LREC-COLING 2024

北京郵電大學 Beijing University of Posts and Telecommunications

DGoT: Dynamic Graph of Thoughts for Scientific Abstract Generation

Xinyu Ning, Yutong Zhao, Yitong Liu*, Hongwen Yang

School of Information and Communication Engineering

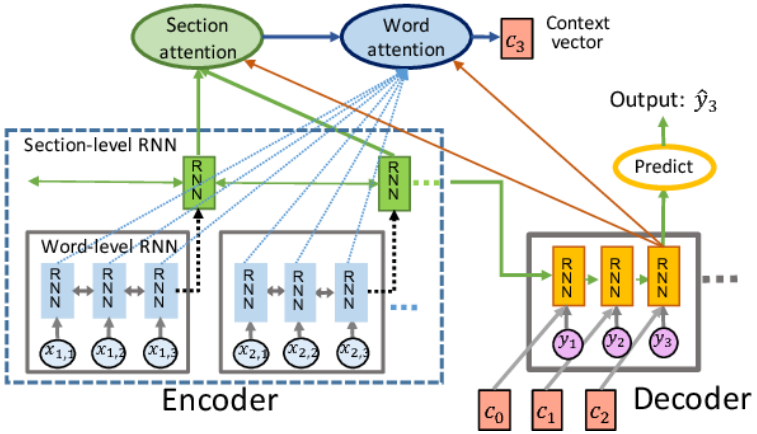
Beijing University of Posts and Telecommunications, China

{nxybupt, zhaoyutong, liuyitong, yanghong}@bupt.edu.cn

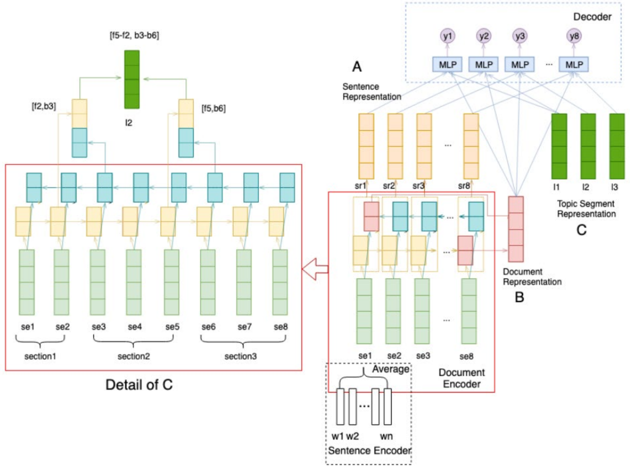
Introduction

- The summarization has achieved tremendous success in natural language processing (NLP) tasks.
- Automating abstract generation encounters challenges due to domain-specific concepts and terminology.

Related works w/o citation information



Cohan et al. (2018)

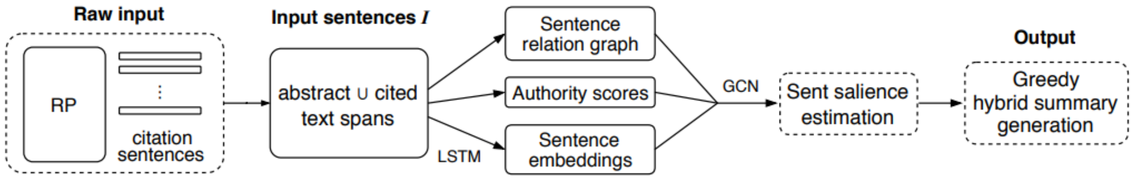


Xiao and Carenini (2019)

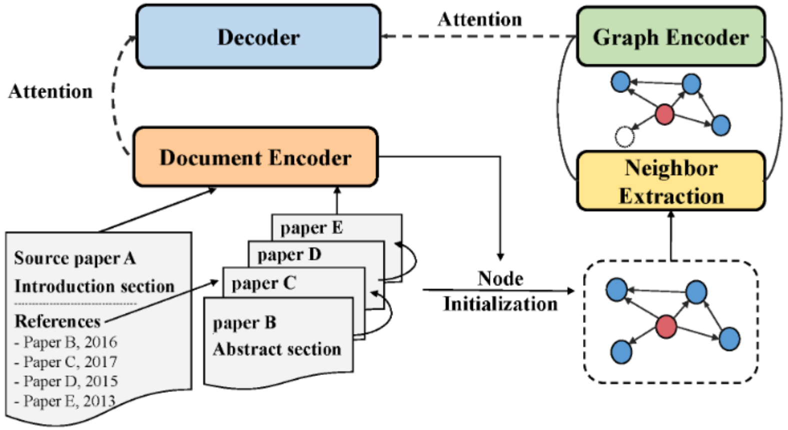
Introduction

- The summarization has achieved tremendous success in natural language processing (NLP) tasks.
- Automating abstract generation encounters challenges due to domain-specific concepts and terminology.

Related works w/ citation information



ScisummNet (Yasunaga et al., 2019)

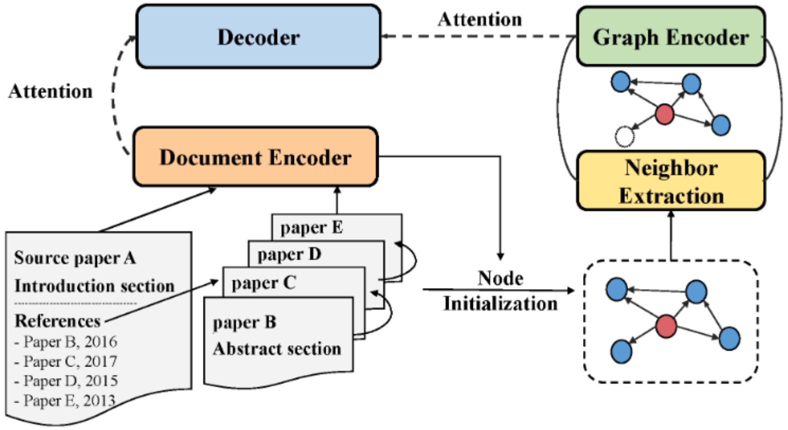


CGSum (An et al., 2021)

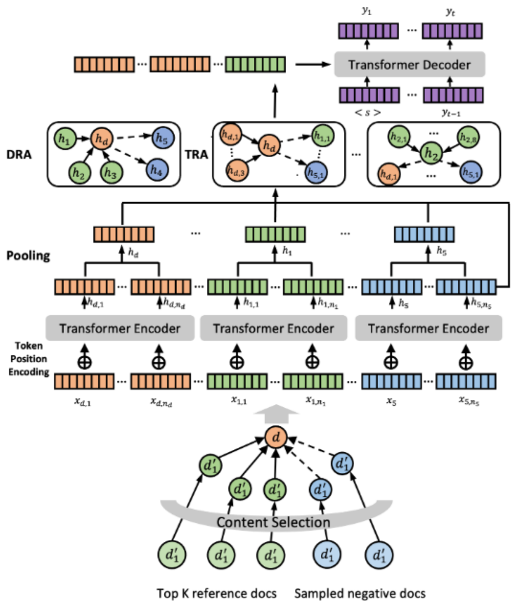
Introduction

- The summarization has achieved tremendous success in natural language processing (NLP) tasks.
- Automating abstract generation encounters challenges due to domain-specific concepts and terminology.

Related works w/ citation information



CGSum (An et al., 2021)



CitationSum (Luo et al., 2023)

Background & Challenges

- **Domain-Specific:** Scientific terminology complicates comprehension.
- **Capturing Complexity:** Extracting the problem, methods, and conclusions from the entire text.
- **Generalization:** Transferring knowledge to unseen domains.
- **Training Cost:** Data, time, and compute-intensive training.

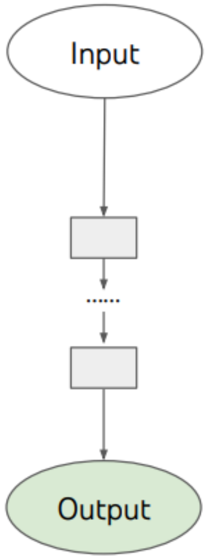
The Rise of Large Language Models (LLMs)

GPT

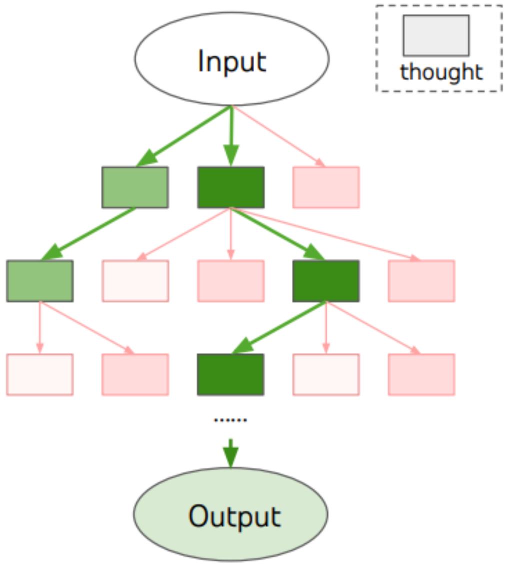
Llama

ChatGLM

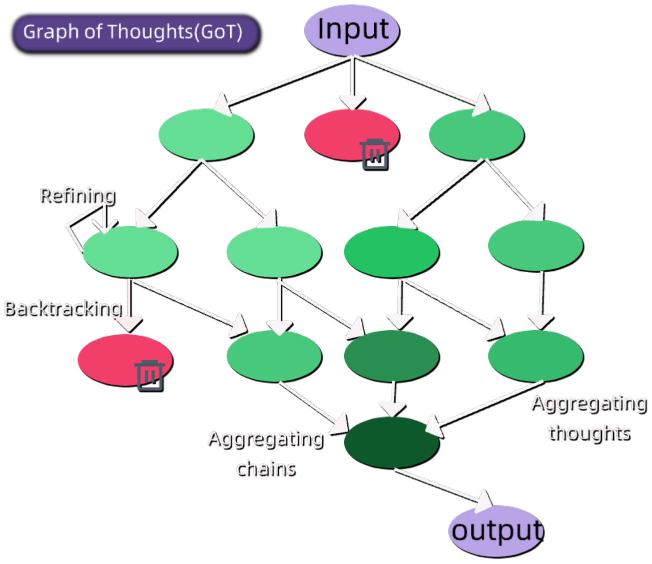
InternLM



CoT (Wei et al., 2022b)



ToT (Yao et al., 2023)



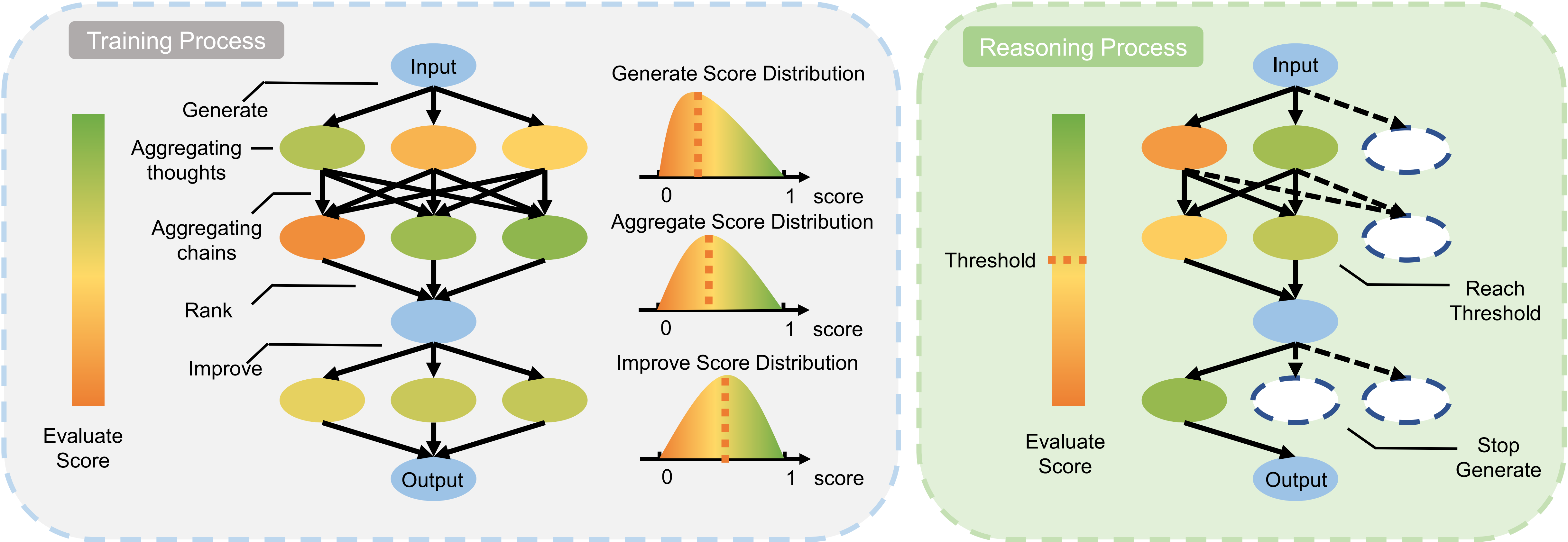
GoT (Besta et al., 2023)

- **Few-Shot Learning:** Adapting output by examples.
- **Prompt Engineering:** Mitigating hallucinations via CoT and combining contents through DoT.

Dynamic Graph of Thoughts

Objectives

- ➢ Enhancing abstract effectiveness
- ➢ Reducing prompt costs



Dynamic Graph of Thoughts

Prompt Framework

Problem Formalization

$$\mathcal{T}_{\mathrm{Gen}}(G, p_\theta) = \mathbf{Y} \sim P(\mathbf{Y} \mid f_{\mathrm{Gen}}(\mathbf{X}, \mathrm{prompt}_{\mathrm{Gen}}))$$

- $G = (V, E)$: reasoning process, where V -> nodes E -> edges
- p_θ: LLMs
- $\mathbf{X}$: basic information
- $\mathrm{prompt}_{\mathrm{Gen}}$: prompt text
- f_{Gen}: prompt framework
- $\mathbf{Y}$: model's output

Prompt Text

$\mathrm{prompt}_{\mathrm{Gen}}$ =

```
"""Please generate the abstract of the article based
on the following information <origin> of the object
article.
    If there are references for the article, the title
and abstract of the reference will also be listed in
<reference>.
    Minimizing redundancy and retaining valid
information as much as possible.
    Only the summaries generated between tags
<Abstract> and </Abstract> are output, and no other
text is output.
<origin>
  {origin}
</origin>
  {reference}
"""
```

$\mathrm{prompt}_{\mathrm{Origin}}$ =

```
"""
<title>
  {title}
</title>
<Introduction>
  {introduction}
</Introduction>
<Other Section>
  {other
section}
</Other
Section>
"""
```

$\mathrm{prompt}_{\mathrm{Reference}}$ =

```
"""
<Reference>
  <title>
    {reference title}
  </title>
  <Abstract>
    {reference
abstract}
  </Abstract>
</Reference>
"""
```

Basic Information

$\mathrm{X} = [x_1, x_2, x_3, x_4, x_5]$

x_1 = Original Title

x_2 = Original Introduction

x_3 = Original Other Section

x_4 = Reference Title

x_5 = Reference Abstract

Prompt Framework

x_1 → {title}

x_2 → {introduction}

x_3 → {other section}

x_4 → {reference title}

x_5 → {reference abstract}

$\mathrm{prompt}_{\mathrm{Origin}}$ → {origin}

$\mathrm{prompt}_{\mathrm{Reference}}$ → {reference}

$\mathrm{prompt} = f_{\mathrm{Gen}}(\mathbf{X}, \mathrm{prompt}_{\mathrm{Gen}})$

Dynamic Graph of Thoughts

Prompt Framework

Problem Formalization

$$\mathcal{T}_{Gen}(G, p_\theta) = \mathbf{Y} \sim P(\mathbf{Y} \mid f_{Gen}(\mathbf{X}, \text{prompt}_{Gen}))$$

3 Types of Transformation

$$\mathcal{T}_{Gen}(G, p_\theta) \mid \mathcal{T}_{Agg}(G, p_\theta) \mid \mathcal{T}_{Impr}(G, p_\theta)$$

Evaluator: ROUGE score

$$\varepsilon(p_\theta, S)$$

Prompt Text

prompt_{Gen} =

```
"""Please generate the abstract of the article based on the following information <origin> of the object article.
    If there are references for the article, the title and abstract of the reference will also be listed in <reference>.
    Minimizing redundancy and retaining valid information as much as possible.
    Only the summaries generated between tags <Abstract> and </Abstract> are output, and no other text is output.
<origin>
  {origin}
</origin>
  {reference}
"""
```

prompt_{Origin} =

```
"""
<title>
  {title}
</title>
<Introduction>
  {introduction}
</Introduction>
<Other Section>
  {other section}
</Other Section>
"""
```

$\text{prompt}_{Reference}$ =

```
"""
<Reference>
  <title>
    {reference title}
  </title>
  <Abstract>
    {reference abstract}
  </Abstract>
</Reference>
"""
```

Basic Information

$\mathrm{X} = [x_1, x_2, x_3, x_4, x_5]$

x_1 = Original Title

x_2 = Original Introduction

x_3 = Original Other Section

x_4 = Reference Title

x_5 = Reference Abstract

Prompt Framework

$x_1 \rightarrow$ {title}

$x_2 \rightarrow$ {introduction}

$x_3 \rightarrow$ {other section}

$x_4 \rightarrow$ {reference title}

$x_5 \rightarrow$ {reference abstract}

$\text{prompt}_{Origin} \rightarrow$ {origin}

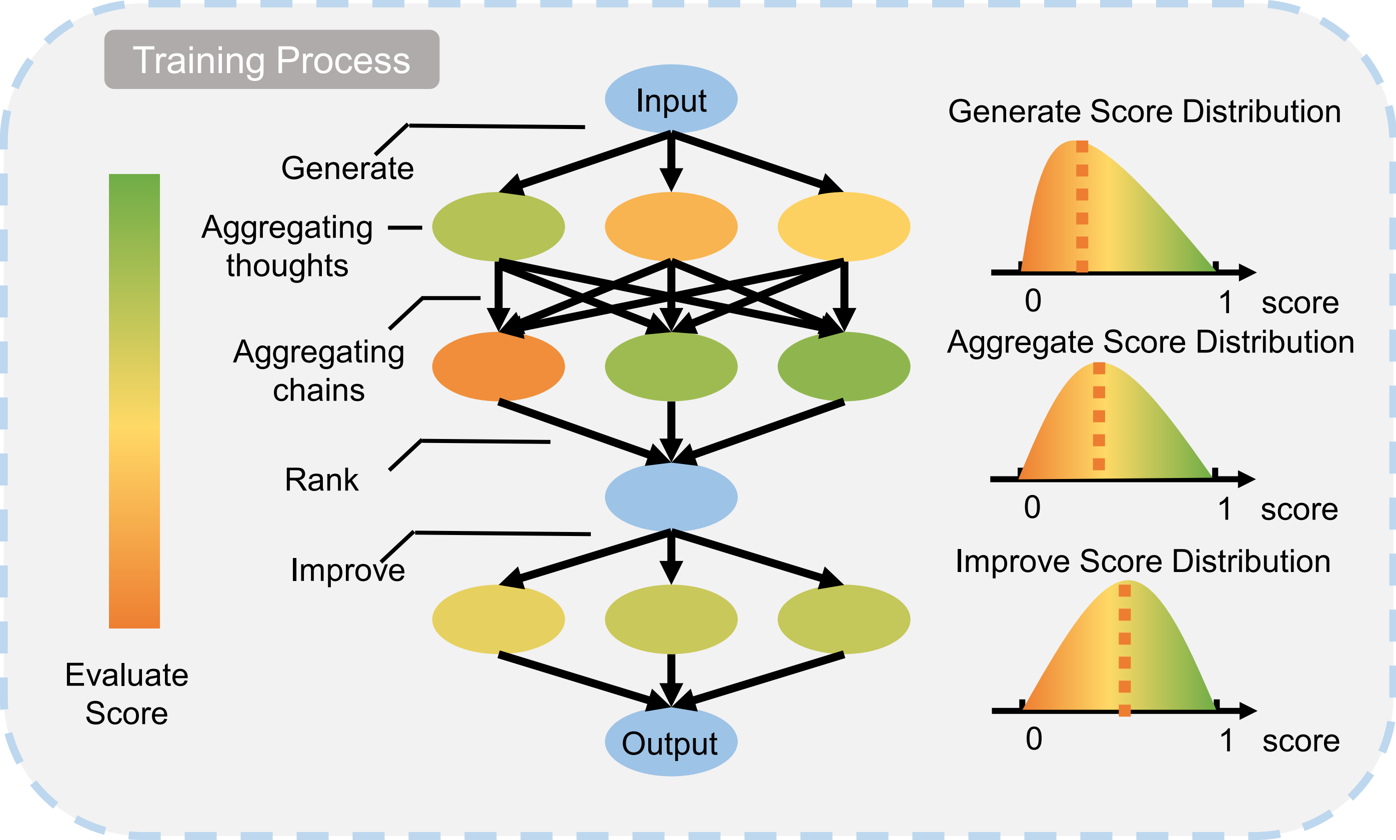
$\text{prompt}_{Reference} \rightarrow$ {reference}

$\text{prompt} = f_{Gen}(\mathbf{X}, \text{prompt}_{Gen})$

Dynamic Graph of Thoughts

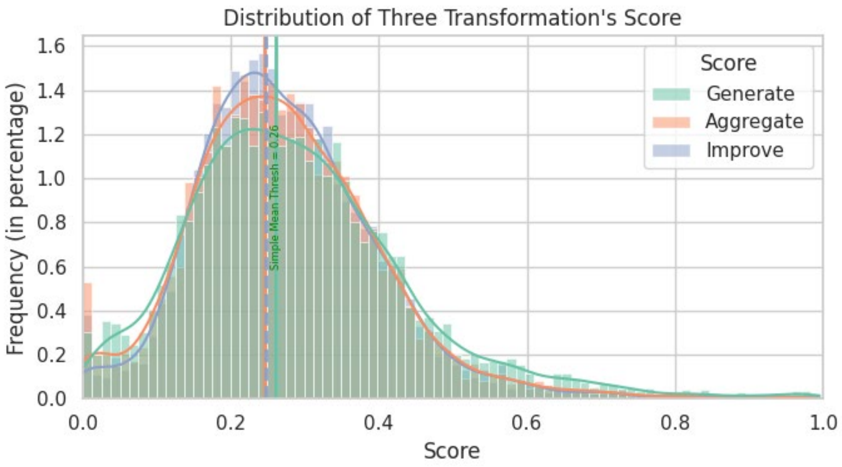
Training Process

Calculate Statistical characteristics of score distributions as **thresholds** for further processing.



Two Types of Thresholds

Simple mean threshold

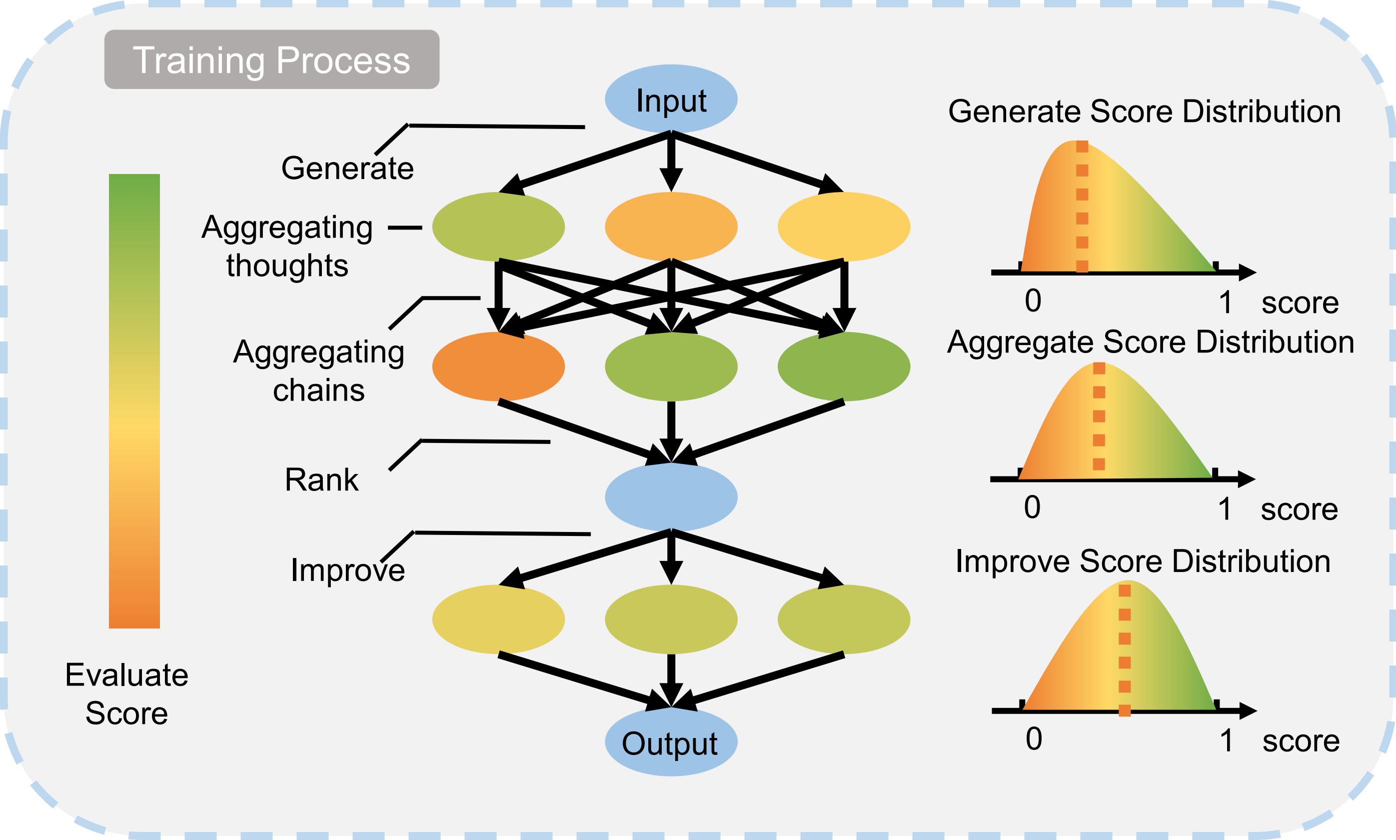


$$\mathrm{Thresh}_{\mathrm{Simple}}=\mu_{\mathrm{score}}$$

Dynamic Graph of Thoughts

Training Process

Calculate Statistical characteristics of score distributions as **thresholds** for further processing.



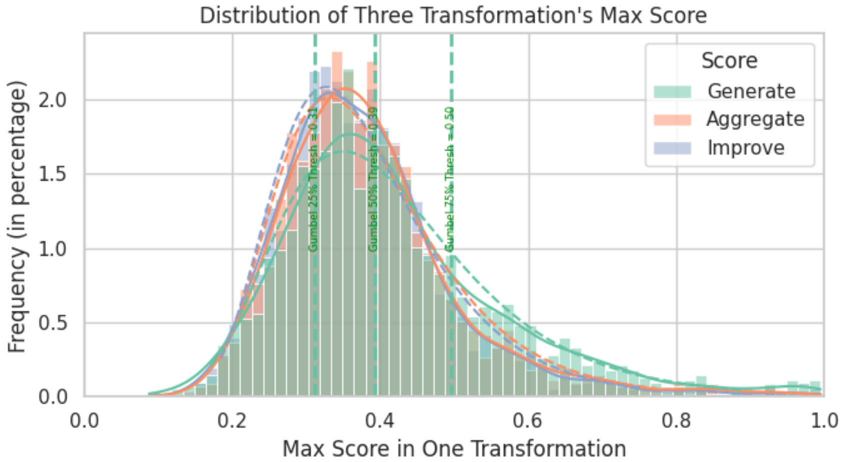
Two Types of Thresholds

Gumbel threshold

CDF: $F(x;\mu,\beta) = e^{-e^{-(x-\mu)/\beta}}$

where $\beta^2 = \frac{6\sigma_{\max}^2}{\pi^2}$ $\quad \mu = \mu_{\max} - \gamma\beta$

$$\mathrm{Thresh}_{\mathrm{Gumbel}} = \mu - \beta \ln(-\ln p_{\mathrm{Thresh}})$$



Dynamic Graph of Thoughts

Reasoning Process

- Threshold function: $H = [H_{\text{Simple}}, H_{\text{Gumbel}}]$
- Evaluator: $\varepsilon(p_\theta, S)$

Dynamic Generate module

$$\mathcal{T}_{\text{DG}}(G, p_\theta, H)$$

Dynamic Aggregate module

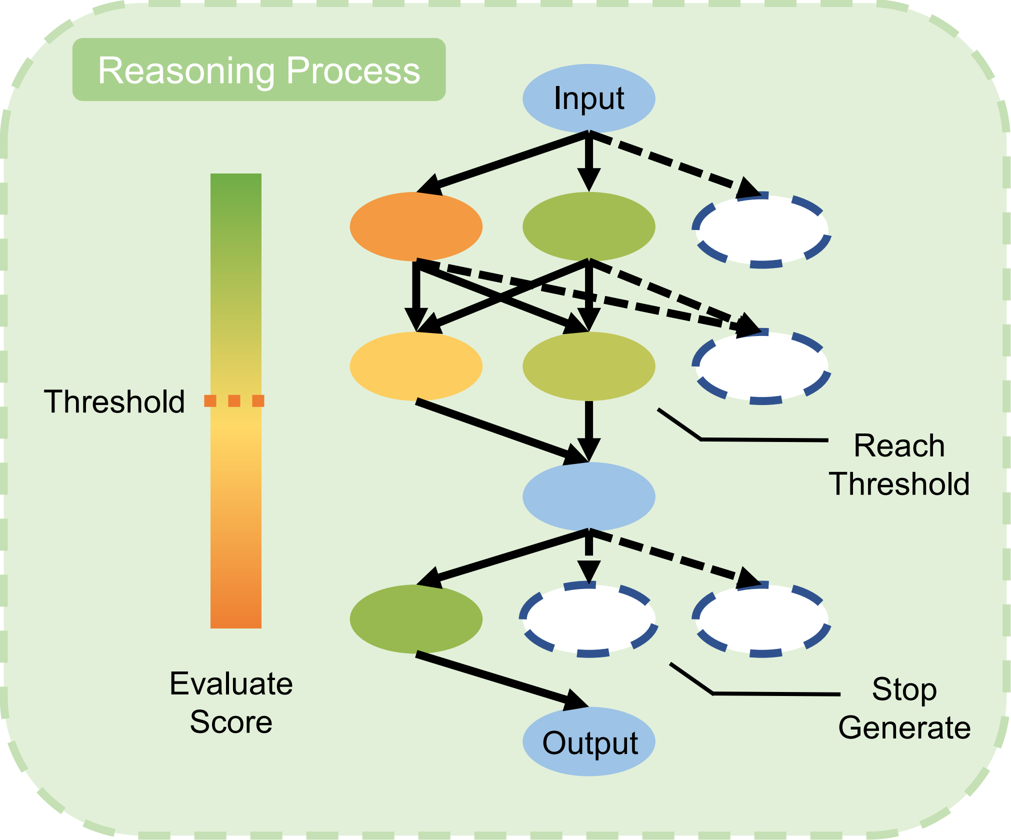
$$\mathcal{T}_{\text{DA}}(G, p_\theta, H)$$

Dynamic Improve module

$$\mathcal{T}_{\text{DI}}(G, p_\theta, H)$$

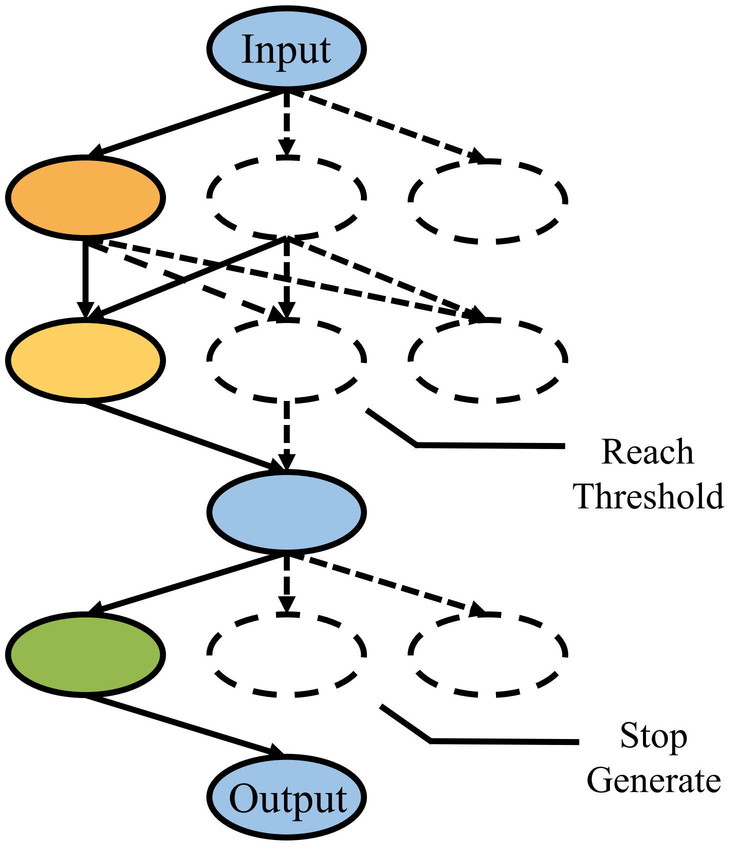
Ranking module

$\mathcal{R}(G, p_\theta, h)$ -> Top h answers



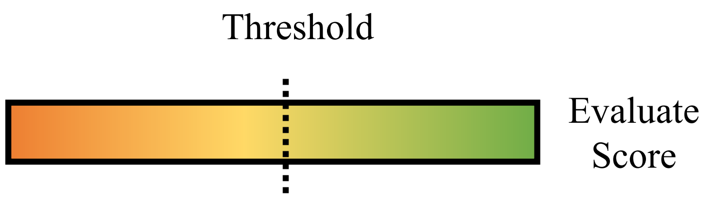
Dynamic Graph of Thoughts

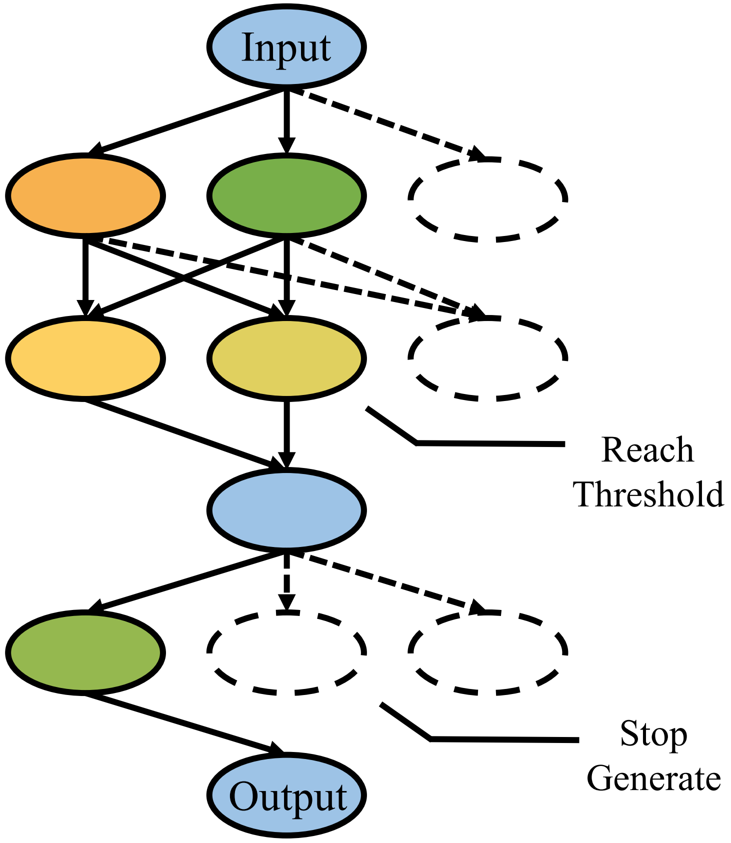
Reasoning Process



Dynamic Graph of Thoughts

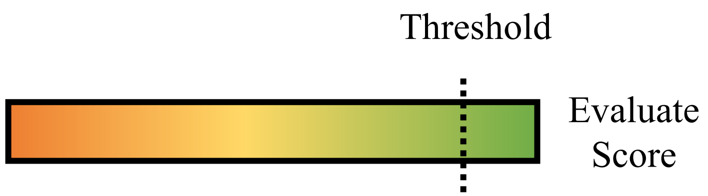
Reasoning Process

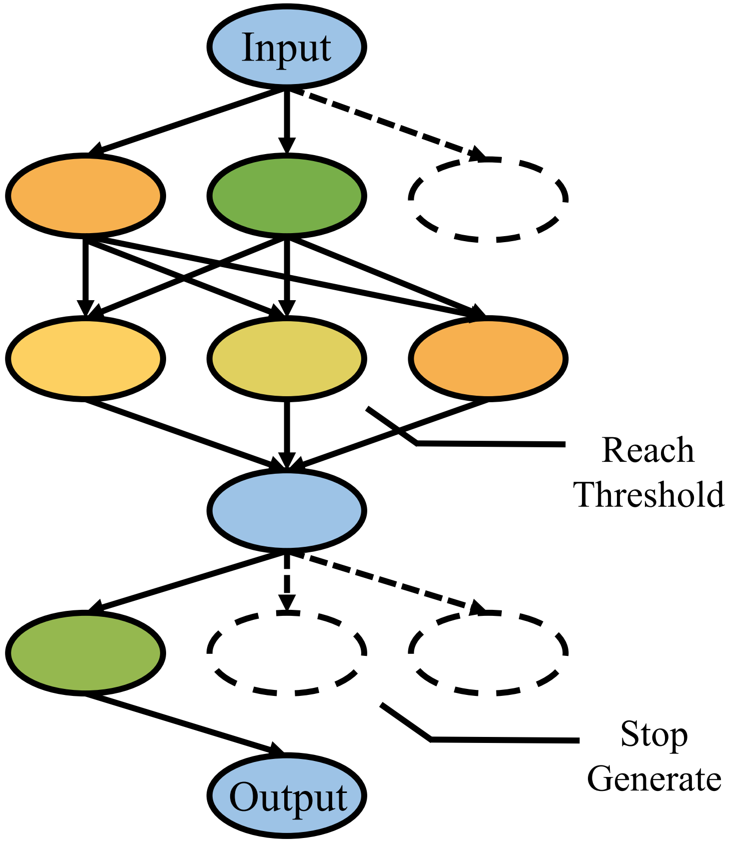




Dynamic Graph of Thoughts

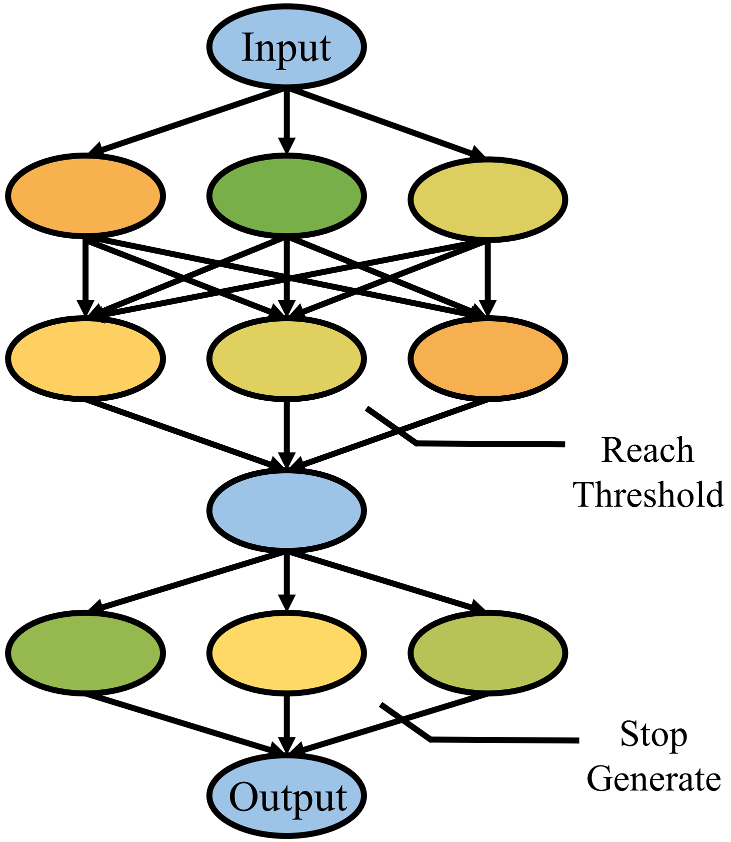
Reasoning Process





Dynamic Graph of Thoughts

Reasoning Process



Main Experimental Result

Setup

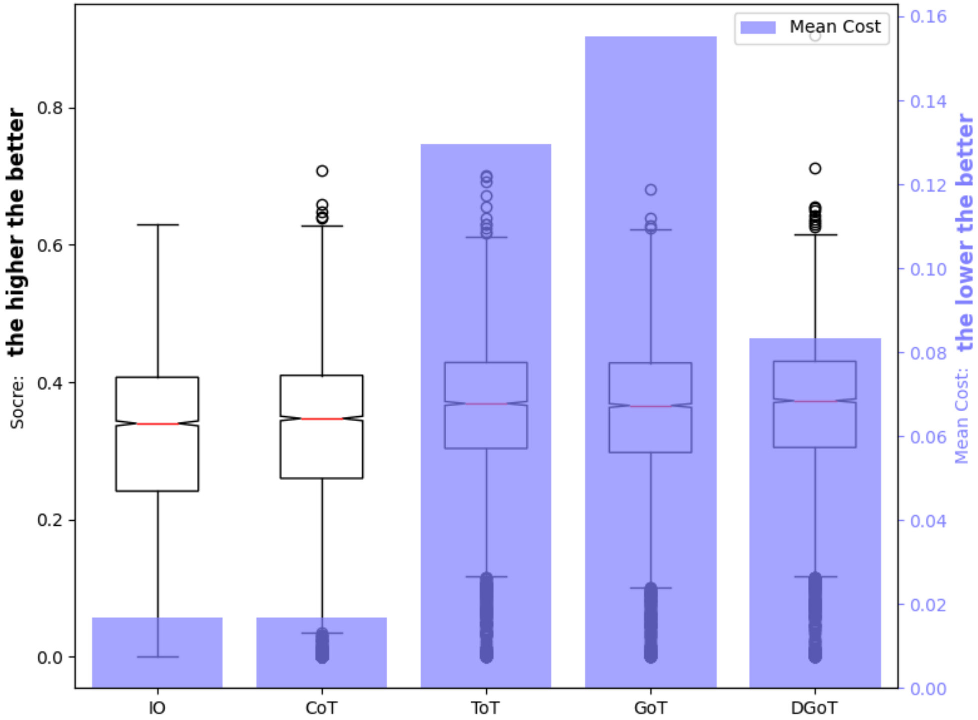
- Datasets: PubMedCite Inductive (Luo et al., 2023)
- Model: ChatGLM2-6B
    - Top p = 0.7
    - Temperature T = 0.7
- GPU: 24G 3090
- Prompt setting
    - Input length: 20000
    - Level: L = 3
    - Branching factor: k = 3

Main Experimental Result

Compared with other prompt approaches

Method	R-1	R-2	R-L	Prompt Tokens	Response Tokens	Cost	Cost-effectiveness
IO	0.303	0.081	0.166	10660.79	402.79	0.0167	
CoT	**0.314**	0.083	0.171	10644.81	358.77	**0.0166**	
ToT	0.356(0.042)	0.098	0.190	82850.63	2606.48	0.1294 (0.1128)	2.686
GoT	0.354(0.040)	**0.099**	0.190	99184.15	3219.40	0.1552 (0.1386)	3.465
DGoT	**0.358**(0.044)	**0.099**	**0.192**	**53414.97**	**1565.12**	**0.0833** (0.0667)	**1.516**

- **Quality:** DGoT achieves the highest LOUGE score.
- **Cost:** 43.7% to 56.4% cost-effectiveness compared to other multi-round query prompt approaches.

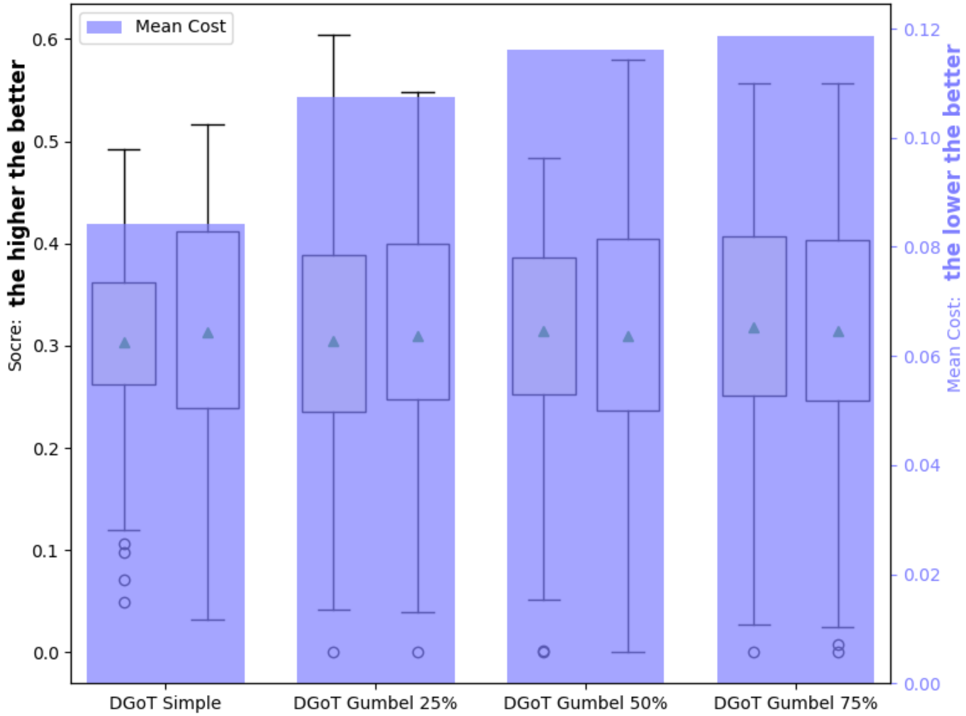


Cost-effectiveness is defined as the cost required to improve the performance of a unit metric compared to a baseline method.
Therefore, the smaller the cost-effectiveness, the better.

Main Experimental Result

Effects of threshold function H setting

Setting	Intro. R-1	Abst. R-1	Abst. R-2	Abst. R-L	Cost
Simple	0.303	0.313	0.075	**0.176**	**0.0841**
25%	0.305	0.309	0.070	0.173	0.1075
50%	0.314	0.309	0.075	0.170	0.1161
75%	**0.317**	**0.314**	**0.078**	0.175	0.1186



- Increasing the threshold leads to higher Introduction R-1 scores.
- But Abstract ROUGE scores do not show a linear trend.

Conclusion

- Introduced a **Dynamic Graph of Thought** (DGoT) prompt method, dynamically adjusting graph structure to minimize language model costs.
- Established a **threshold-setting** mechanism for the DGoT evaluation function to provide a reference for **performance and cost trade-offs**.
- Experimental results demonstrate that our approach achieves the best **cost-effectiveness** in scientific literature abstract generation compared to other multi-round prompt methods.

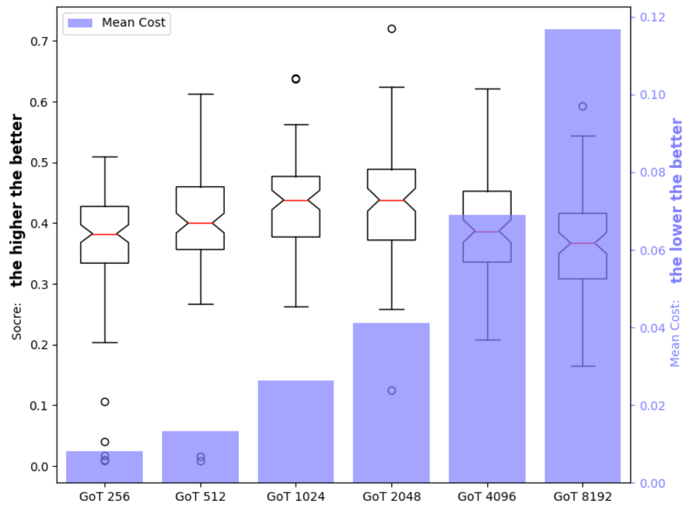
Appendix - Further studies

- Other Potential Influencing Factors on the Results
  - **Effect of Prompt Length**
  - **Effect of Branching Factors**
  - **Results under Optimal Prompt Length**
- Prompt Framework for Transformations

Appendix - Further studies

Effect of Prompt Length

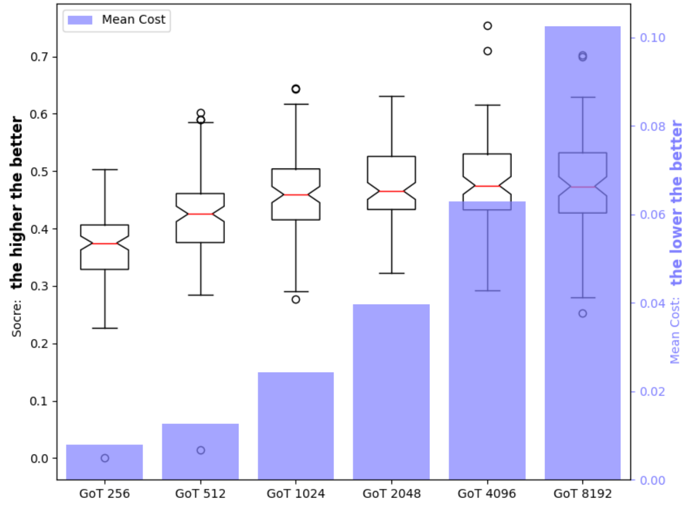
- Longer input is not necessarily better:
  - truncating input may lose citation information.



Performance of ChatGLM2

Model/ Method	Prompt Length	Cut Ratio	Intro. R-1	Abst. R-1	Abst. R-2	Abst. R-L	Prompt Tokens	Resp. Tokens	Infer. Time(s)	Cost
Chat-GLM2/ GoT	256	0.993	0.339	0.367	0.091	0.189	2623.29	2062.74	76.61	0.008
	512	0.973	0.456	0.400	0.111	0.192	5360.43	2613.06	96.05	0.013
	1024	0.852	0.517	0.431	0.144	0.216	12881.78	3545.09	126.69	0.026
	2048	0.775	**0.520**	0.435	0.154	0.221	23189.33	3223.30	120.00	0.041
	4096	0.693	0.510	0.391	0.132	0.203	40675.92	3979.97	154.09	0.068
	8192	0.465	0.436	0.366	0.104	0.191	70249.50	5711.58	282.49	0.116
Intern-LM2/ GoT	256	0.993	0.317	0.368	0.097	0.187	3368.73	1440.99	37.16	0.007
	512	0.973	0.450	0.418	0.125	0.200	5949.60	1892.68	47.75	0.012
	1024	0.830	0.418	0.456	0.164	0.235	13642.57	1894.43	48.89	0.024
	2048	0.740	0.447	0.471	0.176	0.240	23849.84	1965.58	53.24	0.039
	4096	0.670	0.447	**0.482**	**0.190**	**0.259**	39069.64	2139.46	60.76	0.062
	8192	0.436	0.422	0.479	0.183	0.250	65586.77	2049.10	72.41	0.102

Note: Experiment conducted on the first 100 training set data.



Performance of InternLM2

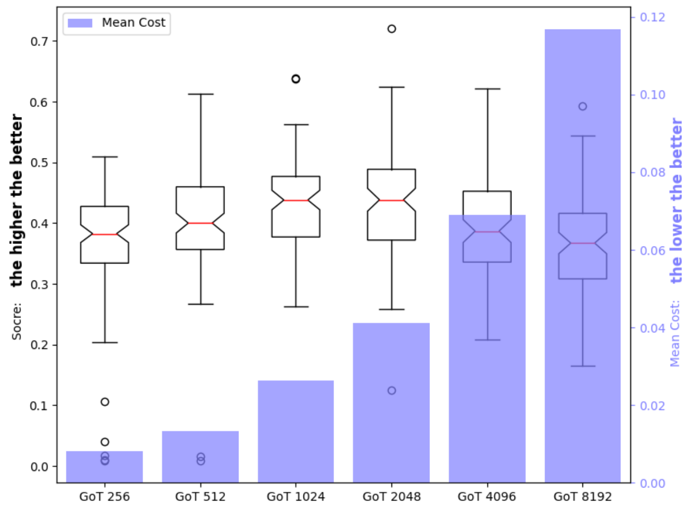
Appendix - Further studies

Effect of Prompt Length

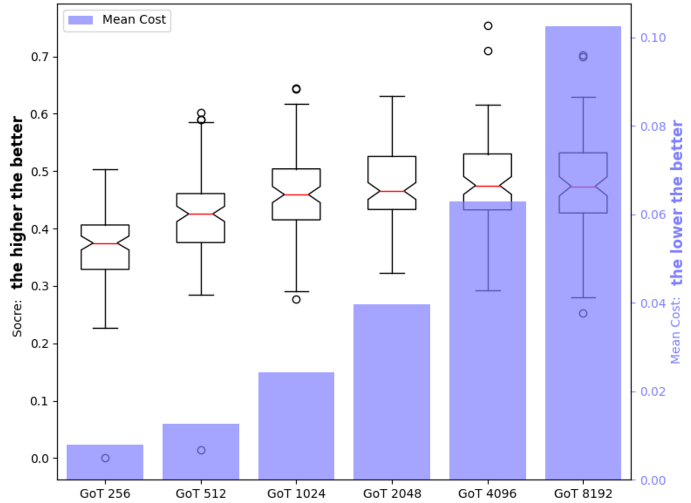
- Models differ in performance:
  - ChatGLM2 retains introduction information well, while InternLM2 excels in abstract generation.

Model/ Method	Prompt Length	Cut Ratio	Intro. R-1	Abst. R-1	Abst. R-2	Abst. R-L	Prompt Tokens	Resp. Tokens	Infer. Time(s)	Cost
Chat-GLM2/ GoT	256	0.993	0.339	0.367	0.091	0.189	2623.29	2062.74	76.61	0.008
	512	0.973	0.456	0.400	0.111	0.192	5360.43	2613.06	96.05	0.013
	1024	0.852	0.517	0.431	0.144	0.216	12881.78	3545.09	126.69	0.026
	2048	0.775	**0.520**	0.435	0.154	0.221	23189.33	3223.30	120.00	0.041
	4096	0.693	0.510	0.391	0.132	0.203	40675.92	3979.97	154.09	0.068
	8192	0.465	0.436	0.366	0.104	0.191	70249.50	5711.58	282.49	0.116
Intern-LM2/ GoT	256	0.993	0.317	0.368	0.097	0.187	3368.73	1440.99	37.16	0.007
	512	0.973	0.450	0.418	0.125	0.200	5949.60	1892.68	47.75	0.012
	1024	0.830	0.418	0.456	0.164	0.235	13642.57	1894.43	48.89	0.024
	2048	0.740	0.447	0.471	0.176	0.240	23849.84	1965.58	53.24	0.039
	4096	0.670	0.447	**0.482**	**0.190**	**0.259**	39069.64	2139.46	60.76	0.062
	8192	0.436	0.422	0.479	0.183	0.250	65586.77	2049.10	72.41	0.102

Note: Experiment conducted on the first 100 training set data.



Performance of ChatGLM2

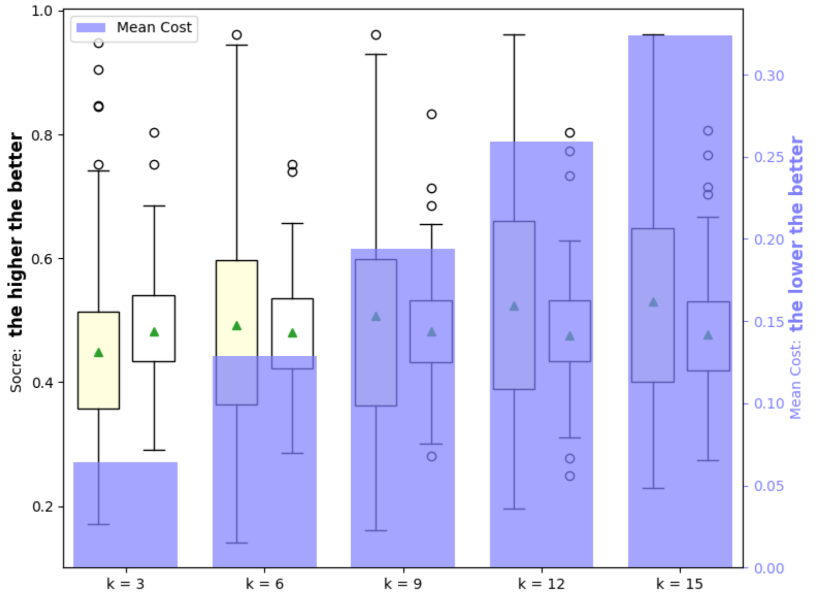


Performance of InternLM2

Appendix - Further studies

Effect of Branching Factors

- The ROUGE scores of introduction and abstract are in trade-off.
- As branching factor k improves,
  - Intro. R-1 scores improve,
  - Abst. R-1 scores do not improve.



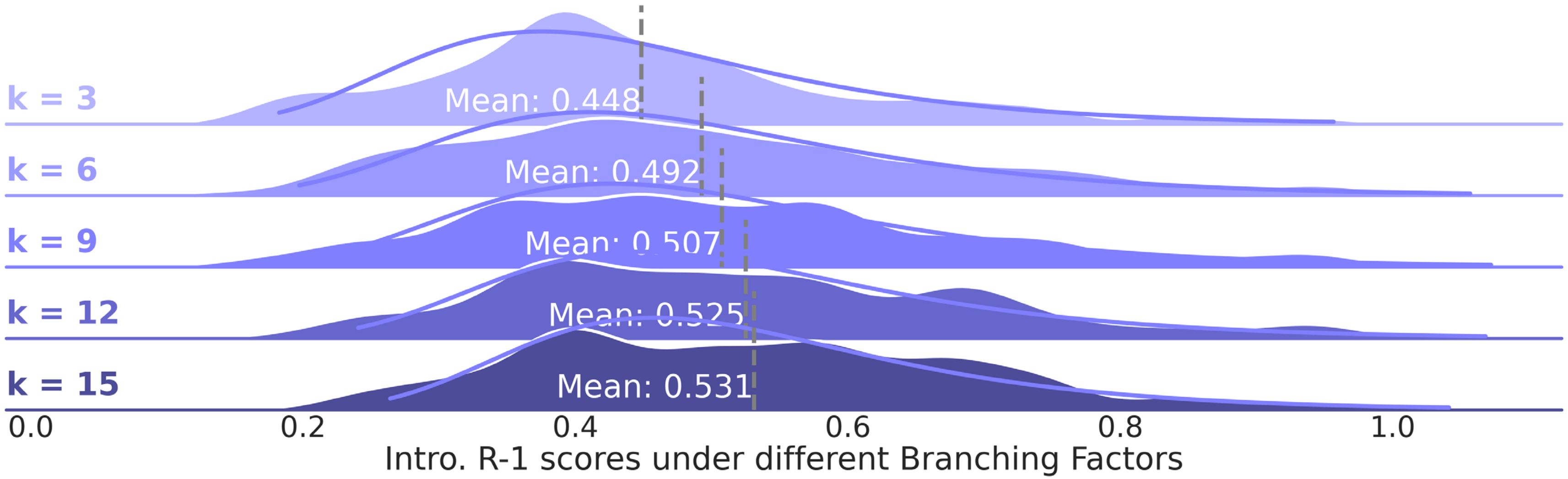
k	Introduction R-1	Abst. R-1	Abst. R-2	Abst. R-L	Prompt Tokens	Resp. Tokens	Infer. Time(s)	Cost	C/E
3	0.448	**0.481**	**0.191**	**0.264**	39940.46	2088.17	60.86	0.064	
6	0.492(0.044)	0.480	0.187	0.250	80091.36	4360.62	122.64	0.128(0.064)	**1.454**
9	0.507(0.059)	0.481	0.188	0.255	120196.65	6761.61	187.29	0.193(0.129)	2.186
12	0.524(0.076)	0.475	0.190	0.255	160543.10	9139.17	251.58	0.259(0.195)	2.565
15	**0.530**(0.082)	0.478	0.187	0.251	200225.20	11821.69	322.74	0.323(0.259)	3.158

Note: Experiment conducted on the first 100 training set data, using InternLM2.

Appendix - Further studies

Effect of Branching Factors

- ➢ Improving scores through additional inquiries demonstrates marginal utility.
  - ➢ The number of agents follows Scaling Laws.
  - ➢ There are limits to the performance gains it can bring.



k	C/E
3	
6	**1.454**
9	2.186
12	2.565
15	3.158

Note: Experiment conducted on the first 100 training set data, using InternLM2.

Appendix - Further studies

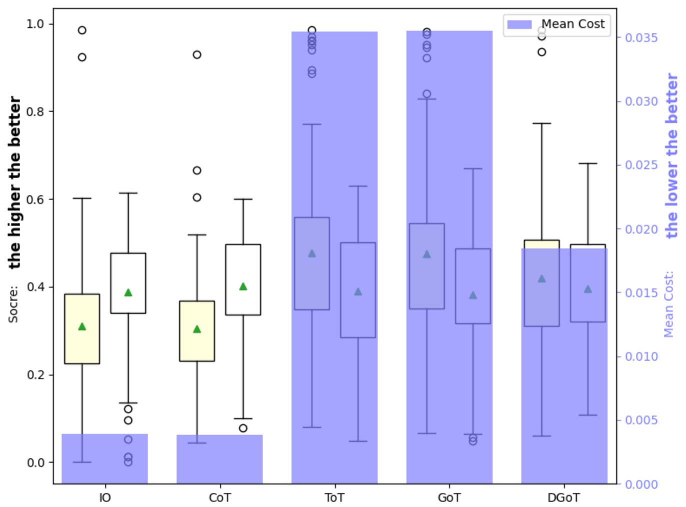
Results under Optimal Prompt Length

- The few-shot capability of large models has the potential to surpass the performance of previous best models †.

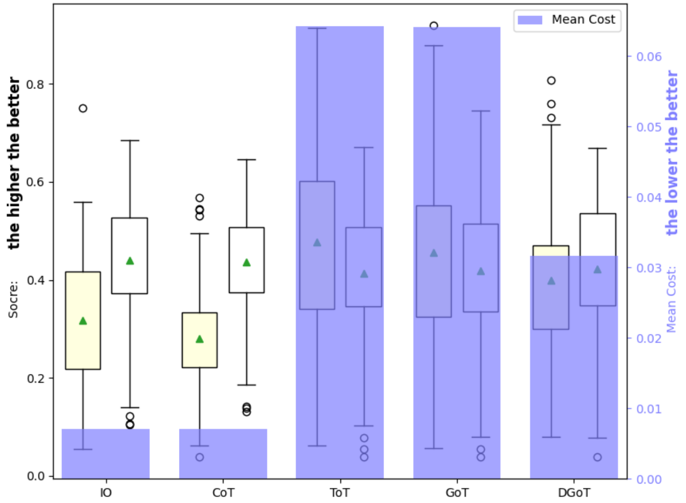
Model	Method	Introduction R-1	Abst. R-1	Abst. R-2	Abst. R-L	Prompt Tokens	Resp. Tokens	Infer. Time(s)	Cost	C/E
Chat-GLM2	IO	**0.311**	0.387	0.126	0.204	2274.14	233.51	10.53	0.003	
	CoT	0.305	**0.401**	**0.129**	**0.213**	2269.77	214.12	9.84	0.003	
	ToT	**0.476**(0.171)	0.390	**0.130**	**0.199**	20465.34	2376.15	96.11	0.035(0.032)	0.187
	GoT	0.475(0.170)	0.382	0.128	0.196	20409.60	2442.31	97.25	0.035(0.032)	0.188
	DGoT	0.418(0.113)	**0.395**	0.129	**0.199**	10602.23	1256.39	55.19	0.018(0.015)	**0.132**
Intern-LM2	IO	**0.317**	**0.439**	**0.164**	**0.242**	4420.49	239.16	8.14	0.007	
	CoT	0.279	0.436	0.158	0.237	4417.75	195.71	7.19	0.007	
	ToT	**0.477**(0.198)	0.414	0.148	0.212	39812.07	2241.35	67.43	0.064(0.057)	0.287
	GoT	0.456(0.177)	0.419	**0.156**	0.220	39732.42	2225.52	67.26	0.064(0.057)	0.322
	DGoT	0.399(0.120)	**0.422**	0.152	**0.222**	19690.67	1016.34	33.78	0.031(0.024)	**0.200**

Note: Experiment conducted on the first 100 testing set data. The input lengths of ChatGLM2 and InterLM2 are 2048 and 4096, respectively.

† Best R-1 score on PubMedCite Inductive dataset is 41.62 (Luo et al., 2023)



Performance of ChatGLM2



Performance of InternLM2

Thanks!

Contact us if you have Questions and advices!

GitHub: https://github.com/JayceNing/DGoT

Xinyu Ning, Yutong Zhao, Yitong Liu*, Hongwen Yang

{nxybupt, zhaoyutong, liuyitong, yanghong}@bupt.edu.cn