.7205	0.8598	0.7223	0.8867	0.7194	0.8761	0.7051	0.8663	-0.13%
DWA	0.7247	0.8595	0.7290	0.8894	0.7268	0.8775	0.7065	0.8706	+0.32%
PCGrad	0.7227	0.8593	0.7286	0.8872	0.7262	0.8745	0.7026	0.8697	+0.11%
GradDrop	0.7258	0.8622	0.7297	0.8908	0.7270	0.8724	0.7061	0.8728	+0.37%
GradVac	0.7249	0.8614	0.7289	0.8889	0.7265	0.8787	0.7062	0.8718	+0.37%
IMTL	0.7238	0.8593	0.7261	0.8821	0.7185	0.8777	0.7047	0.8688	-0.04%
RLW	0.7251	0.8559	0.7285	0.8877	0.7236	0.8771	0.7066	0.8692	+0.16%
RGW	0.7231	0.8582	0.7279	0.8892	0.7269	0.8754	0.7035	0.8707	+0.18%
Nash_MTL	0.7232	0.8628	0.7277	0.8782	0.7257	0.8605	**0.7081**	0.8611	-0.21%
MCLGS	0.7225	0.8543	0.7286	0.8908	0.7271	0.8749	0.7072	0.8684	+0.14%
Aligned_MTL	0.7246	0.8606	0.7277	0.8887	0.7236	**0.8799**	0.7058	0.8726	+0.30%
MoCo	0.7264	0.8571	0.7266	0.8565	0.7268	0.8741	0.7052	0.8675	-0.30%
MoCoGrad	0.7258	0.8583	0.7316	0.8870	0.7244	0.8783	0.7072	0.8691	+0.29%
DTJS	**0.7276**	**0.8632**	**0.7323**	**0.8934**	**0.7271**	0.8788	0.7079	**0.8741**	**+0.64%**

Experiment:

RQ1: How does DTJS perform compared to SOTA?

TABLE III: PERFORMANCE ON NYUV2 DATASET WITH 3 SCENE UNDERSTANDING TASKS

Method	Segmentation		Depth		Surface Normal					△m%↑
	Accuracy↑		Error↓		Angle Distance↓		Within $t°$↑			
	mIoU	Pix Acc	Abs Err	Rel Err	Mean	Median	11.25	22.5	30	
STL	48.04	71.33	0.4188	0.1746	26.46	19.53	30.95	55.47	66.73	–
MGDA	40.49	65.39	0.4625	0.1883	25.68	18.93	31.66	56.79	68.15	-6.04%
DWA	48.47	**71.82**	0.4135	0.1727	26.29	19.46	31.25	55.61	66.87	+0.71%
PCGrad	47.28	70.60	0.4187	0.1787	26.17	19.37	31.47	55.70	66.92	-0.07%
GradDrop	47.87	70.86	0.4127	0.1723	26.37	19.66	31.41	55.96	67.25	+0.51%
GradVac	48.29	71.28	0.4141	0.1731	26.05	19.34	31.05	55.92	67.31	+0.77%
IMTL	46.25	69.84	0.4238	0.1777	26.22	19.31	31.59	55.87	67.07	-0.38%
RLW	48.15	71.49	0.4146	**0.1718**	25.79	18.93	32.08	56.60	67.76	+1.76%
RGW	48.33	71.49	0.4144	0.1741	25.95	19.23	31.47	56.07	67.38	+1.03%
Nash_MTL	47.70	70.81	0.4202	0.1761	25.66	**18.34**	**32.79**	**57.75**	**68.62**	+2.16%
MCLGS	48.48	71.51	0.4159	0.1757	25.87	18.99	32.08	56.47	67.63	+1.44%
Aligned_MTL	47.34	70.31	0.4266	0.1815	25.45	18.61	32.18	57.27	68.40	+1.05%
MoCo	47.48	70.35	0.4349	0.1723	**25.38**	18.92	31.22	56.89	68.38	+0.89%
MoCoGrad	48.15	71.61	0.4146	0.1758	25.71	18.81	32.04	56.91	68.03	+1.73%
DTJS	**49.01**	71.78	**0.4116**	0.1720	25.49	18.54	32.74	57.38	68.42	**+2.93%**

Experiment:

RQ1: How does DTJS perform compared to SOTA?

TABLE IV: EXPERIMENTAL RESULTS ON MULTIMNIST DATASET (2 IMAGE CLASSIFICATION TASKS) AND QM9 DATASET (11 MOLECULAR PROPERTY PREDICTION TASKS)

Method	MultiMNIST		△m%↑	QM9	△m%↑
	Left Acc↑	Right Acc↑		Avg MAE↓	
STL	96.46	95.86	-	0.7474	-
MGDA	96.75	95.68	+0.02%	0.5961	+20.24%
DWA	96.70	95.60	-0.01%	0.5816	+22.17%
PCGrad	96.82	95.81	+0.16%	0.5834	+21.93%
GradDrop	96.70	95.89	+0.14%	0.7168	+4.08%
GradVac	96.74	95.79	+0.11%	0.5673	+24.09%
IMTL	96.62	95.47	-0.11%	0.6372	+14.74%
RLW	96.52	95.54	-0.13%	0.7322	+2.03%
RGW	96.66	95.66	-0.01%	0.7071	+5.40%
Nash_MTL	96.73	95.61	+0.01%	0.6744	+9.77%
MCLGS	96.74	95.68	+0.06%	0.7386	+1.18%
Aligned_MTL	96.71	95.66	+0.03%	0.5999	+19.73%
MoCo	95.94	94.94	-0.74%	1.0602	-41.85%
MoCoGrad	96.04	94.61	-0.86%	0.5864	+21.54%
DTJS	**96.96**	**96.02**	**+0.34%**	**0.5032**	**+32.66%**

Results on Recommendation System:

DTJS achieves best performance in 7 out of 9 metrics, with the remaining two being sub-optimal.

Results on Computer Vision:

On **NYUv2**, DTJS demonstrates superior task-balanced optimization, which achieves the best average performance improvement, despite not attaining absolute dominance in all metrics. On **MultiMNIST**, DTJS achieves the best performance across all metrics.

Results on Recommendation System:

DTJS approach achieved the lowest average MAE scores with 0.5032 and the most significant performance improvement.

Experiment:

RQ2: How do the various components in DTJS affect performance?

TABLE V: ABLATION STUDY RESULTS ON NYUV2 DATASET

Method	Segmentation		Depth		Surface Normal					△m%↑
	Accuracy↑		Error↓		Angle Distance↓		Within $t°$↑			
	mIoU	Pix Acc	Abs Err	Rel Err	Mean	Median	11.25	22.5	30	
STL	48.04	71.33	0.4188	0.1746	26.46	19.53	30.95	55.47	66.73	–
intra-only	47.50	70.59	0.4168	0.1767	26.25	19.60	31.13	55.34	66.73	-0.23%
inter-only	48.13	71.06	0.4138	0.1734	25.79	19.00	31.61	56.54	67.73	+1.38%
fixed-α	48.21	71.22	0.4162	**0.1711**	25.82	18.81	32.18	56.81	67.81	+1.89%
α-only	48.07	71.35	0.4182	0.1712	25.82	19.18	31.52	56.22	67.54	+1.20%
DTJS	**49.01**	**71.78**	**0.4116**	0.1720	**25.49**	**18.54**	**32.74**	**57.38**	**68.42**	**+2.93%**

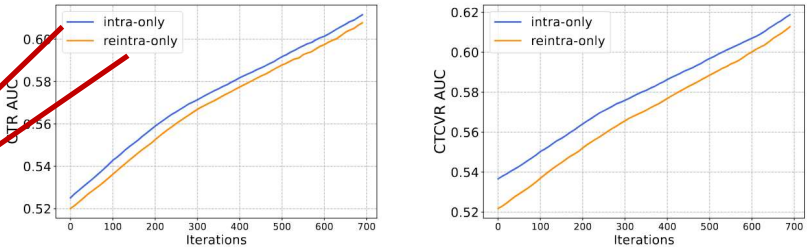
1. intra-only: We focus exclusively on intra-task scheduling;
2. inter-only: We concentrate solely on inter-task scheduling;
3. fixed-α: Replace the learnable task perception parameters with fixed constants.;
4. α-only: Rely solely on learnable parameters without taking task relationships into account.

Experiment:

RQ3: How do DTJS's intra-task scheduling and inter-task scheduling work, and why are they reasonable?

Easy to Hard

Hard to Easy



(a) CTR AUC Changes (b) CTCVR AUC Changes

Fig. 5: Performance changes of two variants of DTJS.

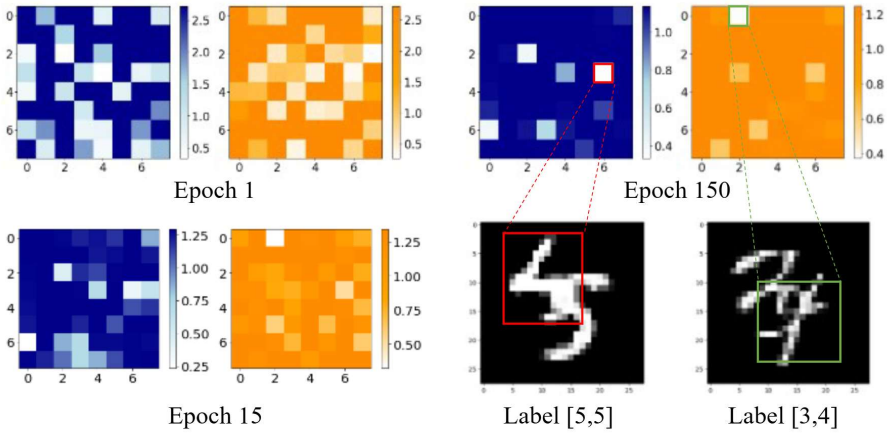
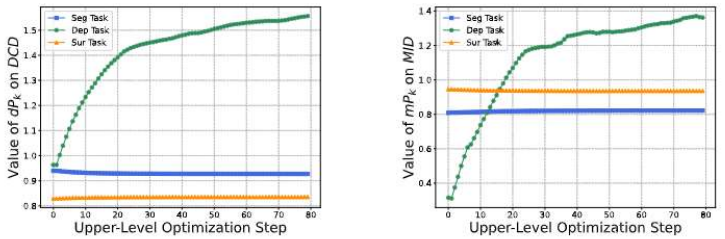
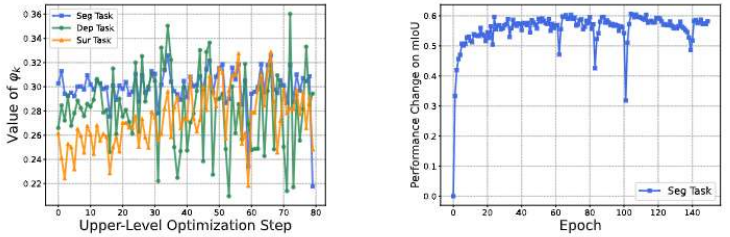


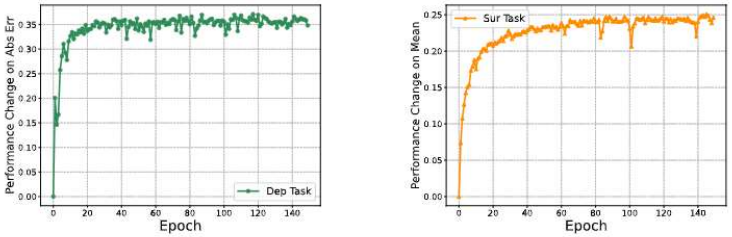
Fig. 6: Visualization of intra-task scheduling.



(a) Direction Perception Parameters dP_k Changes (b) Magnitude Perception Parameters mP_k Changes



(c) Scheduled Tasks' Weight (d) Performance Change on Seg



(e) Performance Change on Dep (f) Performance Change on Sur

Fig. 7: Visualization of inter-task scheduling and performance changes on NYUv2 dataset. (d)-(f) show the performance change when tasks are viewed as equals.

Experiment:

RQ4: Can DTJS reduce the task gradient conflicts?

RQ5: How does DTJS perform together with other MTL architectures?

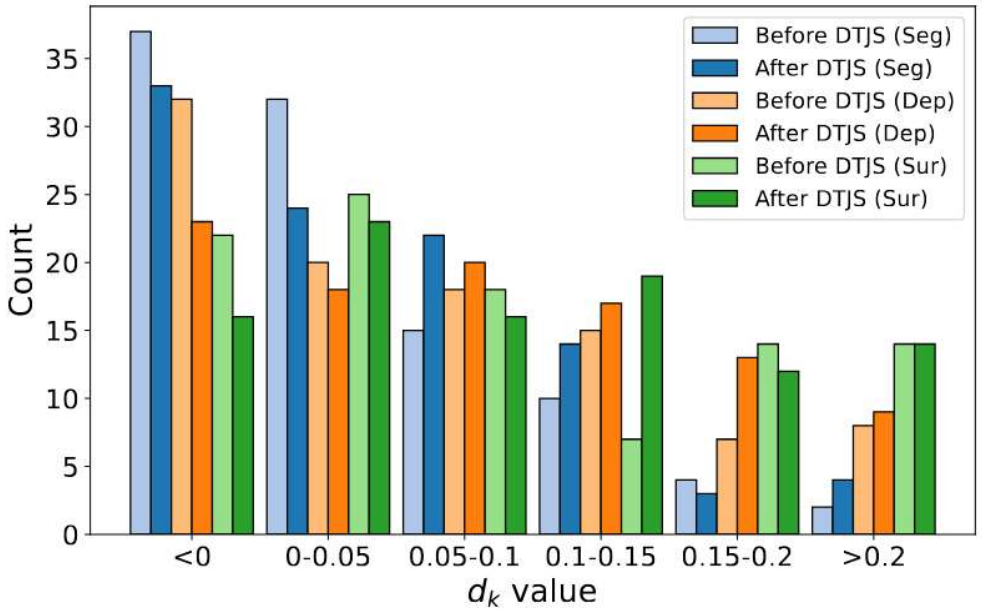


Fig. 8: The frequency of the value of d_k in various ranges.

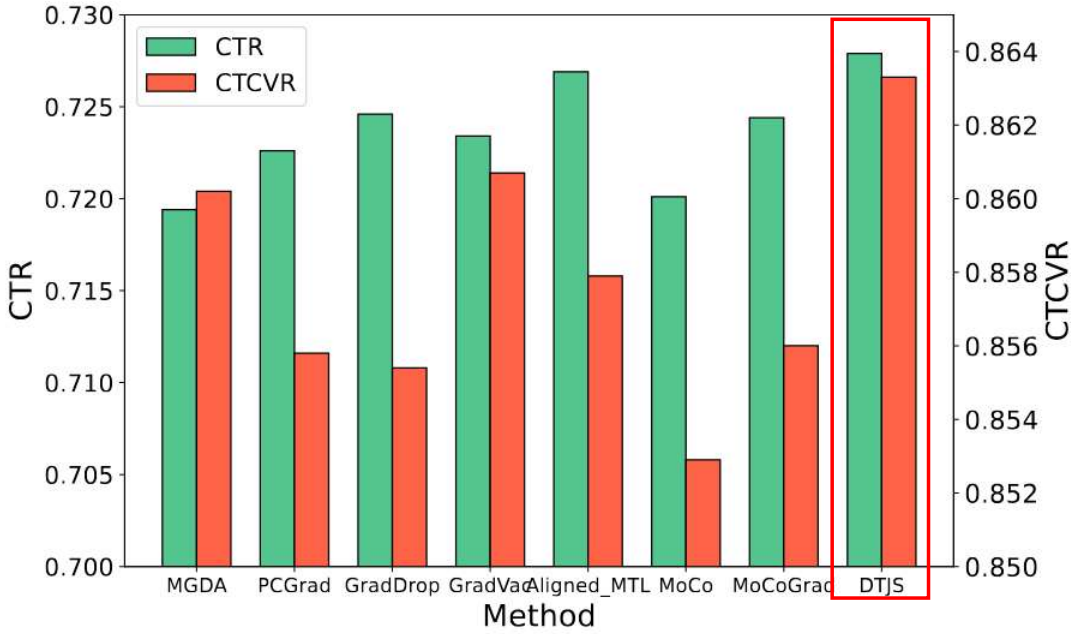


Fig. 9: Experimental results under MMoE architecture.

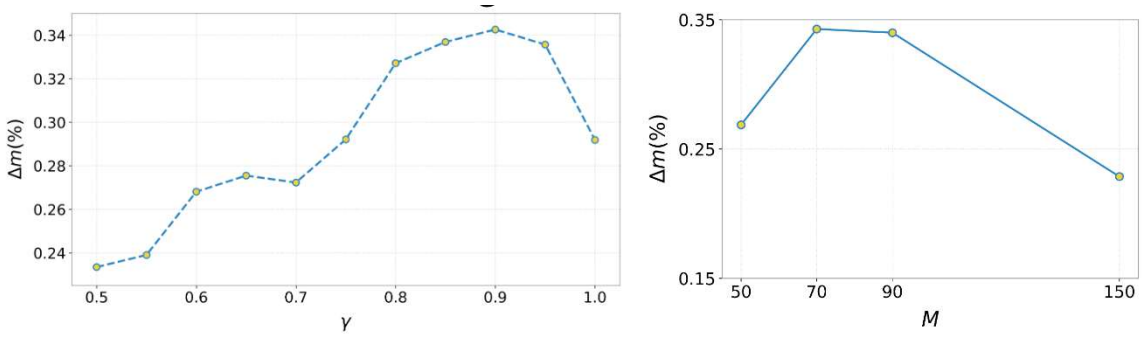
DTJS can reduce the frequency of task gradient conflicts and alleviate the degree of conflict.

It demonstrates the versatility of DTJS across different MTL architectures and its potential for effective integration with other MTL architectures.

Experiment:

Other Experiments

Hyperparameter Experiment



(a) The impact of γ. (b) The impact of M.

Fig. 10: The impact of different values of different γ and M on MultiMNIST dataset.

Empirical Computational Complexity

TABLE VI: COMPUTATIONAL COMPLEXITY

Method	MultiMNIST	AliExpress US
	RT(s/epoch)	RT(s/epoch)
DWA	3.31	116.32
PCGrad	4.98	171.17
MoCo	4.41	187.35
MoCoGrad	5.65	224.23
DTJS	7.21	235.25

Thank you!

Adaptive Data and Task Joint Scheduling for Multi-Task Learning

哈爾濱工業大學（深圳）
HARBIN INSTITUTE OF TECHNOLOGY, SHENZHEN

鹏城实验室
PENG CHENG LABORATORY