

---

# NLP-DL Assignment 3 Report

---

Xiangyu Wang  
Yuanpei College  
Peking University  
2300017816@stu.pku.edu.cn

## 1 Large Language Model (LLM) inference acceleration

In this section, I focus on LLM inference acceleration related topics. More specifically, KV-cache and quantization. In Sec. 1.1, I test the inference speed and the memory occupation of LLM under different settings. In Sec. 1.2, I implement a custom KV-cache mechanism on my own and compare the performance of my own implementation with golden KV-cache mechanism provided by *transformers* package.

### 1.1 Comparing LLM inference efficiency across different methods

In this experiment, I leverage *facebook/opt-125* (Zhang et al. [5]) as the base language model. I concentrate on four decoding settings: decoding without cache, decoding with KV-cache, decoding with various degrees of quantization, and decoding with both quantization and KV-cache. For each setting, average (output) throughput and peak value of GPU memory occupation of the model are measured over a range of new tokens lengths. Note that all the experiments in this subsection and Sec. 1.2 are conducted on a single NVIDIA A40 GPU.

#### 1.1.1 Experimental results

The main experimental results are shown in Fig. 1 and Fig. 2.

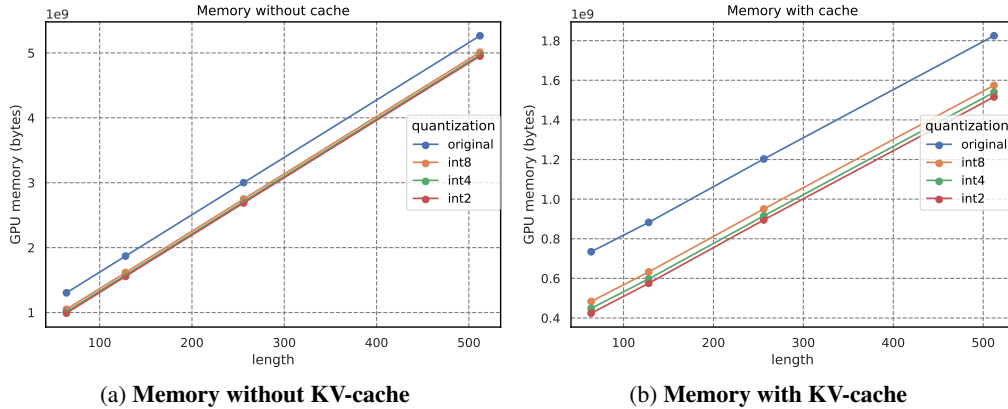


Figure 1: Plots for the GPU memory usage of the model under different settings.

#### 1.1.2 Discussion

Firstly, from Fig. 1, it is evident that no matter whether to use KV-cache or not, leveraging higher degrees of quantization contributes to lower peak GPU memory usage. It proves that utilizing quantization methods to map model parameters or activations to lower precision data can effectively save GPU memory as for LLM.

Secondly, we can detect from Fig. 2 that no matter whether to conduct quantization methods on LLM or not, simply introducing KV-cache mechanism can significantly speed up inference in comparison

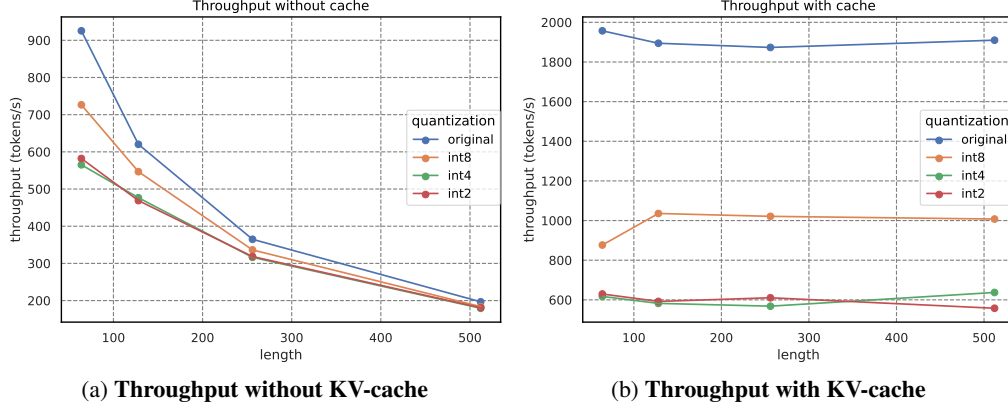


Figure 2: Plots for the throughput of the model under different settings.

to decoding without caching. It's also obvious that this speeding up brought by KV-cache takes more effect when generation length gets longer. In essence, the plots shows that when generation length grows, the inference efficiency of KV-cache settings retains almost the same while that of no KV-cache settings declining rapidly.

Thirdly, comparing the throughput of raw model and quantized model, we can figure that in spite of lower numeric precision, quantized model exhibits worse inference efficiency. My hasty hypothesis is that the calculation process for original precision has been optimized to a large extent, therefore lower the numeric precision cannot save calculation time but introduce extra computational cost for maintaining and utilizing calibration information during inference.

Finally, as shown in Fig. 1, a counter-intuitive phenomenon is that when other configures are identical, decoding with KV-cache inversely occupies less GPU memory. Merely based on the experimental results in this section cannot rule out the possibility that the golden implementation of KV-cache in *transformers* package are specially optimized. Hence I will engage in further discourse about this phenomenon in Sec. 1.2.2

## 1.2 Implementation of customized KV-cache

In this subsection, I implement the custom KV-cache mechanism on the *openai-community/gpt2* (Radford et al. [3]) model. I also conduct experiments to compare the performance of my own implementation with the golden implementation provided by *transformers* package. The detailed configures of the experiments are the same as 1.1.

### 1.2.1 Experimental results

The main experimental results are shown in Fig. 3. Note that the orange line in the first sub-figure of Fig. 3 is hidden by the green line, for their values are almost the same.

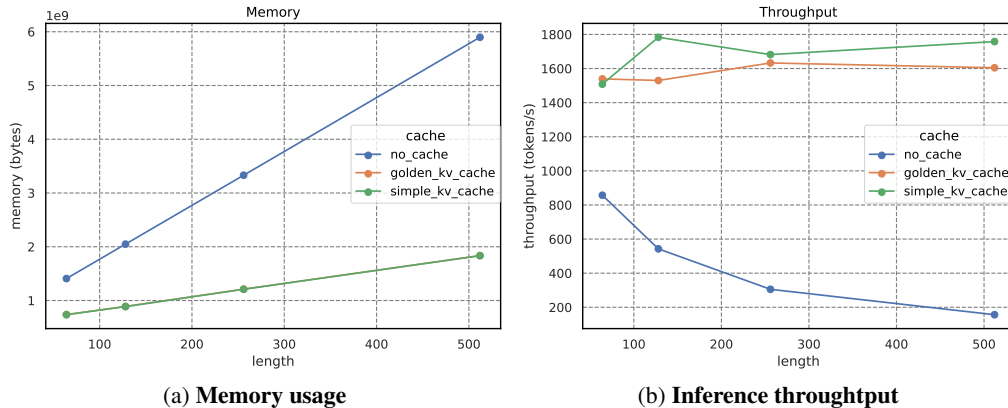


Figure 3: Plots for the GPU memory usage and throughput of the model under different settings.

### 1.2.2 Discussion

As is shown in Sec. 1, our self-implemented KV-cache displays similar performance as the golden implementation. However, the counter-intuitive phenomenon that decoding with KV-cache leads to less GPU memory occupation is akin to that appears in Sec. 1.1. Since our implementation of KV-cache is in essence a simple modification of the original GPT2Model class without any optimization and we use the same pipeline as that of decoding without KV-cache, we can rule out the possible impact of special internal optimization methods. In my opinion, the phenomenon is presumably caused by the reduced calculation due to KV-cache. More specifically, the original inner calculation process of key matrices, value matrices and query matrices, as well as activations might take up large amounts of GPU memory and thus increasing the peak GPU memory occupation.

Besides, it is noteworthy that the self-implemented KV-cache mechanism demonstrates performance comparable to that of the golden implementation, and in my experiments, it even surpasses it.

## 2 LLM reasoning techniques

In this section, I delve into several classic LLM prompting techniques to enhancing the reasoning capability of LLM. I leverage *deepseek-chat* model as the base model and test its performance on GSM8K dataset (Cobbe et al. [1]) with four different prompting methods: (1) vanilla prompting, (2) Chain-of-thought (CoT) prompting, (3) In-Context Learning (ICL) prompting, (4) Reflexion. Note that due to API quota restriction, I only test the model on the first 500 cases of the test set of GSM8K dataset.

## 3 Detailed configurations

For CoT prompting, I just use extra prompt "Let's think stp by step." as proposed in Wei et al. [4]. For ICL prompting, I sample one or five examples from the training set of GSM8K dataset for demonstrating. For Reflexion method, I integrate CoT and ICL prompting to prompt the actor model in order that it can generate well-formatted mathematic. For all prompting methods, I additionally add necessary output format requirements at the end of the prompt. With respect to Reflexion, I use a automatic calculator as the evaluator, which will simply check the validation of the mathematic expressions to give a rough judgment. The actor and self-reflection parts in my implementation of Reflexion are the same LLM API, but with different prompting strategies. For the actor, I incorporate the idea of ICL and CoT methods to elicit trackable mathematic expressions and trajectories. For the self-reflection model, I specially prompt it to output a final judgment for the given solution and use this judgment to determine whether to iterate or to cease the Reflexion procedure.

## 4 Experimental results

The experimental results are listed in Tab. 1. Note that the example prompts and responses as well as corresponding analyses are shown in Appendix A.

Prompting Strategy	Accuracy
vanilla	95.6
CoT	94.4
ICL (1-shot)	94.2
ICL (5-shot)	95.2
Reflexion (1-shot)	94.4
Reflexion (5-shot)	95.0

Table 1: Accuracy of different prompting strategies over the first 500 cases of the test set.

## 5 Discussion

The results shown in Tab. 1 are absolutely counter-intuitive. Even though the four prompting strategies all exhibit great performance, the vanilla-prompted model reaches the highest accuracy in the test. From my perspective, there are several presumed causes. In consideration of the impressive

performance of the vanilla-prompted model, the model itself is well post-trained, the mathematic and reasoning capability of which is particularly enhanced via some possible approaches (e.g., [2]). Since GSM8k was firstly introduced three years ago, inevitable data contamination is another very likely cause. The aforementioned two causes account for the superior performance of the vanilla model. Artificial prompts can mislead the internal knowledge and rationale of well-aligned LLM, which explain the failure of other advanced prompting strategies.

## References

- [1] Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems, 2021. URL <https://arxiv.org/abs/2110.14168>. 3
- [2] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022. 4
- [3] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019. 2
- [4] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022. 3
- [5] Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, Todor Mihaylov, Myle Ott, Sam Shleifer, Kurt Shuster, Daniel Simig, Punit Singh Koura, Anjali Sridhar, Tianlu Wang, and Luke Zettlemoyer. Opt: Open pre-trained transformer language models, 2022. 1

## A Case analysis

I find an interesting phenomenon during the experiment that there some cases in the test set where vanilla prompting fails but other advanced strategies pass. I will give insights into this special kind of cases.

The special case below sheds light on the reasoning capability of the given model. Since it requires the awareness that the increased investment does not include the repair cost to correctly solve this problem, the weakness of the vanilla prompting that it lacks complex reasoning ability and common sense awareness in comparison to other prompting strategies. The other three strategies exhibit different advantages in the example. CoT prompting elicits more comprehensive and detailed rationales. ICL prompting enhance the model's awareness of commonsense and ground the model with golden thoughts structure for solving this kind of problem. Reflexion frame force the model to analyze the previous response and thus detect the possible errors. Despite all the advantages, vanilla prompting reaches the highest total accuracy. I think it is because GSM8K dataset is so simple that the weakness of vanilla prompting is covered up by its other stronger abilities (e.g. natural language understanding ability).

Case 1: An example where vanilla prompting fails but other strategies pass.

### {Problem}

Josh decides to try flipping a house. He buys a house for \$80,000 and then puts in \$50,000 in repairs. This increased the value of the house by 150%. How much profit did he make?

### {Reference Solution}

The cost of the house and repairs came out to  $80,000+50,000=\$80000+50000=130000$  130,000

He increased the value of the house by  $80,000*1.5=\$80000*1.5=120000$  120,000

So the new value of the house is  $120,000+80,000=\$120000+80000=200000$  200,000

So he made a profit of  $200,000-130,000=\$200000-130000=70000$  70,000

#### 70000

### {Vanilla Response}

To solve the problem step by step, let's follow the given information and calculate the profit Josh made from flipping the house.

1. \*\*Initial Purchase and Repairs Cost:\*\*

- Josh buys the house for \$80,000.

- He spends \$50,000 on repairs.

- Total cost = \$80,000 (purchase) + \$50,000 (repairs) = \$130,000.

2. \*\*Increased Value of the House:\*\*

- The value of the house increased by 150%.

- To find the new value, we calculate 150% of the total cost.

- 150% of \$130,000 =  $1.5 * \$130,000 = \$195,000$ .

3. \*\*Profit Calculation:\*\*

- The new value of the house is \$195,000.

- The total cost to Josh was \$130,000.

- Profit = New value - Total cost =  $\$195,000 - \$130,000 = \$65,000$ .

Therefore, the profit Josh made from flipping the house is:

#### \$65,000

### {CoT Response}

Let's break down the problem step by step to find out how much profit Josh made.

1. \*\*Initial Investment:\*\*

- Josh buys the house for \$80,000.

- He spends an additional \$50,000 on repairs.

- Therefore, his total investment is:  $80,000 + 50,000 = 130,000$

2. \*\*Increased Value of the House:\*\*

- The repairs increased the value of the house by 150%.

- To find the new value of the house, we need to calculate 150% of the initial value of the house (before repairs).

- The initial value of the house before repairs is \$80,000.

- 150% of \$80,000 is:  $80,000 \text{ times } 1.5 = 120,000$

- Therefore, the total value of the house after repairs is:  $80,000 + 120,000 = 200,000$

3. \*\*Profit Calculation:\*\*

- The total value of the house after repairs is \$200,000.

- Josh's total investment was \$130,000.

- The profit he made is the difference between the total value of the house and his total investment:  $200,000 -$

$130,000 = 70,000$

So, the profit Josh made is \$70,000.

#### \$70,000

{ICL Response}

To solve the problem, we need to determine the profit Josh made after flipping the house. Here are the steps:

1. \*\*Calculate the total cost of the house:\*\*

- Purchase price: \$80,000

- Repairs cost: \$50,000

- Total cost = Purchase price + Repairs cost

Total cost =  $80,000 + 50,000 = 130,000$

2. \*\*Determine the new value of the house after the repairs:\*\*

- The value increased by 150%.

- New value = Original value + (150% of Original value)

New value =  $80,000 + (1.5 \text{ times } 80,000) = 80,000 + 120,000 = 200,000$

3. \*\*Calculate the profit:\*\*

- Profit = New value - Total cost

Profit =  $200,000 - 130,000 = 70,000$

Thus, the profit Josh made is:

#### 70,000

{Reflexion Response}

response-1

Let's break down the problem step by step.

1. \*\*Initial Cost\*\*: Josh buys the house for \$80,000.

2. \*\*Repairs Cost\*\*: He spends an additional \$50,000 on repairs.

3. \*\*Total Investment\*\*: The total amount invested in the house is the sum of the initial cost and the repairs cost.

First, we calculate the total investment:

Total investment = 80,000

(initial cost) + 50,000 (repairs cost) =  $80,000 + 50,000 = 130,000$  dollars

We need to determine the new value of the house after the repairs. The problem states that the value of the house increased by 150%. This means the new value is 100% (original value) + 150% (increase) = 250% of the original value.

We calculate the new value of the house:

New value = 130,000

(total investment)

times 2.5 =  $130,000 \times 2.5 = 325,000$

dollars

Finally, we calculate the profit made by Josh by subtracting the total investment from the new value of the house:

Profit = 325,000

(new value) - 130,000

(total investment) =  $325,000 - 130,000 = 195,000$

dollars

#### 195,000

reflection-1

Further Analysis of the Solution

The solution provided for the problem is based on a misunderstanding of the term 'value increase by 150%'. The problem states that the value of the house increased by 150%, but this increase is not relative to the total cost (initial cost + repairs). Instead, it is relative to the initial cost of the house alone.

Let's re-evaluate the problem with the correct interpretation:

1. \*\*Initial cost of the house\*\*: Josh buys the house for \$80,000.

Initial cost = 80,000

2. \*\*Cost of repairs\*\*: Josh spends \$50,000 on repairs.

Cost of repairs = 50,000

3. \*\*Total cost\*\*: The total cost is the sum of the initial cost and the cost of repairs.

Total cost =  $80,000 + 50,000 = 130,000$

4. \*\*Value increase due to repairs\*\*: The repairs increase the value of the house by 150% of the initial cost.

Value increase = 80,000

times 1.5 = 120,000

5. \*\*New value of the house\*\*: The new value of the house is the initial cost plus the value increase.

New value =  $80,000 + 120,000 = 200,000$

6. \*\*Profit made\*\*: The profit is the difference between the new value of the house and the total cost.

$\text{Profit} = 200,000 - 130,000 = 70,000$

So, Josh made a profit of \$70,000.

Final Judgement: ##### False

response-2

Let's solve the problem step by step.

1. **Initial cost of the house**: Josh buys the house for \$80,000.

Initial cost = 80,000

2. **Cost of repairs**: Josh spends \$50,000 on repairs.

Cost of repairs = 50,000

3. **Total cost**: The total cost is the sum of the initial cost and the cost of repairs.

Total cost =  $80,000 + 50,000 = 130,000$

4. **Value increase due to repairs**: The repairs increase the value of the house by 150% of the initial cost.

Value increase = 80,000

times 1.5 =  $80,000 \times 1.5 = 120,000$

5. **New value of the house**: The new value of the house is the initial cost plus the value increase.

New value =  $80,000 + 120,000 = 200,000$

6. **Profit made**: The profit is the difference between the new value of the house and the total cost.

Profit =  $200,000 - 130,000 = 70,000$

So, Josh made a profit of \$70,000.

##### 70000

reflection-2

Further Analysis

The solution provided follows a logical sequence of steps to determine the profit made by Josh after flipping the house. Let's break down each step and verify the calculations:

1. **Initial Cost of the House**:

Initial cost = 80,000

This is straightforward and correct.

2. **Cost of Repairs**:

Cost of repairs = 50,000

This is also correct.

3. **Total Cost**:

Total cost =  $80,000 + 50,000 = 130,000$

The calculation is correct.

4. **Value Increase Due to Repairs**:

Value increase = 80,000

times 1.5 = 120,000

The value increase is calculated as 150% of the initial cost, which is correct.

5. **New Value of the House**:

New value =  $80,000 + 120,000 = 200,000$

The new value is the sum of the initial cost and the value increase, which is correct.

6. **Profit Made**:

Profit =  $200,000 - 130,000 = 70,000$

The profit is the difference between the new value and the total cost, which is correct.

### Conclusion

The solution is correct and follows the correct steps to determine the profit made by Josh. The calculations are accurate, and the final profit of \$70,000 is correctly determined.

##### True