Review : Progressive Layered Extraction(PLE)

A Novel Multi-Task Learning (MTL) Model for Personalized Recommendations

Kaist GSDS Master's Degree
20224314
Hyeongu Kang

CONTENTS

1. Summary of Paper

- Define Problem
- Related work / prior method
- Solving Method
- Experiment
- Result

1. Summary Define Problem

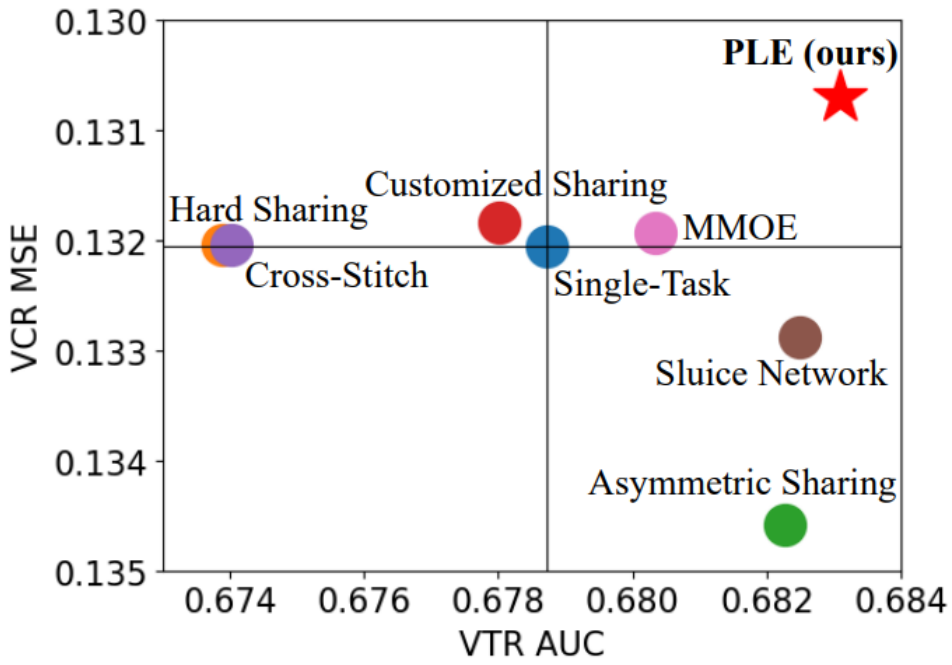
1) Negative Transfer

- Common phenomenon in MTL for loosely correlated tasks
- Performance deterioration in MTL

2) Seesaw Phenomenon

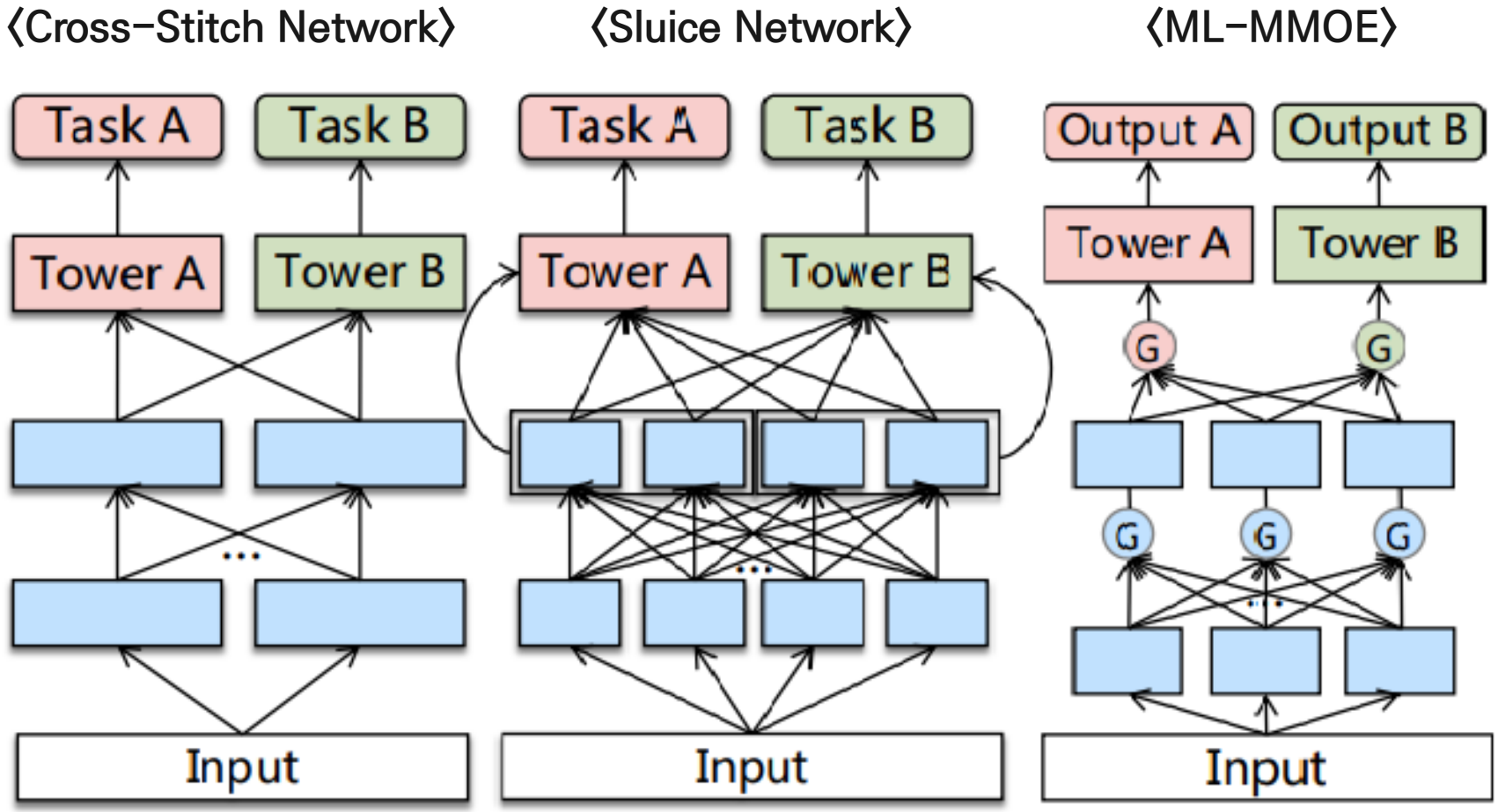
- Occur when tasks are 1) strongly correlated, 2) sample dependent
- Improvement of one task leads to performance deterioration of other tasks

⟨Customized Gate Control(CGC) Model⟩



Negative Transfer and Seesaw Phenomenon are common problem of MTL. Solving both is important.

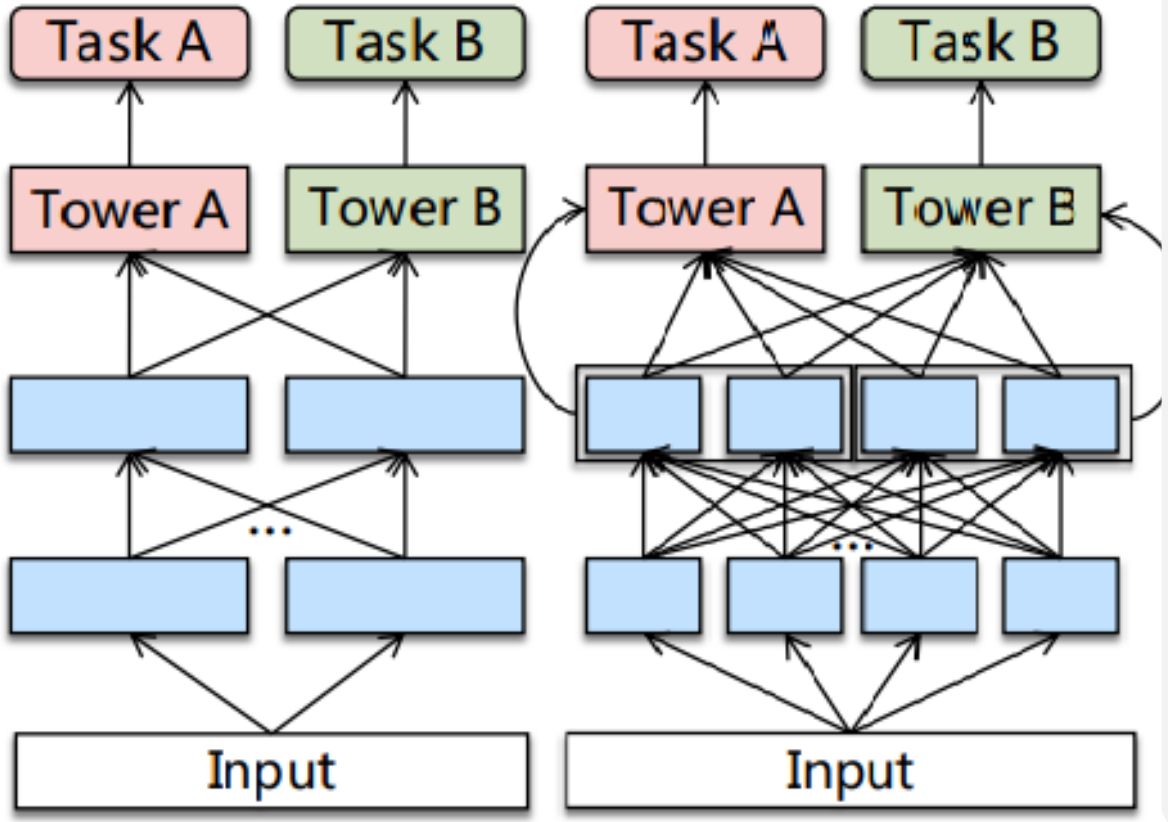
1. Summary Related word / Prior method



Several methods of prior work of MTL
try to solve negative transfer, but neglect the seesaw Phenomenon

1. Summary Related word / Prior method

〈Cross-Stitch Network〉 〈Sluice Network〉



〈Property〉

Learn weights of linear combinations to fuse representations from different tasks
→ Try to deal tasks conflict

But, representations are combined with the same static weights for all samples
→ Can't address Seesaw Phenomenon

* Adaptive combinations of tasks is needed

By reviewing Cross-Stitch Network and Sluice Network,
we can check that adaptive combinations of tasks is needed

1. Summary Related word / Prior method

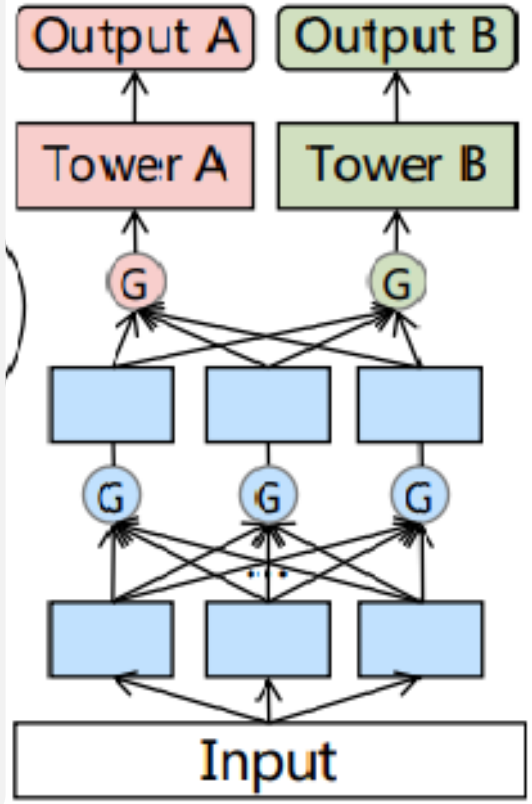
〈Property〉

- Apply gate networks to obtain different fuse weight of task difference among bottom expert
→ 1) Deal with negative transfer and 2) optimize multiple objectives

- But MMOE not consider task correlation
→ Limit the performance of joint optimization

* There is no Task-specific / Task-shared concept

〈ML-MMOE〉



By reviewing ML-MMOE Model, Gate network is effective to fuse different tasks but we also consider task specific / shared concept

1. Summary Related word / Prior method

Trial exist to apply Task specific/shared notion and find good network structure

1) MTAN

- First apply task-specific attention network to fuse shared features selectively
- But different tasks still share the same representation before fusion in attention

→ Tasks are mixed before consider task specific / shared property

2) SNR framework

- Control connections between sub-networks by binary random variables
- Apply NAS to search for the optimal structure

→ Designed with certain simplified assumptions and are not general enough

There are good trial in MTL, but all of prior methods are not enough to become proper and general

1. Summary Solving Method – CGC

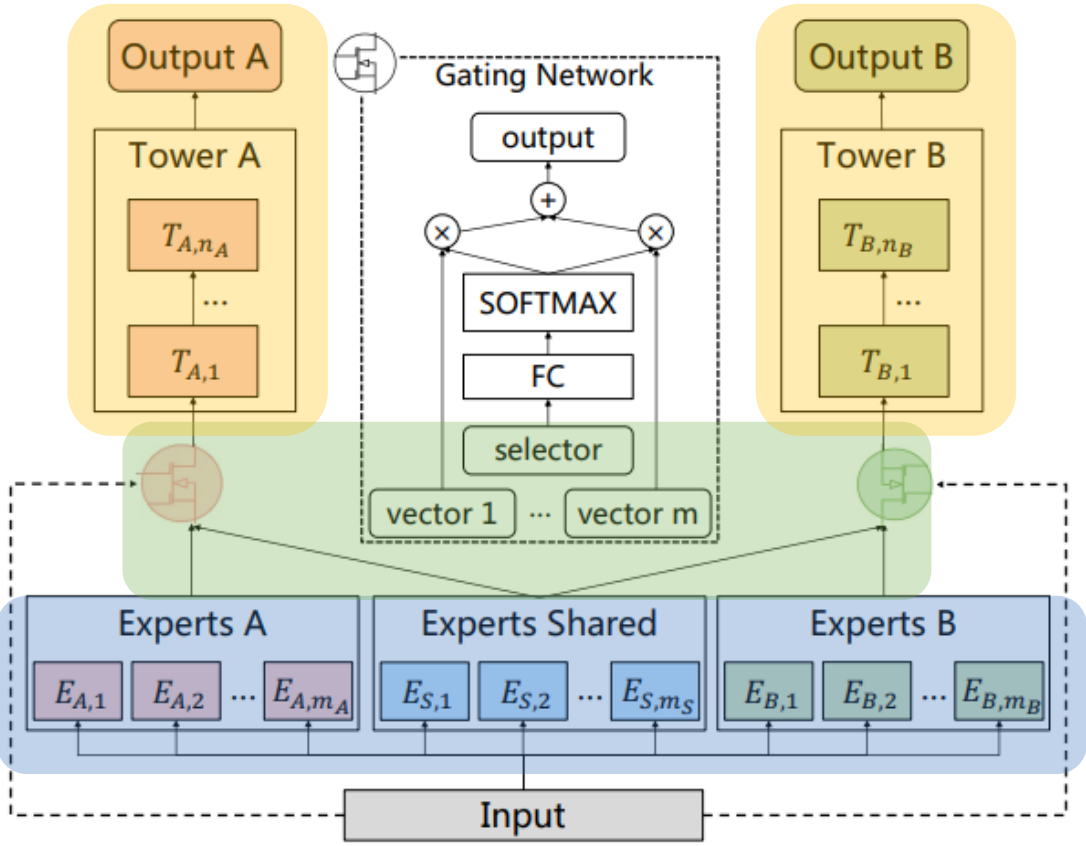
⟨Customized Gate Control⟩

Bottom : Shared & Specific expert modules

Gate : Fuse representation of specific experts and shared experts dynamically

Tower : Absorb knowledge from shared and specific expert through gate

⟨Customized Gate Control(CGC) Model⟩



CGC achieves more flexible balance between tasks and better deals with task conflicts and sample-dependent correlation

1. Summary Solving Method – CGC

〈MMOE vs CGC〉

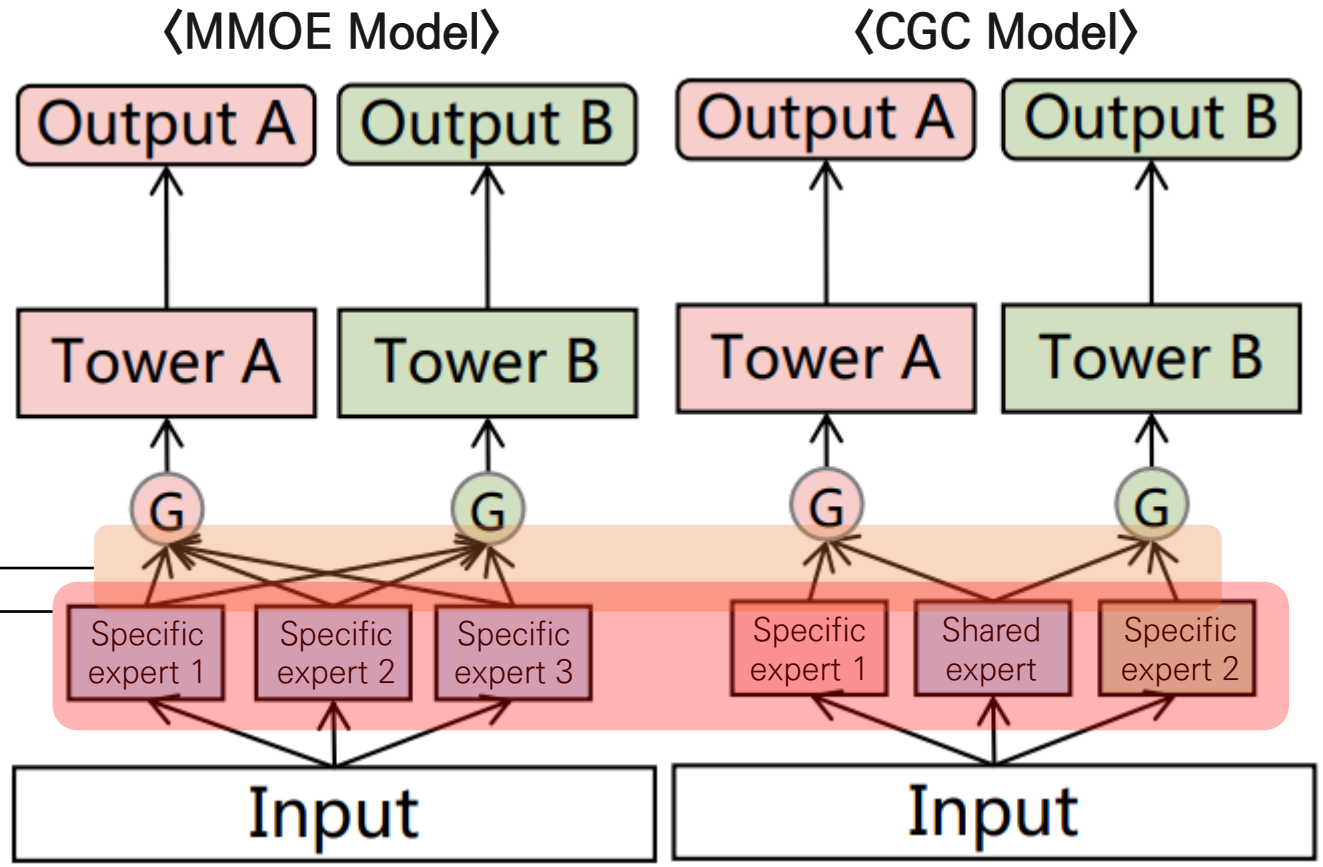
There are two differences

1. Routing

- MMOE connect all of experts
- CGC connect each specific expert and shared expert

2. Bottom ingredient

- MMOE only use task-specific expert
- CGC use task-shared / task-specific experts



The difference of Routing Strategy and Bottom Ingressives allows CGC to well consider the difference and relationship among Tasks.

1. Summary　　Solving Method – PLE

⟨Progressive Layered Extraction⟩

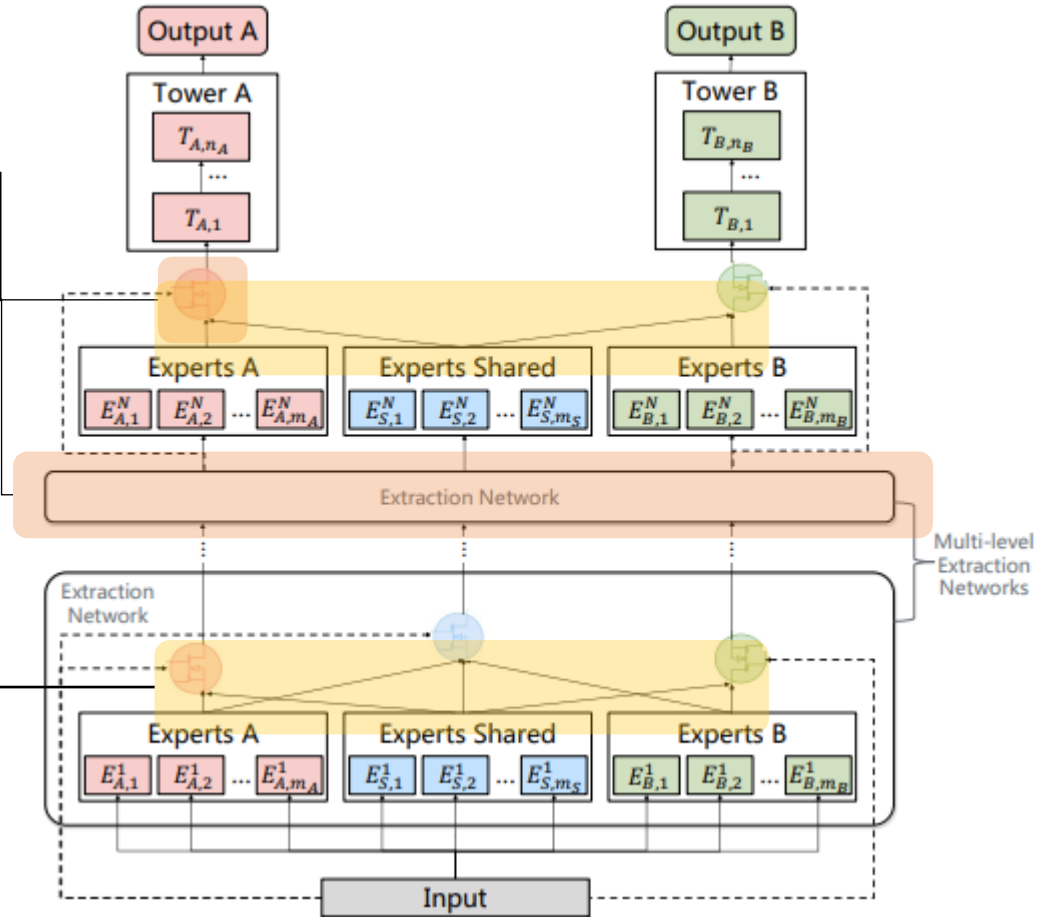
1. Hierarchy structure of information

- Extract higher-level shared information
- Gating network in higher-level take the fusion results of gates in lower-level

2. Progressive separation routing

- Remove connection with other task-specific
- Consider high-level shared semantic representation

⟨Progressive Layered Extraction Model⟩



PLE shape out deeper semantic representations gradually with dividing whether the intermediate representations is shared or task-specific

1. Summary　Solving Method – Joint Loss optimization

〈Joint Loss Optimization for MTL〉

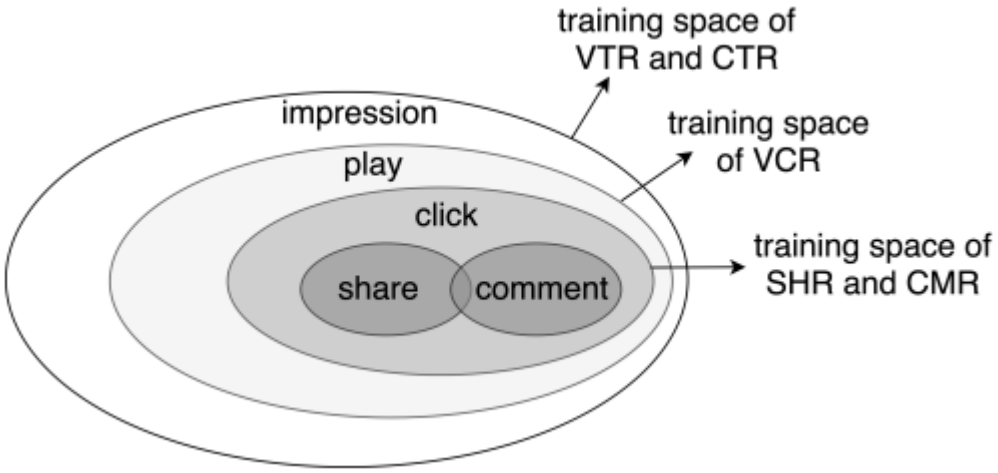
1. Joint Representation

- Solve problem of heterogeneous samples space

ex) Due to sequential users action, share & comment action are considered as heterogeneous

- Consider the union of sample space to train jointly

〈Training Space of Different tasks〉



2. Introducing dynamic weight

- Performance of MTL is sensitive to weight
- Consider of weight tasks fluctuated with time

**By applying Joint Representation and dynamic weight,
solve several issue in making joint optimization of MTL in practice**

1. Summary Experiment

Data
- Industrial dataset from Tencent. 46,926 mil users / 2,682 mil video, 0.996 billion samples in Dataset
- Public Dataset : Synthetic Data, Census-income Dataset, Ali-CCP Dataset

Baseline model
- Single model : Normal, Symmetric, Customized sharing, Cross-stitch, Sluice Network, MMOE
- Multitask model : ML-MMOE

Test Case
- 1) Degree of Correlation between Variables(Strong or Loose)
- 2) Single-task model on Online A/B test
- 3) Single / Multi tasks
- 4) Expert Utilization Analysis between MMOE & PLE

Estimation
- Calculate AUC & MSE of each Indicator(main on VCR, VRT) + MTL Gain

* MTL Gain : Quantitively evaluate the benefit of Multi-task learning over the single-task model

1. Summary Result

⟨Strong Correlation between Variable⟩

Models	AUC VTR	MSE VCR	MTL Gain VTR	MTL Gain VCR
Single-Task	0.6787	0.1321	-	-
Hard Parameter Sharing	0.6740	0.1320	-0.0047	+1.8E-5
Asymmetric Sharing	0.6823	0.1346	+0.0036	-0.0025
Cross-Stitch	0.6740	0.1320	-0.0047	+1.6E-5
Sluice Network	0.6825	0.1329	+0.0038	-0.0008
Customized Sharing	0.6780	0.1318	-0.0007	+0.0002
MMOE	0.6803	0.1319	+0.0016	+0.0001
ML-MMOE	0.6815	0.1329	+0.0028	-0.0009
CGC	**0.6832**	0.1320	**+0.0045**	+3.5E-5
PLE	**0.6831**	**0.1307**	**+0.0044**	**+0.0013**

⟨Loose Correlation between Variable⟩

Models	AUC CTR	MSE VCR	MTL Gain CTR	MTL Gain VCR
Single-Task	0.7379	0.1179	-	-
Cross-Stitch	0.7220	0.1158	-0.0158	+0.0021
Sluice Network	0.7382	0.1157	+0.0004	+0.0021
MMOE	0.7382	0.1175	+0.0003	+0.0004
ML-MMOE	0.7378	0.1169	-0.0001	+0.0010
CGC	0.7398	0.1155	+0.0020	+0.0023
PLE	**0.7406**	**0.1150**	**+0.0027**	**+0.0029**

- identify seesaw phenomenon exist
- PLE's performance is really close to SOTA-method or over-performance

- Even correlation between variables is loose, PLE overscore SOTA-MTL method

Regardless relation between variable is strong or loose,
PLE overscore for most aspect of test then SOTA-MTL method

1. Summary Result

〈Performance on Online A/B test〉

Models	Total View Count	Total Watch Time
Hard Parameter Sharing	-1.65%	-1.79%
Sluice Network	+0.75%	+1.29%
MMOE	+1.94%	+1.73%
ML-MMOE	+1.96%	+1.10%
CGC	+3.92%	+2.75%
PLE	**+4.17%**	**+3.57%**

〈MTL gain of CGC & PLE on Multi Tasks〉

Task Group	Models	MTL Gain VTR	VCR	SHR	CMR
VTR+VCR	CGC	**+0.0131**	**+0.0019**	-0.0001	-
+SHR	PLE	**+0.0132**	**+0.0036**	**+0.0013**	-
VTR+VCR	CGC	**+0.0180**	**+0.0012**	-	+0.0000
+CMR	PLE	**+0.0197**	**+0.0033**	-	+0.0001
VTR+VCR	CGC	**+0.0097**	**+0.0016**	+0.0008	**+0.0012**
+SHR+CMR	PLE	**+0.0128**	**+0.0017**	**+0.0058**	**+0.0080**

- Shows improvement of MTL models over the single-task model on Online total view count & watch time
- CGC & PLE show the benefits of promoting task cooperation
- PLE outperforms CGC in all cases

PLE delivers the best performance in real-world online applications
And it also proved the positive influence of Multi-task learning.

1. Summary Result

MMOE's routing is not Generalization of CGC's

- Distribution between MMOE & ML-MMOE is similar
- But distribution of CGC& PLE is significantly different.

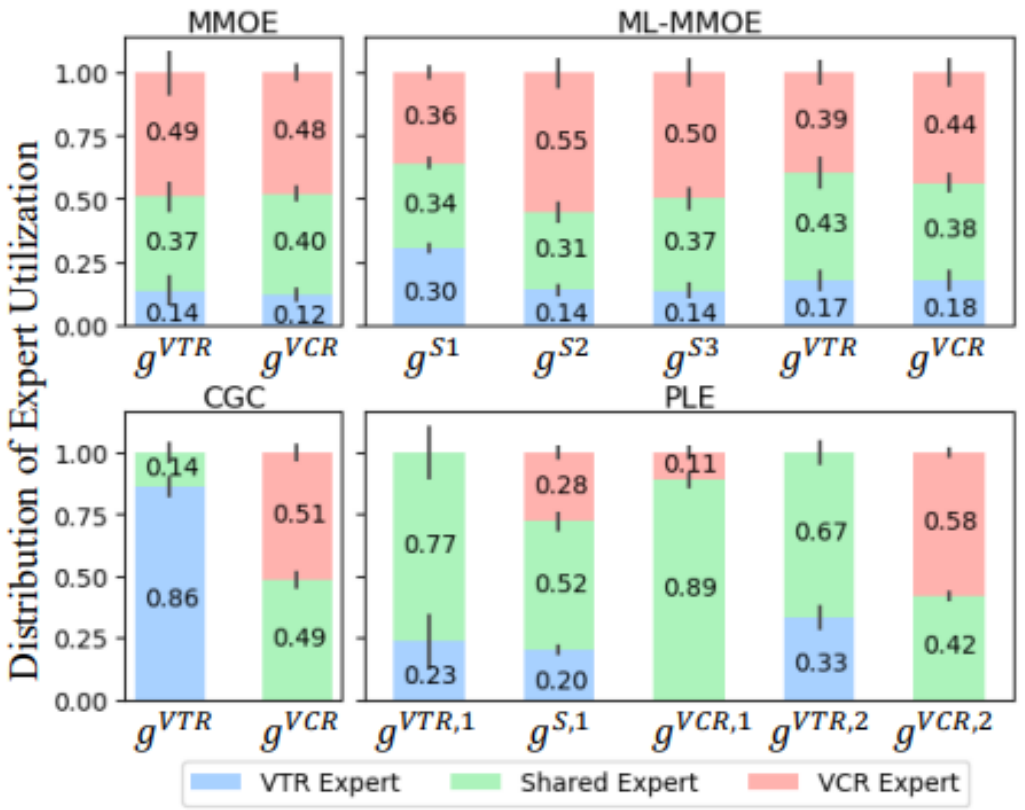
→ CGC's structure helps achieve better differentiation & MMOE hard to converge CGC without prior knowledge

Higher-level deeper representation is Valuable

- PLE performs better than CGC
- Shared expert in PLE have larger influence

→ Show that high-level deeper representation is valuable

⟨Expert Utilization in Gate-Based Models⟩



Expert Utilization show that PLE is good model than ML-MMOE & Higher-level deeper representation is Valuable in MTL

2. Pros & Cons

- Strong point
- Weak point

2. Pros & Cons Strong point

1. Novel Idea(Contribution)

2. Good Motivation

3. Reproducible

4. Convincing Results

2. Pros & Cons Novel Idea(Contribution)

1. Specify the problem “Seesaw Phenomenon” and why this happen.

- Improvement of one task often leads to performance degeneration of the other task
- “Seesaw Phenomenon” occurs when tasks are strongly correlated and especially sample dependent

2. Solving “seesaw Phenomenon” & “Negative transfer” and achieve SOTA’s grade

- Propose CGC and Information routing strategy
- Propose Gating Network in Higher-level extraction Network

3. Show that considering all of relation among tasks not guarantee optimal result

- MMOE’s gate connect all of tasks but, PLE only connect each specific tasks & shared tasks
- PLE overscore MMOE in most case.
- There is significant difference between distribution of PLE and ML-MMOE’s expert Utilization

2. Pros & Cons Good to Motivation

1. Solving common problem of MTL : “Negative transfer” & “Seesaw Phenomenon”

- Since ‘Negative transfer’ frequently occurs in loosely correlated multi tasks.
- On the other side, ‘Seesaw Phenomenon’ frequently occurs in strongly correlated multi tasks, especially sample-dependent situation

→ **In most case of MTL, one of problem easily occurs**

2. 0.1% improve of AUC/MSE contributes significant improvement to online metric

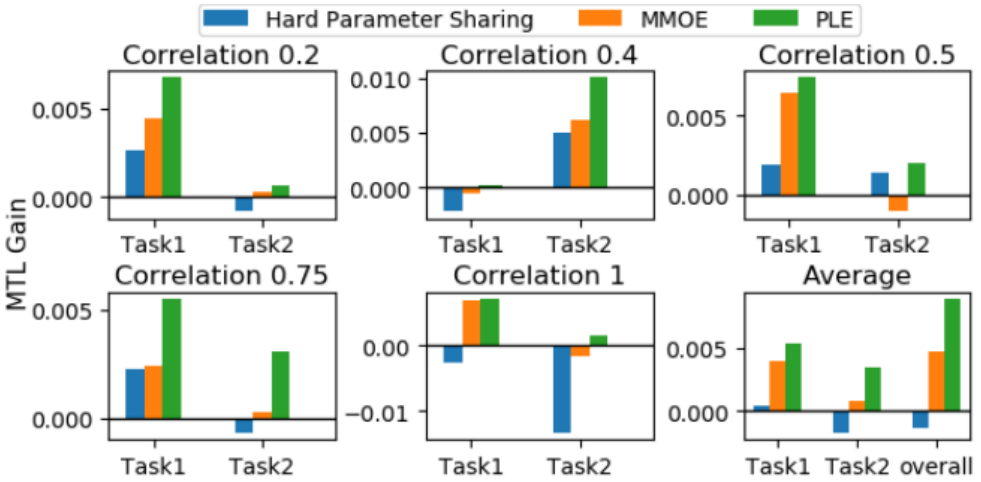
- VCR’s MSE decrease 1.1% & VTR’s AUC increase 0.7%
- It result in increase of 4.17% Total view count & 3.57% of Total watch time in single-task model on Online A/B test

2. Pros & Cons Reproducible

1. The PLE method is also effective in Public Data.

- 1.4 mil samples of Synthetic Data are used
- Prove that PLE consistently performs best for both tasks across different correlations
- Achieves 82.7% increase in MTL gain over MMOE on average

⟨MTL gain on Synthetic Data⟩



- PLE overscore MMOE in public data of Census-income Dataset and Ali-CCP Dataset

⟨Experiment Results on Census-income and Ali-CCP Dataset⟩

Models	**Census-income** Task1		**Census-income** Task2		**Ali-CCP** CTR		**Ali-CCP** CVR	
	AUC	MTL Gain	AUC	MTL Gain	AUC	MTL Gain	AUC	MTL Gain
Single-Task	0.9445	-	0.9923	-	0.6088	-	0.6040	-
MMOE	0.9393	+0.0048	0.9928	+0.0005	0.6094	+0.0006	0.5738	-0.0302
PLE	**0.9522**	**+0.0078**	**0.9945**	**+0.0022**	**0.6112**	**+0.0024**	**0.6097**	**+0.0057**

2. Pros & Cons Convincing Results

1. The PLE method ensured the best performance in most cases with large data

- Using 1 billion samples of Tencent Video recommendation dataset
- Show that there is 'seesaw phenomenon' and prove that PLE can solve this problem.
- Regardless of the correlation degree between tasks, it brought the best performance.

⟨Strong Correlation between Variable⟩

Models	AUC VTR	MSE VCR	MTL Gain VTR	MTL Gain VCR
Single-Task	0.6787	0.1321	-	-
Hard Parameter Sharing	0.6740	0.1320	-0.0047	+1.8E-5
Asymmetric Sharing	0.6823	0.1346	+0.0036	-0.0025
Cross-Stitch	0.6740	0.1320	-0.0047	+1.6E-5
Sluice Network	0.6825	0.1329	+0.0038	-0.0008
Customized Sharing	0.6780	0.1318	-0.0007	+0.0002
MMOE	0.6803	0.1319	+0.0016	+0.0001
ML-MMOE	0.6815	0.1329	+0.0028	-0.0009
CGC	**0.6832**	0.1320	**+0.0045**	+3.5E-5
PLE	**0.6831**	**0.1307**	**+0.0044**	**+0.0013**

⟨Loose Correlation between Variable⟩

Models	AUC CTR	MSE VCR	MTL Gain CTR	MTL Gain VCR
Single-Task	0.7379	0.1179	-	-
Cross-Stitch	0.7220	0.1158	-0.0158	+0.0021
Sluice Network	0.7382	0.1157	+0.0004	+0.0021
MMOE	0.7382	0.1175	+0.0003	+0.0004
ML-MMOE	0.7378	0.1169	-0.0001	+0.0010
CGC	0.7398	0.1155	+0.0020	+0.0023
PLE	**0.7406**	**0.1150**	**+0.0027**	**+0.0029**

- Increase Total view count and Total Watch Time by 4.17% and 3.57% respectively in the online A/B test.

⟨Performance on Online A/B test⟩

Models	Total View Count	Total Watch Time
ML-MMOE	+1.96%	+1.10%
CGC	+3.92%	+2.75%
PLE	**+4.17%**	**+3.57%**

2. Pros & Cons Weak point

1. Additional considerations arising from the introduction of shared experts

1) Calculation volume increase due to updating value
2) Calculation volume increase due to parameter increase
3) Selection problem about appropriate shared expert

2. Lack of explanation

1) Does really 0.1% increase of AUC / MSE statistically meaningful?
2) Why use different metric(AUC/MSE) in figure 3?

2. Pros & Cons Weak point

1. Additional considerations arising from the introduction of shared experts

1) Calculation volume increase due to updating value

- Parameters of Shared experts are affected by all tasks
  while parameters of task-specific expert are only affected by the corresponding specific task

2) Calculation volume increase proportionally with the number of tasks

- Size of the network parameters grows proportionally with respect to the total number of tasks
- When there are n independent tasks, the number of cases of shared experts become very large.
  Even if shared experts only consider relationship between each two tasks,
  the number of shared experts required is $n(n-1)/2$.

3) Selection problem about appropriate shared expert

- Human resource is needed to select appropriate shared experts

2. Pros & Cons Weak point

2. Lack of explanation

1) Does really 0.1% increase of AUC / MSE statistically meaningful?

- Paper only mention "It is worth noting that 0.1% increase of AUC or MSE contributes significant improvement to online metrics in our system, which is also mentioned in [4, 6, 14]."
- [14] : "This is a significant improvement for industrial applications where 0.1% AUC gain is remarkable."
- [6] : use as a reason of [4] such as,
  "Wide & Deep improves AUC by 0.275% (offline) and the improvement of online CTR is 3.9%."
- But when check [4] Cheng et al,
  [6] only describes the transition from
  the base model(Wide) to the Wide & Deep model,
  but does not explain the case of the Deep model
  in which the AUC has decreased but the
  Online Acquisition Gain has increased.

⟨Offline & Online metrics of different models⟩

Model	Offline AUC	Online Acquisition Gain
Wide (control)	0.726	0%
Deep	0.722 ↓	+2.9% ↑
Wide & Deep	0.728	+3.9%

* Figure from [4] Cheng et al

2) Why use different metric(AUC/MSE) in figure 3?

3. Research Idea

3. Research Idea

1.Reduce the amount of calculation

1) Introducing Dropout
2) Check applicability of parallel and distributed MTL
3) Dimension reduction about domain of tasks

2. Expand to multi-domain learning

3. Combination of multi view/domain/tasks

4. More deep hierarchy PLE Model

Q n A