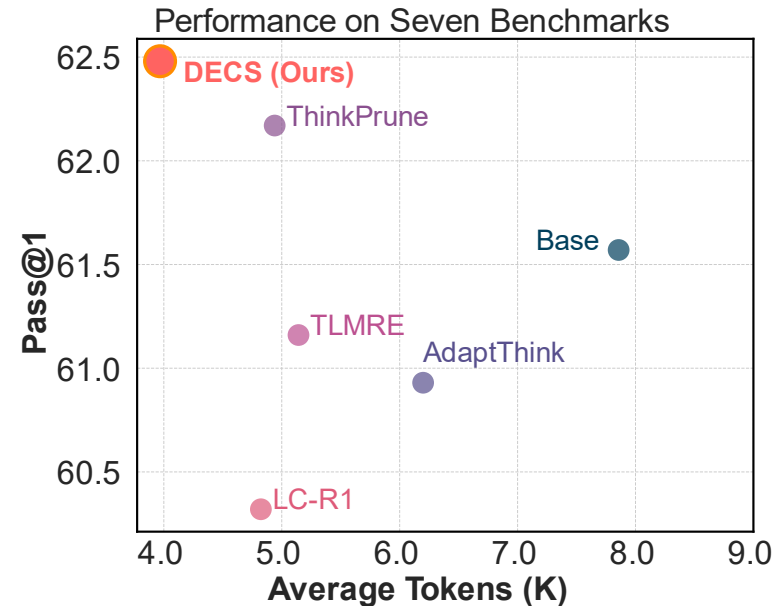


Overthinking Reduction with Decoupled Rewards and Curriculum Scheduling



Shuyang Jiang*, Yusheng Liao, Ya Zhang, Yanfeng Wang, Yu Wang✉

DECS reduces overthinking by over 50% while improving reasoning performance, as a free lunch.



Problem Setting

- ▶ **Overthinking:** generating excessively long reasoning paths without performance benefit.

Define $p = \sum_{k=1}^{\infty} \frac{1}{k^2}$ and $q = \sum_{k=1}^{\infty} \frac{1}{k^3}$. Find a way to write $\sum_{j=1}^{\infty} \sum_{k=1}^{\infty} \frac{1}{(j+k)^3}$

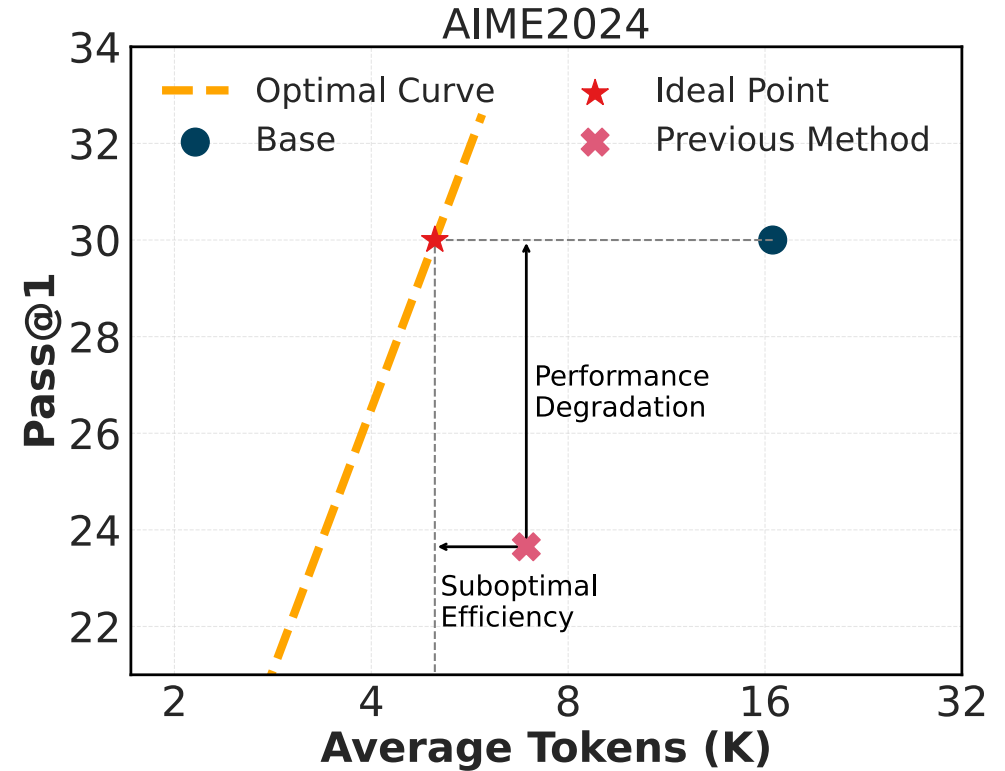
Necessary Thinking: 3283 Tokens

Okay, so I need to find a way to express the double sum $\sum_{j=1}^{\infty} \sum_{k=1}^{\infty} \frac{1}{(j+k)^3}$ in terms of (p) and (q) , where $(p = \sum_{k=1}^{\infty} \frac{1}{k^2})$ and $(q = \sum_{k=1}^{\infty} \frac{1}{k^3})$. Hmm, let me think about how to approach this. First, I know that (p) is the well-known Basel problem, which sums to $(\frac{\pi^2}{6})$, but I don't think I need the exact value here. Similarly, (q) is the sum of reciprocals of cubes, which converges to Apéry's constant, approximately 1.202..., but again, exact value isn't necessary here. The double sum given is over all positive integers (j) and (k) , and the term inside is $(\frac{1}{(j+k)^3})$. So, it's a double series where each term depends on the sum of two variables. This seems a bit tricky because it's a two-dimensional sum, unlike single sums which are straightforward. Maybe I can change variables or find a way to express this double sum in terms of (p) and (q) . Let me consider switching the order of summation or using some symmetry. Another idea: perhaps express $(\frac{1}{(j+k)^3})$ as an integral. Since $(\frac{1}{n^3})$ can be written as an integral, maybe I can use that to interchange the sum and the integral. Recall that for positive integers (n) , we have: $(\frac{1}{n^3} = \frac{1}{2} \int_0^1 \int_0^1 (xy)^{n-1} dx dy)$. Wait, is that correct? Let me check. Actually, [omitted...] So, that gives us $(p - q)$.

Overthinking: 492 Tokens

Hmm, that seems too straightforward. Wait, but is that correct? [omitted...] Which is equal to: $(\sum_{m=1}^{\infty} \frac{1}{m^2} - 1) - (\sum_{m=1}^{\infty} \frac{1}{m^3} - 1) = (p - 1) - (q - 1) = p - q$. So, is that correct? Hmm. Let me verify. [omitted...] which is indeed $(p - q)$. Therefore, the double sum is equal to $(p - q)$. So, the answer is $(\boxed{p - q})$. **Final Answer**: $(\boxed{p - q})$

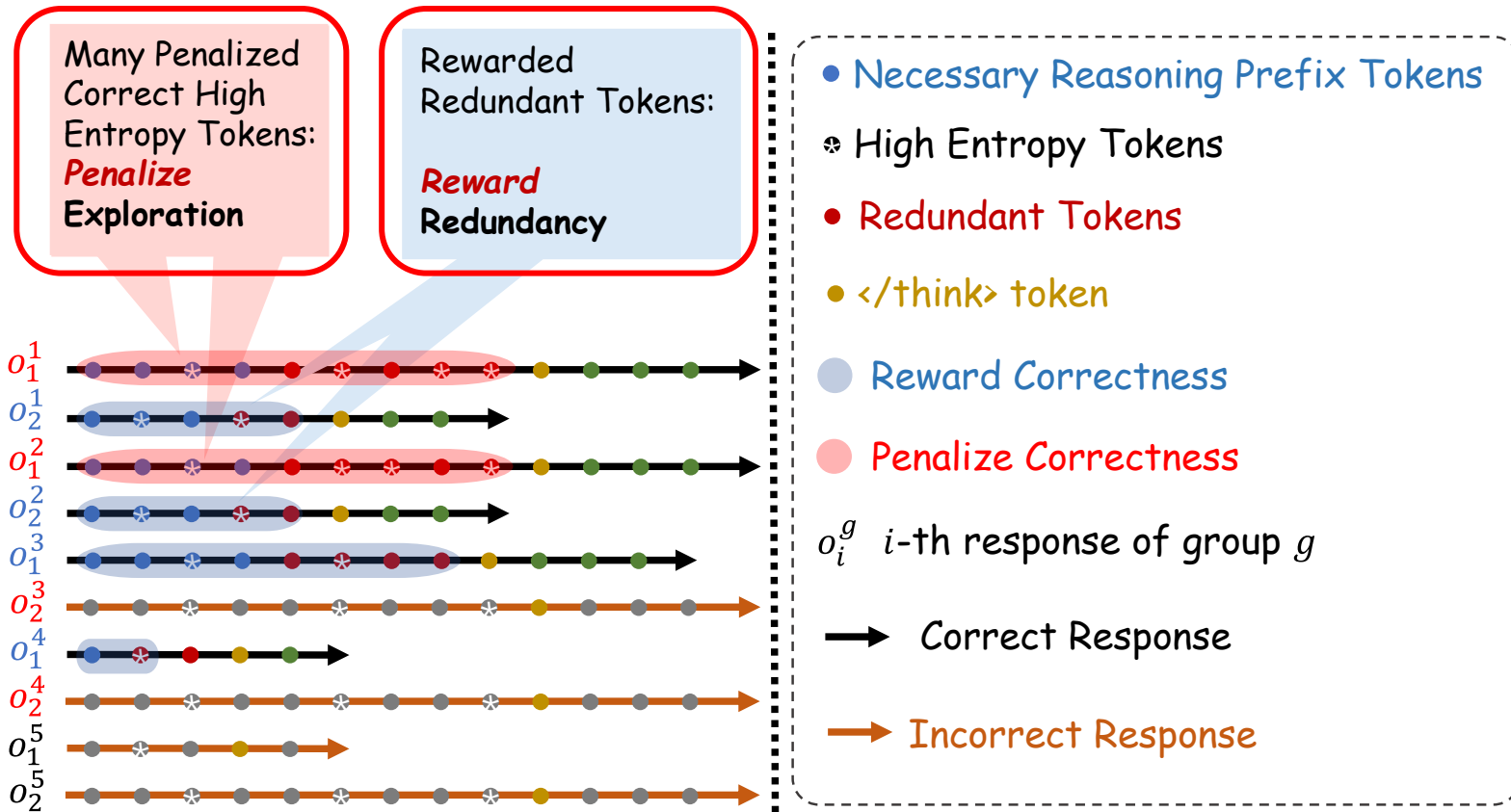
- ▶ **Goal and Challenge**



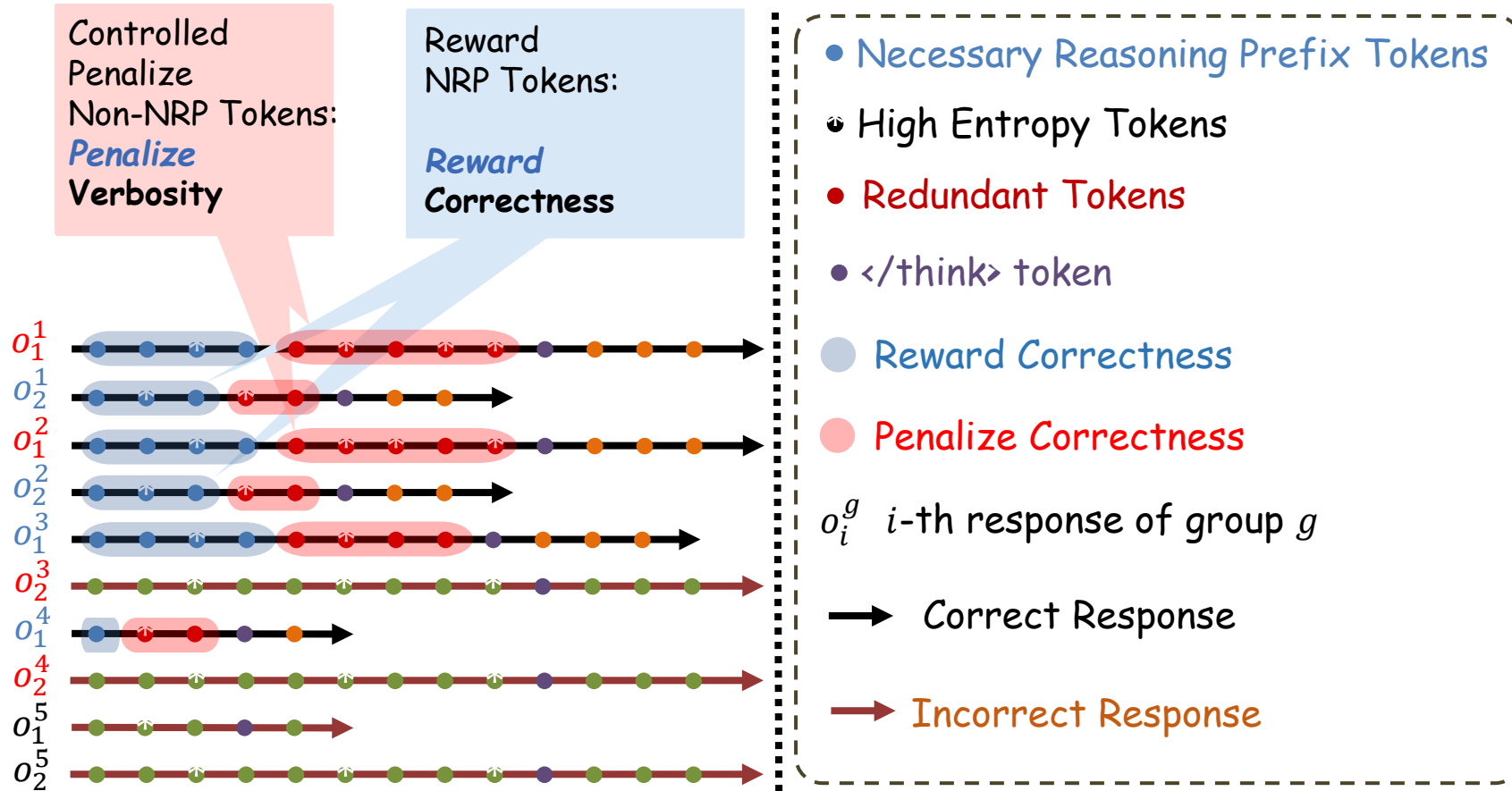
Why not just use a length penalty?

- ▶ Sequence-level length rewards are misaligned with token-level optimization.

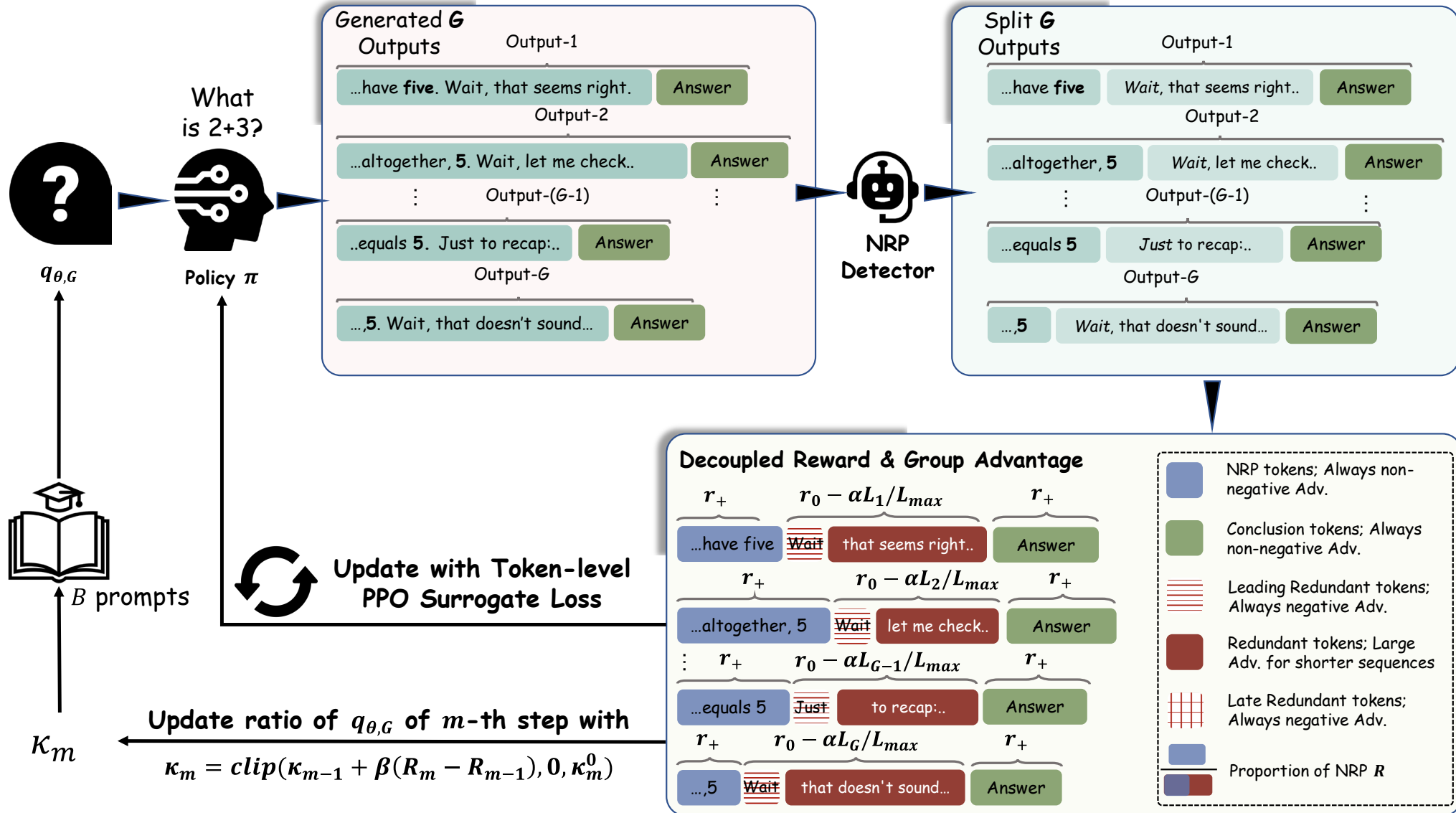
$$r'(o) = \begin{cases} r(o) - \gamma|o| & o \text{ is correct} \\ r(o) & \text{otherwise} \end{cases}$$



Where is the solution?



Method overview — DECS



Necessary Reasoning Prefix

- ▶ The shortest reasoning prefix that first justifies the correct answer.
- ▶ We aim to **reduce overthinking**, instead of **promoting short responses**

Define $p = \sum_{k=1}^{\infty} \frac{1}{k^2}$ and $q = \sum_{k=1}^{\infty} \frac{1}{k^3}$. Find a way to write $\sum_{j=1}^{\infty} \sum_{k=1}^{\infty} \frac{1}{(j+k)^3}$

NRP: 3283 Tokens

Okay, so I need to find a way to express the double sum $\sum_{j=1}^{\infty} \sum_{k=1}^{\infty} \frac{1}{(j+k)^3}$ in terms of (p) and (q) , where $(p = \sum_{k=1}^{\infty} \frac{1}{k^2})$ and $(q = \sum_{k=1}^{\infty} \frac{1}{k^3})$. Hmm, let me think about how to approach this.

First, I know that (p) is the well-known Basel problem, which sums to $(\frac{\pi^2}{6})$, but I don't think I need the exact value here. Similarly, (q) is the sum of reciprocals of cubes, which converges to Apéry's constant, approximately 1.202..., but again, exact value isn't necessary here.

The double sum given is over all positive integers (j) and (k) , and the term inside is $(\frac{1}{(j+k)^3})$. So, it's a double series where each term depends on the sum of two variables. This seems a bit tricky because it's a two-dimensional sum, unlike single sums which are straightforward.

Maybe I can change variables or find a way to express this double sum in terms of (p) and (q) . Let me consider switching the order of summation or using some symmetry.

Another idea: perhaps express $(1/(j+k)^3)$ as an integral. Since $(1/n^3)$ can be written as an integral, maybe I can use that to interchange the sum and the integral.

Recall that for positive integers (n) , we have: $(\frac{1}{n^3}) = \frac{1}{2} \int_0^1 \int_0^1 (xy)^{n-1} dx dy$

Wait, is that correct? Let me check. Actually, [omitted...]

So, that gives us $(p - q)$.

Overthinking: 492 Tokens

Hmm, that seems too straightforward.

Wait, but is that correct? [omitted...]

Which is equal to: $(\sum_{m=1}^{\infty} \frac{1}{m^2} - 1) - (\sum_{m=1}^{\infty} \frac{1}{m^3} - 1) = (p - 1) - (q - 1) = p - q$

So, is that correct? Hmm. Let me verify. [omitted...]

which is indeed $(p - q)$.

Therefore, the double sum is equal to $(p - q)$.

So, the answer is $(p - q)$.

Final Answer $(p - q)$

Method detail 1 — NRP Detector

- ▶ Goal: Lightweight, Precise, and Implementation-friendly
- ▶ Solution:
 - ▶ Split reasoning trajectories into reasoning chunks
 - ▶ Verdict whether a reasoning chunk contain the ground truth.
 - ▶ Collect chunks from start to the chunk that firstly contain the ground truth.

Method detail 1 — NRP Detector Pretraining

1. Preparing training data

- ▶ Curated 5000 cases from OpenR1-Math-220K
- ▶ Splitting reasoning trajectories into chunks: Wait, But, Alternatively, Hmm, However, Maybe

Necessary Thinking: 3283 Tokens

Okay, so I need to find a way to express the double sum $\sum_{j=1}^{\infty} \sum_{k=1}^{\infty} \frac{1}{(j+k)^3}$ in terms of (p) and (q) , where $(p) = \sum_{k=1}^{\infty} \frac{1}{k^2}$ and $(q) = \sum_{k=1}^{\infty} \frac{1}{k^3}$. Hmm, let me think about how to approach this.

First, I know that (p) is the well-known Basel problem, which sums to $\frac{\pi^2}{6}$, but I don't think I need the exact value here. Similarly, (q) is the sum of reciprocals of cubes, which converges to Apéry's constant, approximately 1.202..., but again, exact value isn't necessary here.

The double sum given is over all positive integers (j) and (k) , and the term inside is $\frac{1}{(j+k)^3}$. So, it's a double series where each term depends on the sum of two variables. This seems a bit tricky because it's a two-dimensional sum, unlike single sums which are straightforward.

Maybe I can change variables or find a way to express this double sum in terms of (p) and (q) . Let me consider switching the order of summation or using some symmetry.

Another idea: perhaps express $\frac{1}{(j+k)^3}$ as an integral. Since $\frac{1}{n^3}$ can be written as an integral, maybe I can use that to interchange the sum and the integral.

Recall that for positive integers (n) , we have: $\frac{1}{n^3} = \frac{1}{2} \int_0^1 \int_0^1 (xy)^{n-1} dx dy$

Wait, is that correct? Let me check. Actually, [omitted...] So, that gives us $(p - q)$.

CNK1: Okay, so I need to find a way to express the double sum $\sum_{j=1}^{\infty} \sum_{k=1}^{\infty} \frac{1}{(j+k)^3}$ in terms of (p) and (q) , where $(p) = \sum_{k=1}^{\infty} \frac{1}{k^2}$ and $(q) = \sum_{k=1}^{\infty} \frac{1}{k^3}$.

CNK2: *Hmm*, let me think about how to approach this.

First, I know that (p) is the well-known Basel problem, which sums to $\frac{\pi^2}{6}$, but I don't think I need the exact value here. Similarly, (q) is the sum of reciprocals of cubes, which converges to Apéry's constant, approximately 1.202...,

CNK3: *but* again, exact value isn't necessary here.

The double sum given is over all positive integers (j) and (k) , and the term inside is $\frac{1}{(j+k)^3}$. So, it's a double series where each term depends on the sum of two variables. This seems a bit tricky because it's a two-dimensional sum, unlike single sums which are straightforward.

CNK4: *Maybe* I can change variables or find a way to express this double sum in terms of (p) and (q) . Let me consider switching the order of summation or using some symmetry.

Another idea: perhaps express $\frac{1}{(j+k)^3}$ as an integral. Since $\frac{1}{n^3}$ can be written as an integral, maybe I can use that to interchange the sum and the integral.

Recall that for positive integers (n) , we have: $\frac{1}{n^3} = \frac{1}{2} \int_0^1 \int_0^1 (xy)^{n-1} dx dy$

CNK5: *Wait*, is that correct? Let me check. Actually, [omitted...] So, that gives us $(p - q)$.

Method detail 1 — NRP Detector Pretraining

2. Query a Large model to obtain ground truth
 - ▶ Human checked to ensure no false-negative or false-positive
 - ▶ Distill the answer to a small model: Qwen2.5-1.5B-Instruct

NRP Detector Prompt

```
Given a math problem and a segment of a long reasoning process to solve the problem, your task is to identify whether this segment has presented a correct final answer. If the segment contains information that can serve as the final answer to the problem and the answer is semantically correct when referring to the ground truth, simply explain the reason and output \boxed{yes}. Otherwise, directly output \boxed{no}.
```

```
**Problem**:  
{problem}  
  
**Reasoning segment**:  
{segment}  
  
**Ground Truth**:  
{answer}
```

Method detail 1 — NRP Detector Pretraining

3. Fetching NRP tokens

- ▶ Tokens up to the first chunk that is labeled as True

CNK-1: Okay, so I need to find a way to express the double sum $\sum_{j=1}^{\infty} \sum_{k=1}^{\infty} \frac{1}{(j+k)^3}$ in terms of (p) and (q) , where $(p) = \sum_{k=1}^{\infty} \frac{1}{k^2}$ and $(q) = \sum_{k=1}^{\infty} \frac{1}{k^3}$.

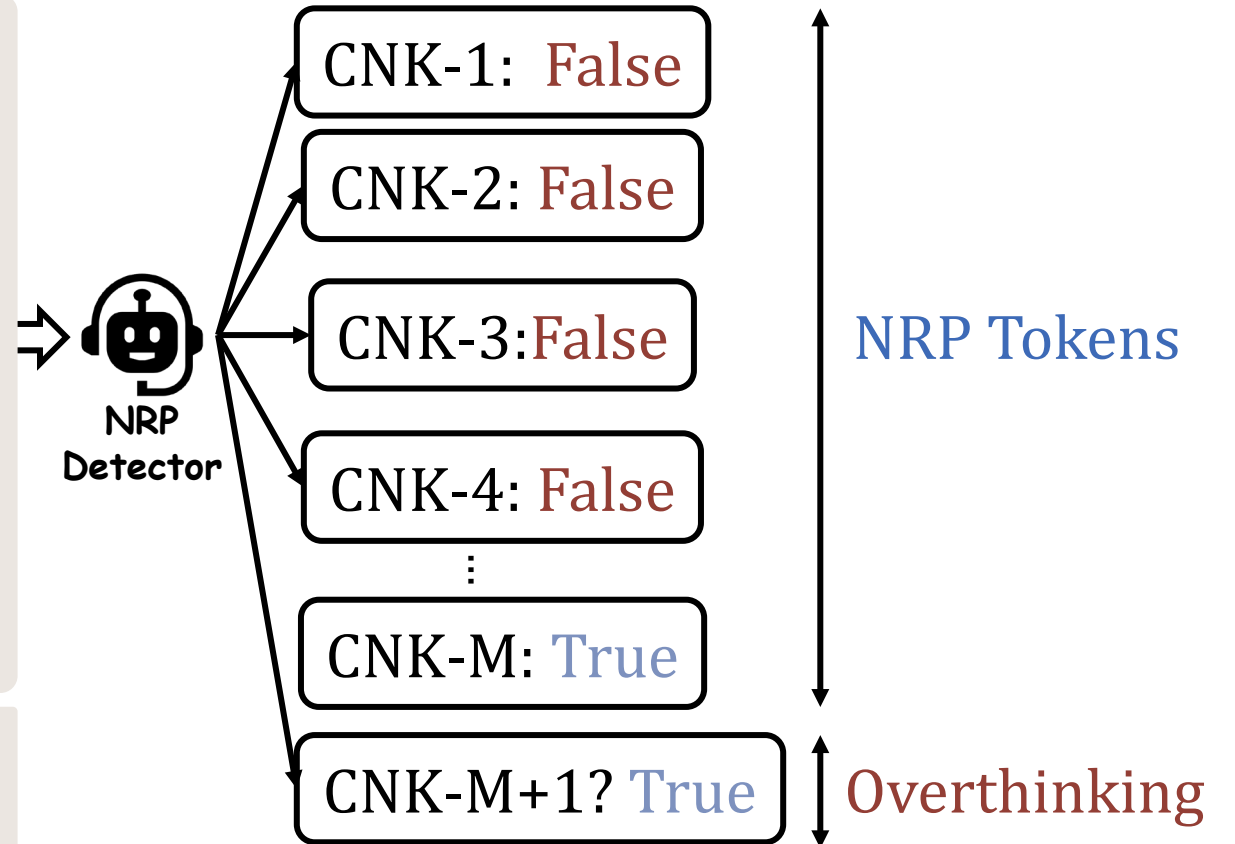
CNK-2: Hmm, let me think about how to approach this. First, I know that (p) is the well-known Basel problem, which sums to $\frac{\pi^2}{6}$, but I don't think I need the exact value here. Similarly, (q) is the sum of reciprocals of cubes, which converges to Apéry's constant, approximately 1.202..., but again, exact value isn't necessary here. The double sum given is over all positive integers (j) and (k) , and the term inside is $\frac{1}{(j+k)^3}$. So, it's a double series where each term depends on the sum of two variables. This seems a bit tricky because it's a two-dimensional sum, unlike single sums which are straightforward. Maybe I can change variables or find a way to express this double sum in terms of (p) and (q) .

CNK-3: Let me consider switching the order of summation or using some symmetry. Another idea: perhaps express $\frac{1}{(j+k)^3}$ as an integral. Since $\frac{1}{n^3}$ can be written as an integral, maybe I can use that to interchange the sum and the integral. Recall that for positive integers (n) , we have: $\frac{1}{n^3} = \frac{1}{2} \int_0^1 \int_0^1 (xy)^{n-1} dx dy$

CNK-4: Wait, is that correct? Let me check. Actually, [omitted...]

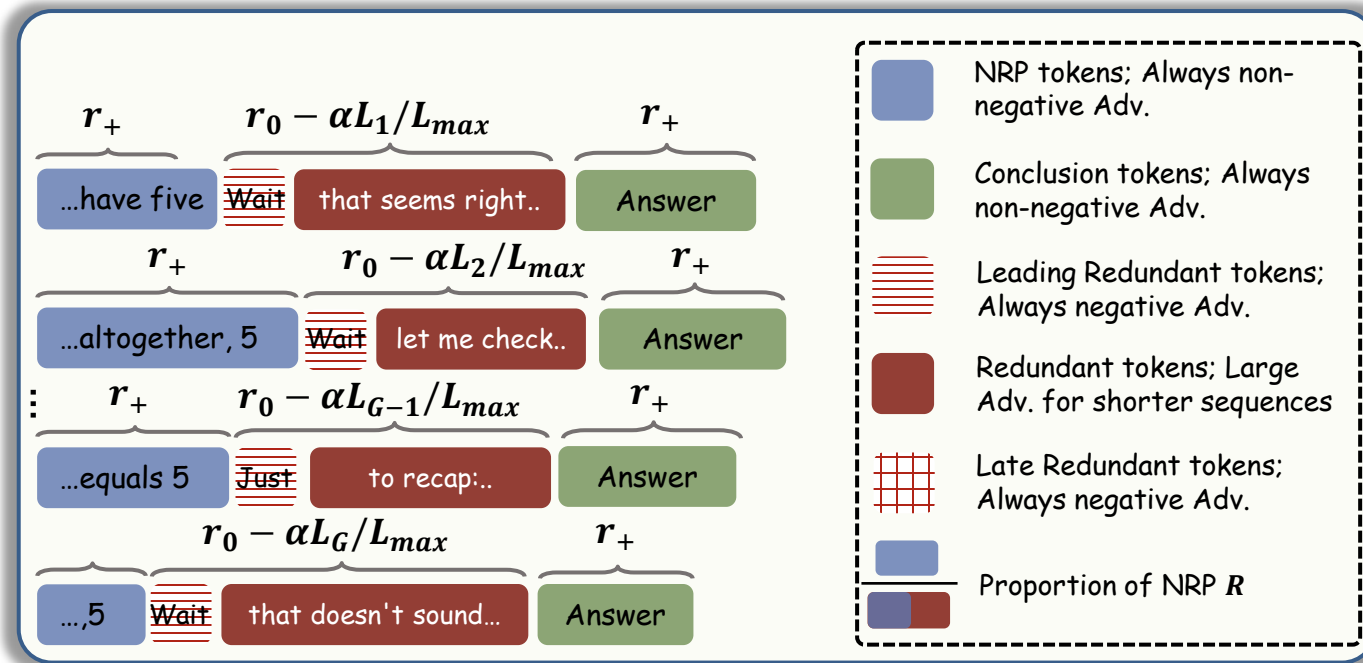
CNK-M: So, that gives us $(p - q)$.

CNK-M+1: Hmm, that seems too straightforward. Wait, but is that correct? [omitted...] Which is equal to: $\frac{1}{2} \left(\sum_{m=1}^{\infty} \frac{1}{m^2} - 1 \right) - \left(\sum_{m=1}^{\infty} \frac{1}{m^3} - 1 \right) = (p - 1) - (q - 1) = p - q$. So, is that correct? Hmm. Let me verify. [omitted...] which is indeed $(p - q)$. Therefore, the double sum is equal to $(p - q)$. So, the answer is $(p - q)$. **Final Answer** $(p - q)$



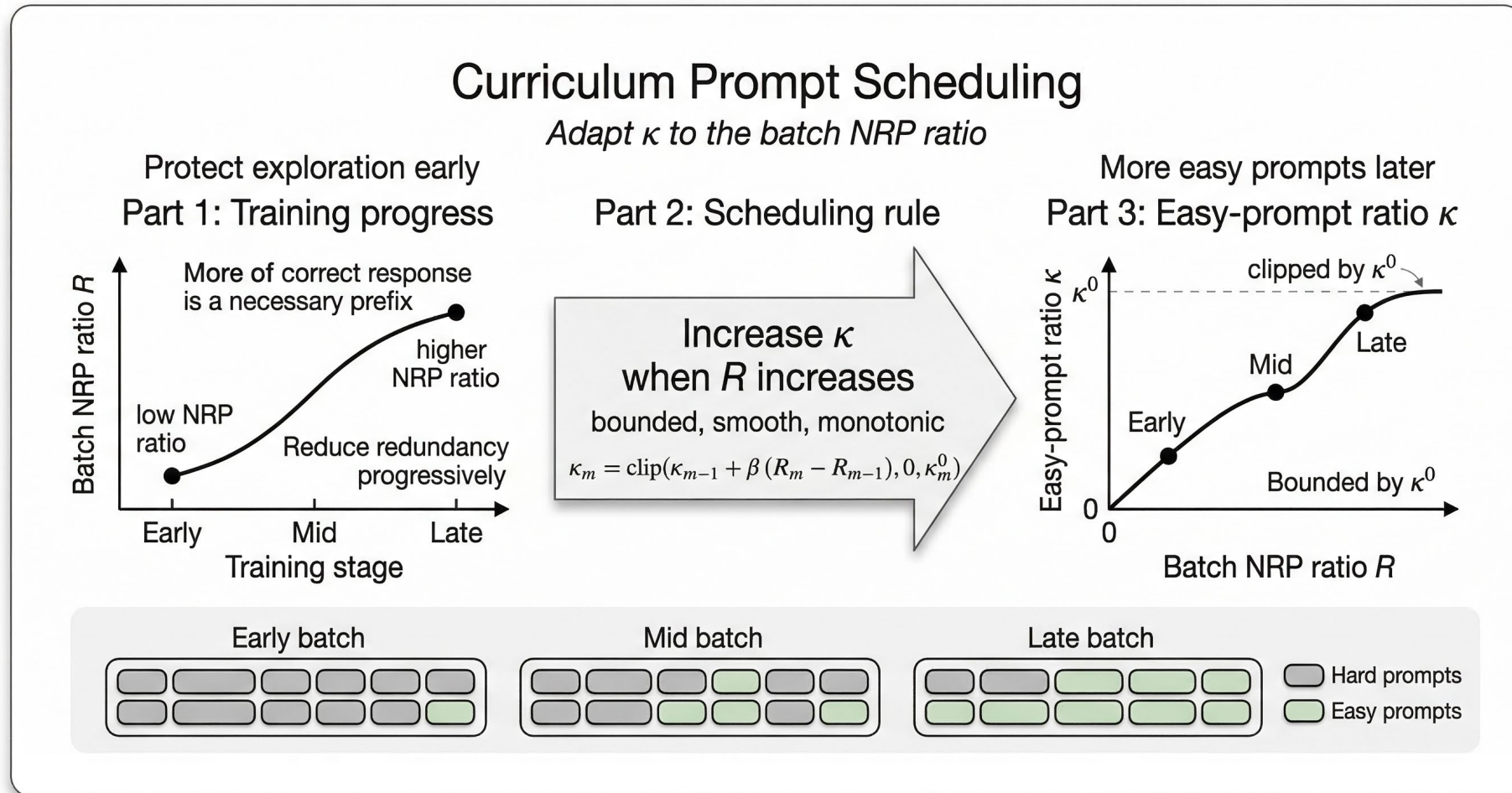
Method detail 2 — Decoupled reward

- ▶ NRP Tokens & Answer Tokens: Largest positive reward for non-negative Advantage
 - ▶ For protection of correct non-verbose reasoning logics: $r_{NRP} = 1.1$
- ▶ Overthinking Tokens: The simple linear length penalty
 - ▶ For always penalizing redundancy: $r_{overthinking} = 1 - \frac{\alpha L_1}{L_{max}}$



Method detail 3 — Curriculum Scheduling

Schedule easy prompts according to current redundancy level.

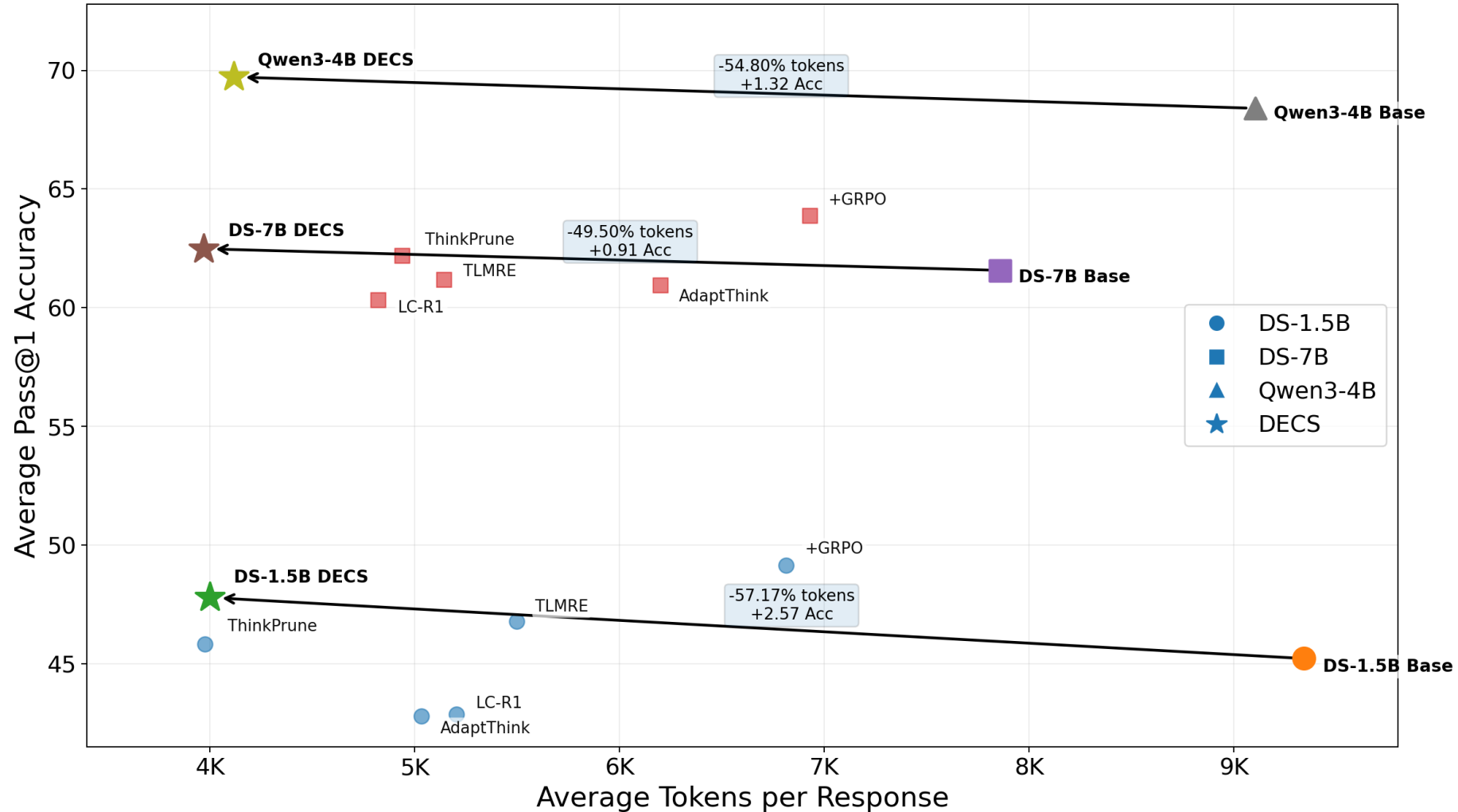


Experiments

- ▶ Backbone Language models
 - ▶ DeepSeek-R1-Distilled-1.5B, 7B, Qwen3-4B
- ▶ Fine-tuning corpus
 - ▶ DeepScaleR
- ▶ Benchmark datasets
 - ▶ In-domain dataset: AIME24, AIME25, AMC23, MATH500, OlympiadBench
 - ▶ Out-of-domain dataset: GPQA-D, LiveCodeBench-v6
- ▶ Baselines:
 - ▶ ThinkPrune, TLMRE, LC-R1, AdaptThink

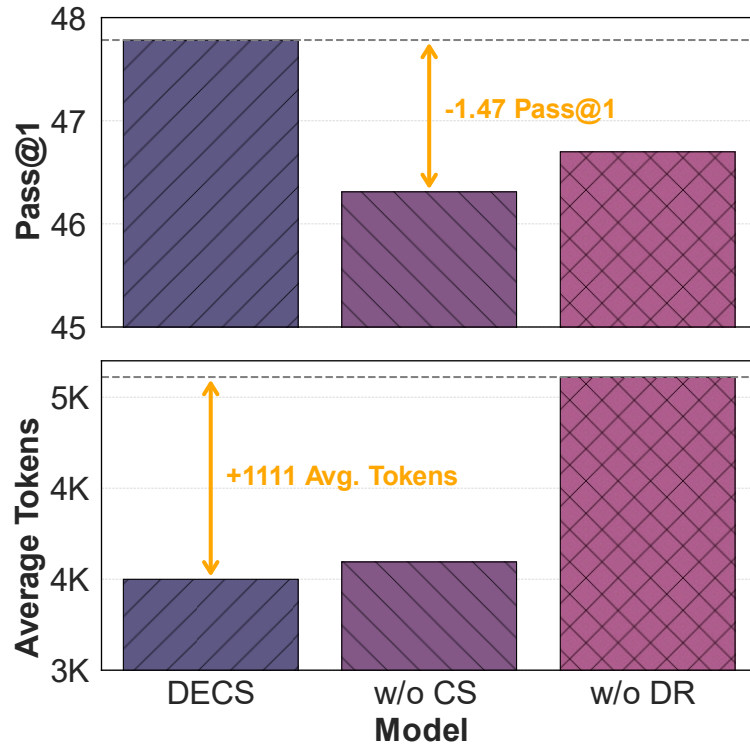
Experiments- DECS is optimized on a better frontier

Main Results Across 7 Benchmarks

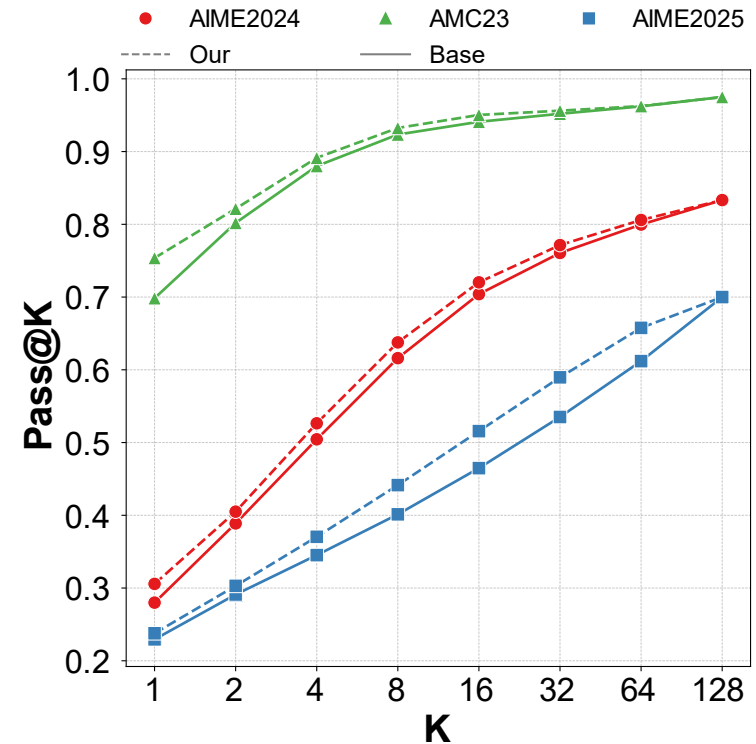


DECS reduces redundancy without hurting exploration

- ▶ Ablation: CS for correctness, DR for efficiency



- ▶ Exploration: preserving pass@K across different Ks.



Conclusion

- ▶ Efficient reasoning should penalize redundancy, not reasoning itself.
- ▶ DECS separates necessary reasoning from redundant reasoning.
- ▶ It reduces overthinking by about 50% without sacrificing capability.



Thank you!

To whom is interested in DECS:

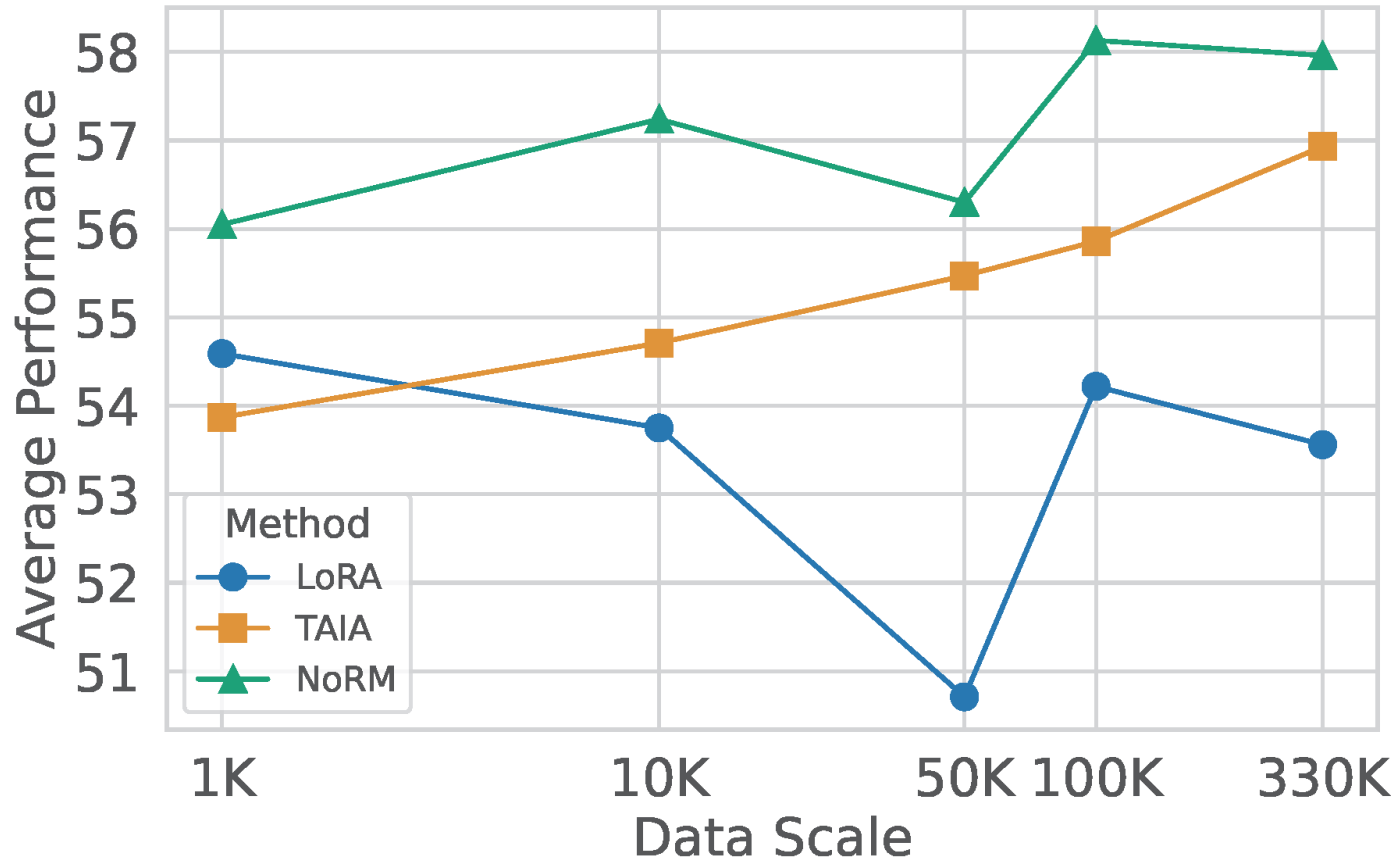
Code Available: <https://pixas.github.io/decs-iclr26-site/>

Corresponding to: <https://yuwangsjtu.github.io/>

Contact me: <https://pixas.github.io/>

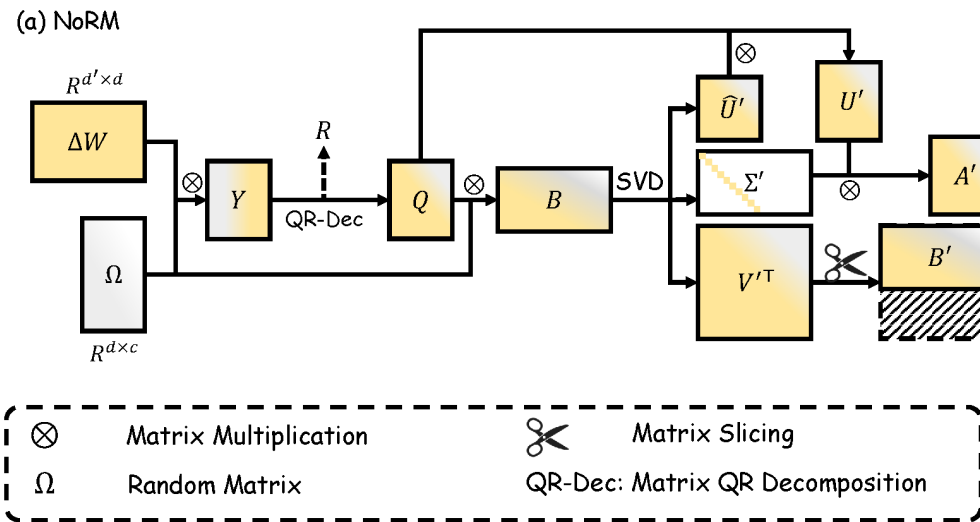
Strength-Higher data sensitivity

- ▶ Performance with increasing fine-tuning corpus ([1K, 10K, 50K, 100K, 330K])



- ▶ *High data robustness and generalization*

NoRM



- ▶ Selecting major $c < r$ largest singular values and corresponding singular vectors:

$$\Sigma' = \text{diag}(\sigma_1, \sigma_2, \dots, \sigma_c), U' = U[:, 1:c], V' = V[:, 1:c]$$
- ▶ The re-constructed delta weights are formed by:

$$\Delta W = BA \approx U'\Sigma'V'^T$$
- ▶ **Pre-processing of ΔW is computationally heavy**

NoRM

- ▶ Instead, we use random-SVD (Haiko et al., 2011) to speed up and approximate B and A
- ▶ Random SVD gives $U', \Sigma', V' = \text{Random - SVD}(\Delta W)$

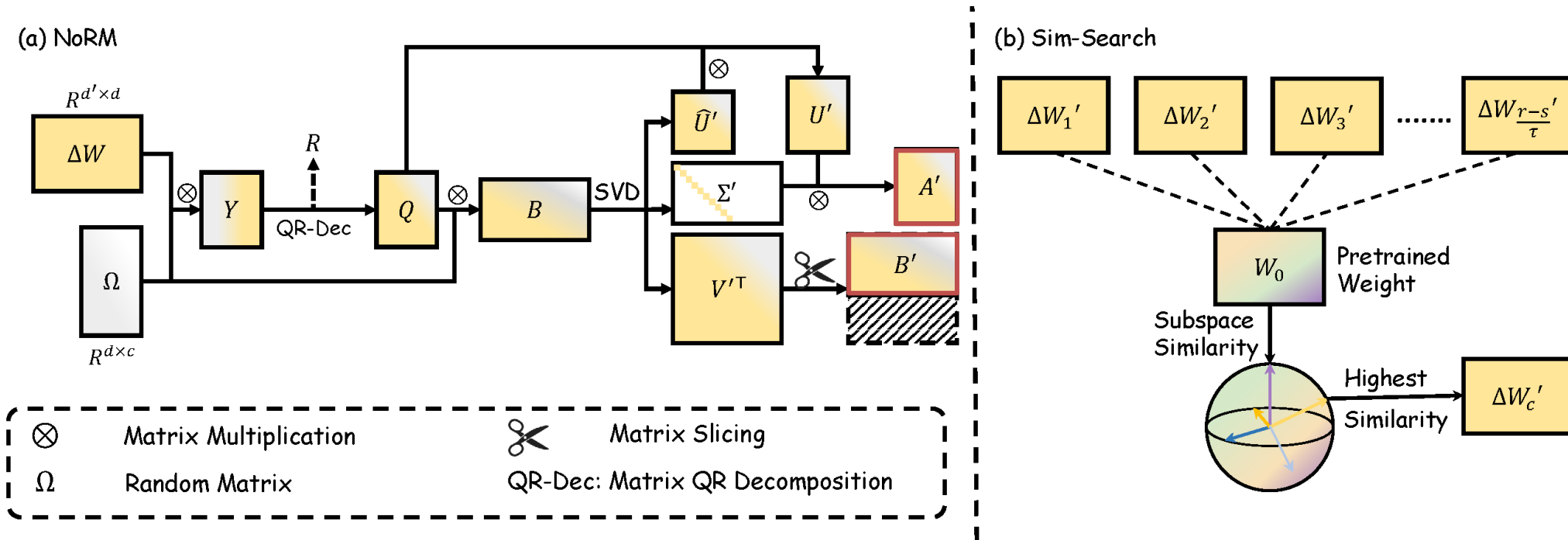
NoRM

- ▶ Instead, we use random-SVD (Haiko et al., 2011) to speed up and approximate B and A
- ▶ Random SVD is computed as follows:
 1. Initialize a Gaussian random matrix: $\Omega \sim \mathcal{N}(0, I) \in R^{d \times c}$
 2. Compute column major sub-space of ΔW : $Y = \Delta W \Omega$
 3. Compute the approximation of orthogonal basis of ΔW : $Y = QR$
 4. Obtain the projection of ΔW in Q 's low-dimensional space: $B = Q^\top \Delta W$
 5. Conduct normal SVD on B : $B = \hat{U}' \Sigma' V'^\top$
 6. The singular vectors are approximated by: $U' = Q \hat{U}'$

NoRM

- ▶ Obtaining the quantities above, we reconstruct A' and B' as such:

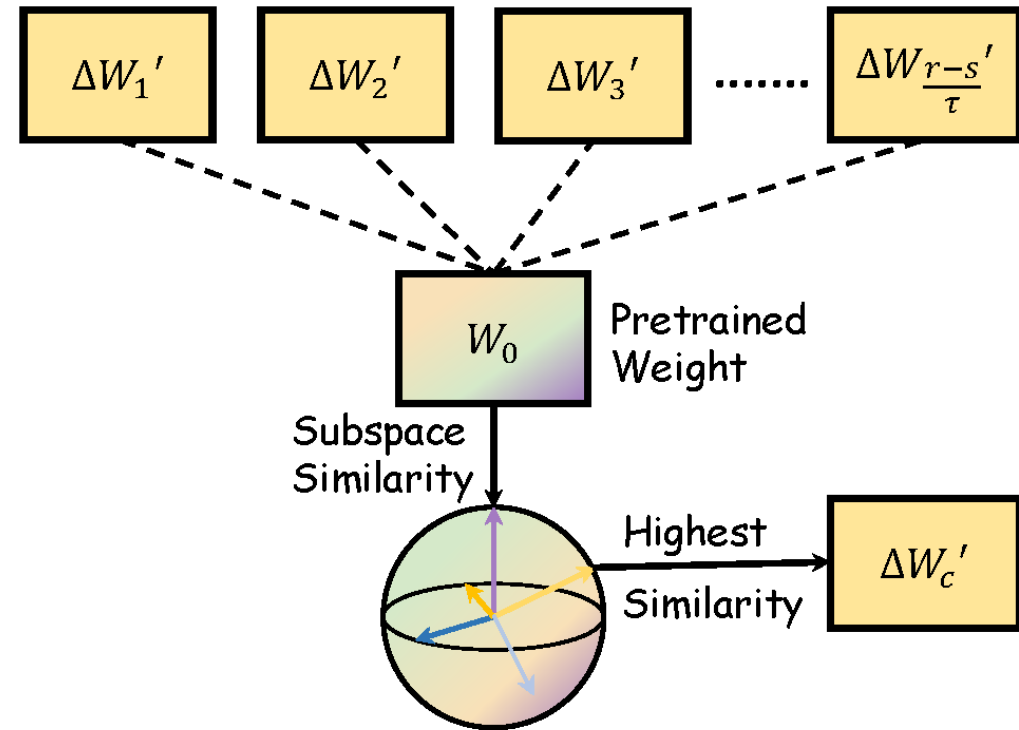
$$B' = U' \cdot \text{diag}(\Sigma') \quad A' = V'^T [1:c, :]$$



Sim-Search

- ▶ How to determine c ?
- ▶ Hu et al., 2022; Wang et al., 2024 empirically demonstrate that the subspace similarity between the delta weight ΔW and the pretrained weight W correlates positively to downstream performance

(b) Sim-Search



Philosophy of Sim-Search

- ▶ Perform grid-search to select the most suitable c values which results in the highest sub-space similarity between delta weights and pre-trained weights

Methodology of Sim-Search

- ▶ Perform grid-search to select the most suitable c values which results in the highest sub-space similarity between delta weights and pre-trained weights
- ▶ Introduce a search step τ and search the c values ranging from a given start value s to r : $\{\tau \cdot s, (\tau + 1) \cdot s, \dots, r\}$, where s is the step size.
- ▶ Perform random-SVD to obtain corresponding A'_c and B'_c for varying c

Methodology of Sim-Search

- ▶ Perform grid-search to select the most suitable c values which results in the highest sub-space similarity between delta weights and pre-trained weights
- ▶ Introduce a search step τ and search the $c = r \cdot \beta$ values ranging from a given start value s to r : $\{\tau \cdot s, (\tau + 1) \cdot s, \dots, r\}$
- ▶ Perform random-SVD to obtain corresponding A'_c and B'_c for varying c
- ▶ Compute the singular vectors of re-constructed delta weights (compute before):

$$U_c, \Sigma_c, V_c^\top = \text{SVD}(B'_c A'_c)$$

- ▶ Obtain the pre-trained singular vectors same as above:

$$U, \Sigma, V^\top = \text{SVD}(W)$$

Methodology of Sim-Search

- ▶ Perform grid-search to select the most suitable c values which results in the highest sub-space similarity between delta weights and pre-trained weights
- ▶ Creating a c list and corresponding B'_c, A'_c and U_c, Σ_c, V_c^T
- ▶ Obtain the pre-trained singular vectors same as above:

$$U, \Sigma, V^T = \text{SVD}(W)$$

Methodology of Sim-Search

- ▶ Obtain the pre-trained singular vectors same as above:

$$U, \Sigma, V^T = \text{SVD}(W)$$

- ▶ Extract the r left singular vectors of U_c and U and compute the subspace similarity (measured by Grassmann distance):

$$\phi_c = \frac{\|U_{cr}^T \cdot U_r\|_F^2}{r}$$

- ▶ Selecting the reduced delta weights B'_c, A'_c by the largest subspace similarity:

$$\{c, B'_c, A'_c\} = \arg \max_c \phi_c$$

Experiments-Surpass PEFT baselines by significant margin

Model	Method	BBH	MMLU	TydiQA	CQA	TruthfulQA	GSM8K	Logiqa-EN	Average
Qwen2-7B	Base	41.73	66.84	45.36	74.20	57.65	87.19	40.40	59.05
	LoRA	44.31	67.95	48.45	75.84	50.80	83.02	43.93	59.19
	LoRA+	42.64	67.81	52.15	78.71	50.18	83.24	43.78	59.79
	DoRA	43.11	67.94	44.64	75.43	53.98	82.94	44.85	58.99
	MoRA	36.31	62.43	46.70	72.65	53.37	75.44	44.09	55.85
	TAIA	44.79	66.36	46.81	75.10	58.51	85.60	45.78	60.42
	NoRM	45.19	68.89	50.80	78.05	57.77	85.06	46.08	61.69
Llama3-8B	Base	40.68	59.18	40.90	71.83	63.28	77.79	41.17	56.40
	LoRA	36.38	63.42	43.56	77.64	48.10	72.02	38.40	54.22
	LoRA+	40.30	63.30	38.01	75.35	37.70	73.62	36.71	52.14
	DoRA	36.95	63.89	42.54	77.97	42.59	71.27	37.94	53.31
	MoRA	–	25.09	–	20.48	20.07	–	26.42	23.01
	TAIA	35.83	62.02	47.26	76.66	54.22	77.10	37.94	55.86
	NoRM	43.11	64.61	46.21	77.72	54.10	77.71	43.47	58.13
Mistral-7B	Base	39.26	54.07	30.04	66.83	56.30	55.27	36.87	48.38
	LoRA	34.79	53.68	41.88	73.05	53.37	41.47	33.79	47.43
	LoRA+	32.51	56.67	39.99	70.52	42.23	45.49	38.40	46.54
	DoRA	34.66	53.31	32.01	68.88	38.19	42.99	36.25	43.76
	MoRA	26.69	27.58	27.40	49.14	33.66	20.24	33.18	31.13
	TAIA	38.06	54.97	39.20	71.91	53.24	50.87	38.25	49.50
	NoRM	39.29	58.05	40.56	73.55	54.47	52.01	37.63	50.79

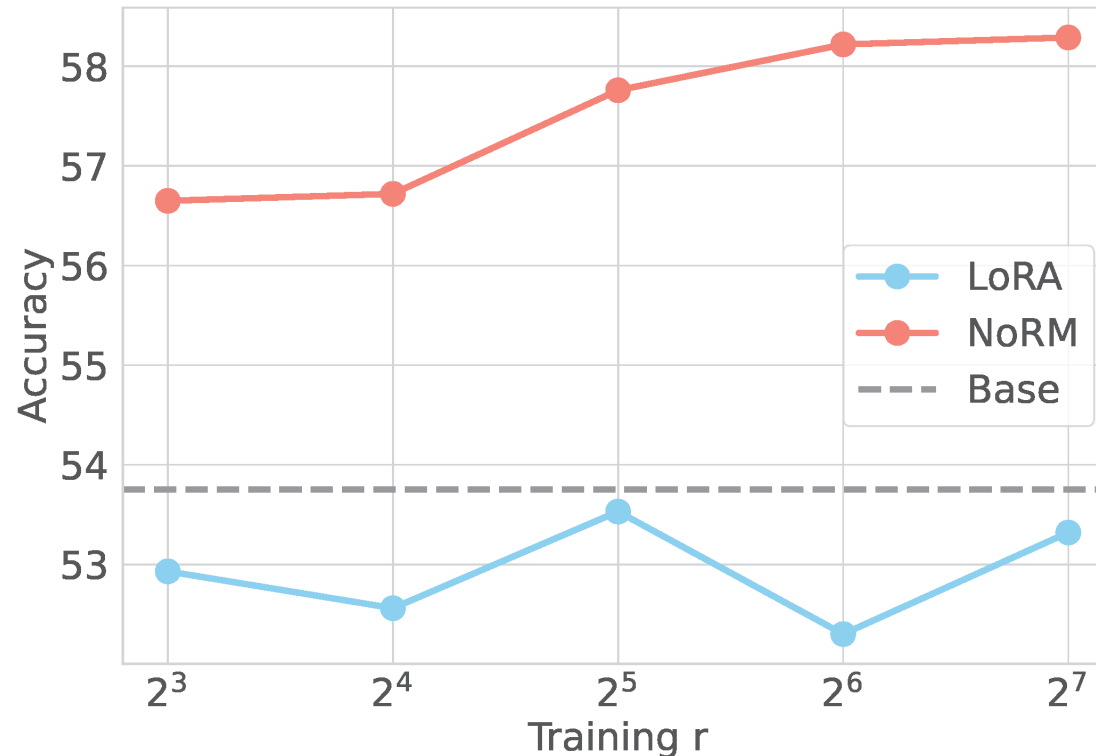
Experiments-Surpass PEFT baselines by significant margin

Model	Math			Code			Avg.
	GSM8k	MATH	HumanEval	HumanEval+	MBPP	MBPP+	
Pretrained	77.79	31.06	53.00	48.80	71.70	60.60	57.16
<i>PEFT Method</i>							
Full	77.86	27.86	58.50	53.70	65.30	55.00	56.37
LoRA (Hu et al., 2022)	80.21	29.27	48.80	44.50	67.20	57.40	54.56
LoRA+ (Hayou et al., 2024)	78.77	28.22	56.70	52.40	68.50	58.50	57.18
DoRA (yang Liu et al., 2024)	80.82	29.84	51.80	46.30	66.90	57.40	55.51
MoRA (Jiang et al., 2024b)	63.15	19.48	43.30	39.00	47.90	39.20	42.01
<i>PREFT Method</i>							
TAIA	79.08	32.60	59.10	53.00	69.30	60.60	58.95
NoRM	82.79	33.76	63.40	59.80	71.40	60.80	61.99

Strength-Scaling LoRA parameters

- ▶ Performance with increasing LoRA ranks

- ▶ rank sequence: [8, 16, 32, 64, 128]



- ▶ ***NORM can intelligently remove noise distraction and largely benefit from the fine-tuning corpus***

Strength-Learn more

- ▶ AF: Amplification factor of new knowledge
- ▶ RAF: Reverse AF (Amplification factor of known knowledge)

	LoRA				NoRM			
	ΔW_q	W_q	AF \uparrow	RAF \downarrow	ΔW_q	W_q	AF \uparrow	RAF \downarrow
$\ U^\top W_q V\ =$	1.16	34.26	1.88	0.03	0.33	21.99	1.83	0.008
$\ W_q\ = 74.0$	$\ \Delta W_q\ = 2.1875$				$\ \Delta W_q\ = 0.6133$			

Table 5: The Frobenius norm of $U^\top W_q V$ where U and V are the left/right top r singular vector directions of either ΔW_q and W_q . The weight matrices are taken from the 16th layer of Llama3-8B. “AF” and “RAF” indicate the amplification factor and reverse amplification factor, respectively.

- ▶ ***Significant lower overparameterization over learned knowledge***

Strength- Forget less

- ▶ How much NoRM learns from not-amplified features

Method	Training Data		
	General	Coding	Math
Base		3.7016	
LoRA	3.7102	3.7512	3.7323
NoRM	3.5992	3.6450	3.6519

Table 4: Forgetting loss on WikiText-103 test dataset. NoRM reduces hallucination and hence gains superior memorization of internal knowledge over other baselines.

- ▶ *Learn more on already learned data*

Strength-Negligible catastrophic forgetting

- ▶ Learns only beneficial information from fine-tuning corpus
 - ▶ Continual learning with minimal catastrophic forgetting
 - ▶ Standard CL benchmarks:

Model	Order1	Order2	Order3	Average
LoRA	65.34	74.56	70.44	70.11
NoRM	78.88	80.08	78.76	79.24

References

- ▶ Halko N, Martinsson P G, Tropp J A. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions[J]. SIAM review, 2011, 53(2): 217-288.
- ▶ Hu E J, Shen Y, Wallis P, et al. Lora: Low-rank adaptation of large language models[J]. ICLR, 2022, 1(2): 3.
- ▶ Wang H, Li Y, Wang S, et al. Milora: Harnessing minor singular components for parameter-efficient llm finetuning[J]. arXiv preprint arXiv:2406.09044, 2024.