Unlocking the Future: Exploring Look-Ahead Planning Mechanistic Interpretability in Large Language Models

Tianyi Men[1,2], Pengfei Cao[1,2], Zhuoran Jin[1,2], Yubo Chen[1,2,*], Kang Liu[1,2,3], Jun Zhao[1,2]
[1]The Key Laboratory of Cognition and Decision Intelligence for Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing, China
[2]School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing, China
[3]Shanghai Artificial Intelligence Laboratory
{tianyi.men, pengfei.cao, zhuoran.jin, yubo.chen, kliu, jzhao}@nlpr.ia.ac.cn

Abstract

Planning, as the core module of agents, is crucial in various fields such as embodied agents, web navigation, and tool using. With the development of large language models (LLMs), some researchers treat large language models as intelligent agents to stimulate and evaluate their planning capabilities. However, the planning mechanism is still unclear. In this work, we focus on exploring the look-ahead planning mechanism in large language models from the perspectives of information flow and internal representations. First, we study how planning is done internally by analyzing the multi-layer perception (MLP) and multi-head self-attention (MHSA) components at the last token. We find that the output of MHSA in the middle layers at the last token can directly decode the decision to some extent. Based on this discovery, we further trace the source of MHSA by information flow, and we reveal that MHSA mainly extracts information from spans of the goal states and recent steps. According to information flow, we continue to study what information is encoded within it. Specifically, we explore whether future decisions have been encoded in advance in the representation of flow. We demonstrate that the middle and upper layers encode a few short-term future decisions to some extent when planning is successful. Overall, our research analyzes the look-ahead planning mechanisms of LLMs, facilitating future research on LLMs performing planning tasks.

1 Introduction

Planning is the process of formulating a series of actions to transform a given initial state into a desired goal state (Valmeekam et al., 2024; Zhang et al., 2024). As the core module of agents (Xi et al., 2023; Wang et al., 2024a), planning has been widely applied in many fields such as embodied agents (Shridhar et al., 2020; Wang et al., 2022),

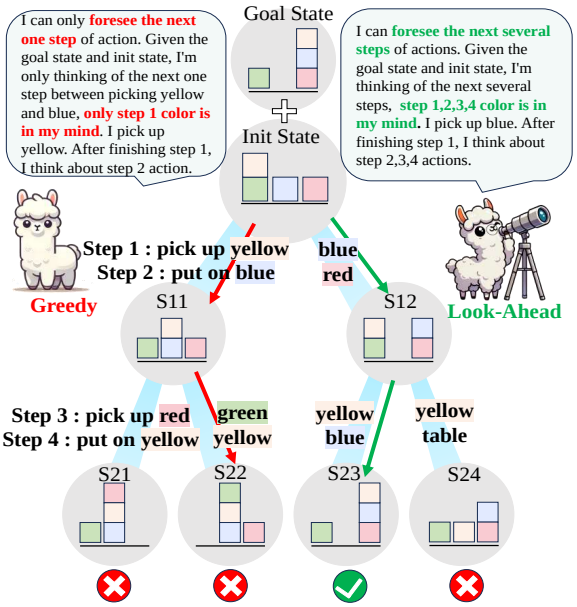

Figure 1: An example of greedy and look-ahead planning.

web navigation (Zhou et al., 2023; Deng et al., 2024) and tool using (Xu et al., 2023; Qin et al., 2023). With the development of large language models, some researchers treat large language models as intelligent agents to solve complex tasks. This is because large language models may possess some preliminary planning capabilities (Huang et al., 2022). Recently, researchers have made efforts to stimulate and evaluate the planning capabilities of large language models. They propose prompt engineering (Zheng et al., 2023) and instruction fine-tuning (Zeng et al., 2023) to boost the planning abilities of large language models. Additionally, some researchers construct benchmarks such as AgentBench (Liu et al., 2023) and AgentGym (Xi et al., 2024) to evaluate the planning capabilities of large models. Although they have made some progress, the underlying mechanisms in planning capabilities of large language models remain a largely unexplored frontier. Revealing

*Corresponding authors.

Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, pages 7713–7724
November 12-16, 2024

the planning mechanisms of large language models helps to better understand and improve their planning capabilities. Therefore, we focus on exploring the underlying mechanisms behind the planning abilities of large language models.

In this work, we focus on exploring look-ahead planning mechanisms in large language models. We study the classical planning task Blocksworld, which is a fully-observed setting. All entity states are known from the init state and goal state, so exploration is not needed (Zhang et al., 2024). As illustrated in Figure 1, given an initial state and a goal state of Blocksworld, the model can only pick up or put down one block. The model must generate a sequence of actions to transform the initial state into the goal state, as shown by the green path. However, it is still unclear whether the model, at step t, greedily considers only the action at $t+1$ or look-ahead considers the actions at $t+2$ and beyond. Inspired by psychology, humans engage in look-ahead thinking when making plans (Baumeister et al., 2016). Based on this, we further propose the hypothesis of model look-ahead planning, which is as follows:

- **Look-Ahead Planning Decisions Existence Hypothesis**: In the task of planning with large language models, given a rule, an initial state, a goal state, and task description prompts. At the current step, the model needs to predict the next action, the probe can detect decisions to some extent for future steps in the internal representations in the short term within a fully-observed setting when planning is successful.

We design a two-stage paradigm to verify this hypothesis. It can be divided into the finding information flow stage and the probing internal representations stage. The first stage is to analyze the information flow and component functions during planning (§5). The second stage is examining whether the model stores future information in internal representations (§6). The specifics are as follows:

(1) In the first stage, we study how planning is done internally by analyzing the MLP and MHSA components at the last token. Inspired by methods of calculating extraction rates methods (Geva et al., 2023), we find the output of MHSA in the middle layers at the last token can directly decode the correct colors to some extent (§5.1). Based on this discovery, we further investigate the sources of information on MHSA. We trace the source of the decisions. And find that planning mainly depends on spans of the goal states and recent steps (§5.2).

(2) In the second stage, we study what information is encoded in the information flow and whether this information has been considered in advance for future decisions. For future decisions existence, we use the probing method to probe future decisions and reveal that the middle and upper layers encode a few short-term future decisions when planning is successful (§6.1). For history step causality, we prevent the information flow from history steps and explore the impact of different history steps on the final decision (§6.2).

In summary, our contributions are as follows:

- To the best of our knowledge, this work is the first to investigate the planning interpretability mechanisms in large language models. We demonstrate the **Look-Ahead Planning Decisions Existence Hypothesis**.
- We reveal that the internal representations of LLMs encode a few short-term future decisions to some extent when planning is successful. These look-ahead decisions are enriched in the early layers, with accuracy decreasing as planning steps increase.
- We prove that MHSA mainly extracts information from spans of the goal states and recent steps. The output of MHSA in the middle layers at the last token can directly decode the correct decisions partially in planning tasks.

2 Experimental Setup

In this paper, we study the Blocksworld task in a fully-observed setting where all entity states are known from the init state and goal state, so exploration is not needed (Zhang et al., 2024). Given a rule R, an initial state S_{init}, a goal state S_{goal}, task description prompts C, the current step t, history $a_1 \dots a_t$, model needs to predict the next action a_{t+1} in accordance with its generative distribution $p(a_{t+1} \mid R, S_{\text{init}}, S_{\text{goal}}, C, a_1 \dots a_t)$ (Hao et al., 2023). In this paper, all inputs are in text form. All inferences are performed using the teacher-forcing method. Previous evaluation works (Valmeekam et al., 2023) mainly involved generating a complete plan and then placing it into the environment for assessment. However, since our primary focus is on open-source models, we have reduced the difficulty by using a fill-in-the-blank format for evaluating the models. An example is shown in Figure 2.

Data Previous Blocksworld evaluation benchmarks (Valmeekam et al., 2023) put the plans generated by models into an environment to verify the correctness. However, existing interpretability methods, such as information flow (Wang et al., 2023a), require gold labels. Therefore, we synthesize a dataset containing optimal plans, with specific data statistics shown in Table 1. We generate data with 4, 5, and 6 color varieties, 4 piles, and a maximum of 6 steps, where pick-up and stack are considered as two different steps. There are three levels: LEVEL1 (L1) with two steps, LEVEL2 (L2) with four steps, and LEVEL3 (L3) with six steps. We choose the optimal path from the initial step to the final step. For samples with multiple optimal paths, we select one to include in the training set, ensuring that samples in the test set have unique optimal paths. We split the dataset into training and test sets with a ratio of 1:3.

Metric In the Blocksworld task, we use two metrics: single-step success rate and complete plan success rate. The single-step success rate evaluates whether each individual action is correct, defined as:

$$S_{\text{step}} = \frac{1}{N}\sum_{i=1}^{N} r_i \tag{1}$$

where N is the total number of steps and r_i indicates the success of the i-th step (1 for success, 0 otherwise). The complete plan success rate evaluates whether the entire planning process is correct, defined as:

$$S_{\text{plan}} = \frac{1}{M}\sum_{j=1}^{M} R_j \tag{2}$$

where M is the total number of tested plans and R_j indicates the success of the j-th plan (1 for complete success, 0 otherwise).

Model We evaluate two large language models: Llama-2-7b-chat (Touvron et al., 2023) and Vicuna-7B (Chiang et al., 2023). Since open-source models have preliminary planning capabilities, we enhance the ability of large language models to complete planning tasks through training.

Experiment Setting We conduct full parameter fine-tuning on Llama-2-7b-chat-hf and Vicuna-7B for 3 epochs. The training process involves a global batch size of 20, using the Adam optimizer with a learning rate of 5e-5. Llama-2-7b-chat-hf and Vicuna-7B achieve complete plan success rates of 61% and 63%, respectively, at LEVEL 3 with 6 blocks. We sample 400 correct data points from LEVEL 3 with 6 blocks for our analysis. We conduct experiment based on HuggingFace's Transformers[1], PyTorch[2], baukit [3] and pyvene[4] (Wu et al., 2024b).

Rule:
You can pick-up color1. stack color1 on-top-of color2. All the blocks are on the table. There is no order in the piles. Please output the optimal plan.
Init state:
<empty>
<black>
<white on gray on red on blue>
<green>
Goal state:
<gray on black>
<red on blue>
<green>
<white>
Plan:
step 1: pick-up ____ (answer: **white**)
step 2: stack white on-top-of ____ (answer: **table**)
step 3: pick-up ____ (answer: **gray**)
step 4: stack gray on-top-of ____ (answer: **black**)

Figure 2: An example of Blocksworld.

3 Background

A transformer-based language model begins by converting an input text into a sequence of N tokens, denoted as $s_1, \ldots, s_N$. Each token s_i is mapped to a vector $x_i^0 \in \mathbb{R}^d$. $E \in \mathbb{R}^{|V|\times d}$ is the decoder matrix in the last layer, where V is the vocabulary, d is embedding dimension. Each layer comprises a multi-head self-attention (MHSA) sublayer followed by a multi-layer perception (MLP) sublayer (Vaswani et al., 2017). Formally, the representation x_i^ℓ of token i at layer ℓ can be obtained as follows:

$$\mathbf{x}_i^\ell = \mathbf{x}_i^{\ell-1} + \mathbf{attn}_i^\ell + \mathbf{m}_i^\ell \tag{3}$$

$\mathbf{attn}_i^\ell$ and $\mathbf{m}_i^\ell$ represent the outputs of the MHSA and MLP sub-layers of the ℓ-th layer, respectively. By using E, an output probability distribution can be obtained from the final layer representation:

$$p_i^L = \text{softmax}\left(Ex_i^L\right) \tag{4}$$

[1] https://github.com/huggingface/transformers/
[2] https://github.com/pytorch/pytorch/
[3] https://github.com/davidbau/baukit/
[4] https://github.com/stanfordnlp/pyvene

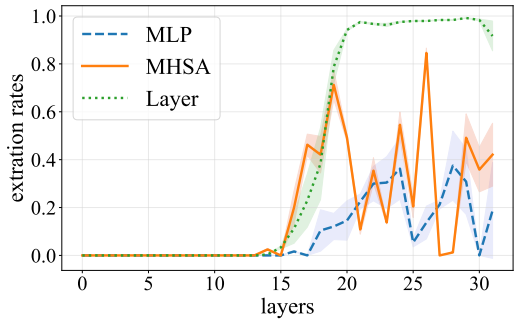

Figure 3: Extraction rate of different components in Llama-2-7b-chat-hf.

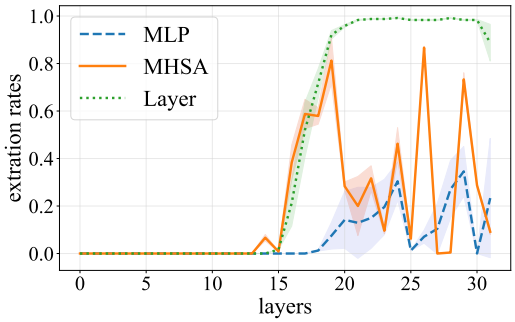

Figure 4: Extraction rate of different components in Vicuna-7b.

4 Overview of Analysis

We analyze the look-ahead planning mechanisms of the models from two stages. (1) In the first stage, we explore the internal mechanisms of this process in planning tasks from the perspectives of information flow and component functions. We demonstrate that the middle layer MHSA can directly decode the answers to a certain extent, and we prove that MHSA mainly extracts information from spans of the goal states and recent steps (§5). (2) In the second stage, to determine the presence of future decisions, we employ the probing method to examine future decisions, uncovering that the intermediate and upper layers encode these decisions. Regarding the causality of historical steps, we inhibit the information flow from past steps and analyze the effects of different historical steps on the ultimate decision (§6).

5 Information Flow in Planning Tasks

To trace the source of the correct answer, we begin with the last token. For example, in the first step "pick up", the last token is "up". The model should process the initial state, target state, and history of steps to decide which color to pick up, such as "blue". We analyze this process from two perspectives. (1) First, we study MLP and MHSA functions at the last token by extraction rates (Geva et al., 2023). We find that the output of MHSA in the middle layers can directly decode the correct colors to a certain extent (§5.1). (2) Based on this, we further trace the source of the correct colors by information flow (Wang et al., 2023a). From the perspective of early and late planning stages, we prove that MHSA mainly extracts information from spans of the goal states and recent steps (§5.2).

5.1 Attention Extract the Answers

From the perspective of the model's internal components, we analyze the functions of different components of the models. The first question is how the model extracts answers from history. We start from the position of the last token and study the roles of the MLP and MHSA components in the answer generation process. Specifically, we investigate whether different components at different layers can directly decode the final answer.

Experiments We use the extraction rate (Geva et al., 2023) to analyze the functions of different components. Specifically, we calculate the extraction rate:

$$e^* := argmax\left(p_N^L\right) \tag{5}$$

$$\widehat{e} := argmax\left(Eh_N^\ell\right) \tag{6}$$

In this equation, h represents the internal representation of the MLP, MHSA and layer output, N is the position of the last token, ℓ is the layer of models, $\ell \in [1, L]$. When $e^* = \widehat{e}$, it is considered as an extraction event. We calculate the extraction rate of the last token for each layer for each step in the Blocksworld. We then compute the mean and variance of these rates.

Results and Analysis As shown in Figure 3 and Figure 4, we observe that (1) MHSA has a higher extraction rate compared to MLP, indicating that attention is primarily responsible for answer extraction. (2) Layer output gradually forms a stable answer in the middle to upper layers (from the 15th layer to the 20th layer). In these layers, the extraction rate of MHSA is significantly higher than MLP, suggesting that MHSA plays a major role during the decision-making period. (3) The variance in extraction rates across different steps is

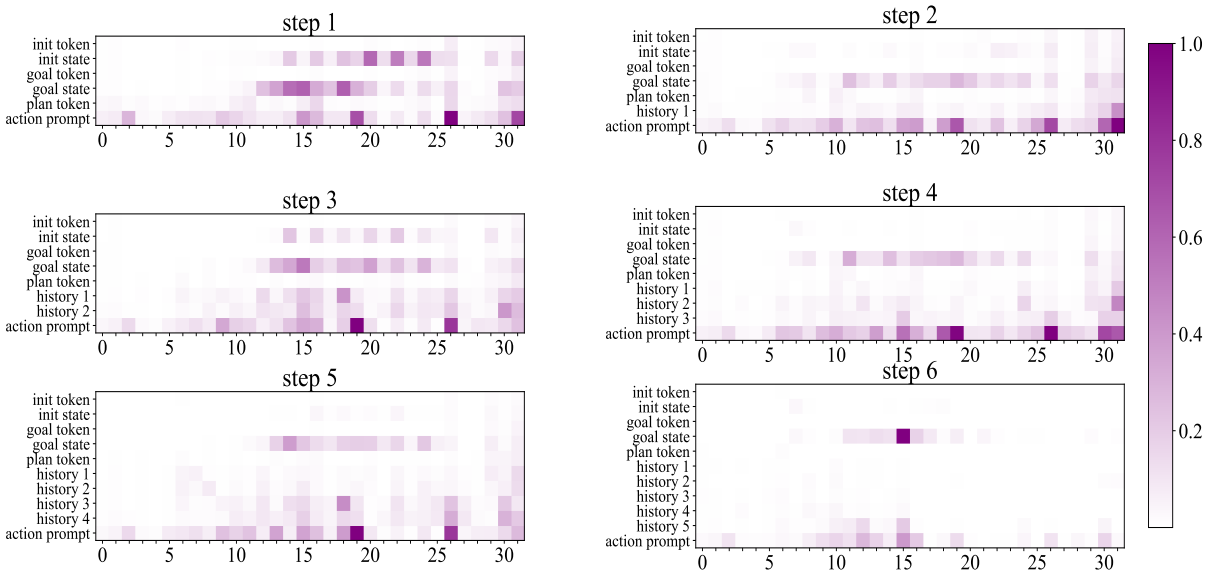

Figure 5: Information flow of last token in Llama-2-7b-chat-hf.

smaller for MHSA compared to MLP, indicating that MHSA layers show higher consistency across different steps.

5.2 Attention Extract from Goal and History

In the previous section, we discover that MHSA is responsible for extracting answers from the context, but which chunk to extract the answer from is still unclear. In this section, we decompose the input into several chunks to identify which chunk MHSA primarily relies on. We use the information flow method (Wang et al., 2023a), first calculating the information flow at the token granularity, and then taking the average of different tokens within the same chunk to represent the information flow at the chunk granularity. This will help us locate the influence of different chunks on the last token.

Experiments We calculate the information flow between layers. Specifically, for the input, we divide it into different chunks, including init token (which is "Init:"), init state (which is "<blue on red>"), target token, target state, six history steps (For step 1, which is "step 1: pick-up white"), action prompt (pick-up or stack on-top-of) and last token. We calculate the information flow $I_{token,\ell}$ for each token at the $\ell - th$ layer. The specific calculation method is as follows:

$$I_{token,\ell}(i,j) = \left| \sum_{hd} A_{hd,l}(j,i) \odot \frac{\partial L(x)}{\partial A_{hd,\ell}(j,i)} \right| \quad (7)$$

Where $A_{hd,l}$ is the attention score of the ℓ-th layer, hd is the hd-th head, and $L(x)$ is the loss function. Here, we use $I(i,j)$ to represent the score flowing from token j to token i. Based on the token information flow, we calculate the chunk information flow, denoted as $I_{chunk,\ell}$:

$$I_{chunk,\ell} = \frac{\sum_{i=k_1}^{k_2} \sum_{j=t_1}^{t_2} I_{token,\ell}(i,j)}{(k_2 - k_1 + 1)(t_2 - t_1 + 1)} \quad (8)$$

Specifically, we consider the information flow from the span $[k1, k2]$ of a chunk k to the span $[t1, t2]$ of another chunk t. We calculate the average of information flow from chunk k to t. Due to the causal attention, we only compute the information flow for the lower triangular matrix. We calculate the chunk information flow for each prediction step.

Results and Analysis The results are shown in Figure 5 and Figure 16. The vertical axis represents the information flow from the chunk to the last token. The horizontal axis represents the information flow at layer ℓ. The values inside represent the scores of information flow. We calculate the information flow for six decision steps. It shows that: (1) In steps 1 to 6, the goal states are highlighted at each step. This indicates that MHSA extracts information from the goal state, demonstrating that it mainly relies on goal states. (2) Taking the step 5 as an example, history 3 and history 4 are more prominent compared to history 1 and history 2. It reveals that MHSA also mainly relies on recent history rather than earlier spans of steps.

6 Internal Representations Encode Planning Information

Based on the previous sections, we discover that MHSA directly extracts answers from the context, but it is still unclear what information is encoded in internal representations. In this section, we demonstrate the look-ahead capability of models from both future decisions existence and history step causality perspectives. (1) For future decisions existence, we use the probing method to probe each layer of the main positions in the context. We find that the accuracy of the current state information gradually decreases as the steps progress. We also find that the middle and upper layers encode future decisions with accuracy decreasing as planning steps increase, proving the look-ahead planning hypothesis (§6.1). (2) For history step causality, we employ a method that involves setting certain information keys of MHSA to zero. We find there is still a probability of generating the correct answer by relying solely on a single step, but it's difficult to support plan for the long-term (§6.2).

6.1 Internal Representations Encode Block States and Future Decisions

In this section, we analyze what information is encoded in the internal representations within the information flow and how this information evolves layer by layer. We examine whether the internal representations encode two types of information (Li et al., 2022; Pal et al., 2023): *Current Block States* and *Future Decisions*. *Current Block States* refer to the state of the blocks at step t. For example, in Figure 1, when following the green path, the *Current Block State* initially starts in the S_{init}. After executing the first and second steps, the internal representation of the *Current Block State* transitions from the S_{init} to S_{12}. *Future Decisions* refer to the information about future decisions at step t. For example, in Figure 1, when following the green path and executing the first step (blue), the question is whether the model's internal representation already stores information about future decisions (red, yellow, blue).

Experiments We probe internal representations of the initial state, goal state, and steps with layer $\ell \in [1, L]$. We train linear probes and nonlinear probes for each chunk and each layer. A linear probe can be represented as $p_\theta(x_n^\ell) = \text{softmax}(W x_n^\ell)$. And a nonlinear probe can be described as $p_\theta(x_n^\ell) = \text{softmax}(W_1 \,\text{ReLU}(W_2 x_n^\ell))$.

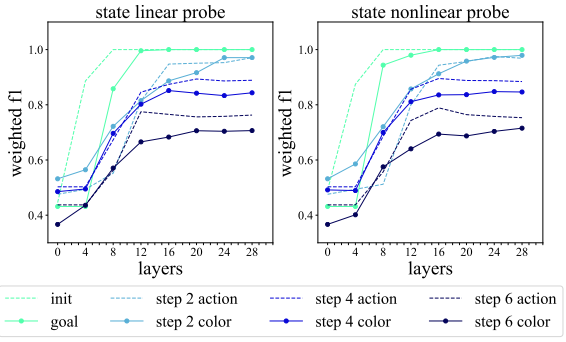

Figure 6: State probe in Llama-2-7b-chat-hf.

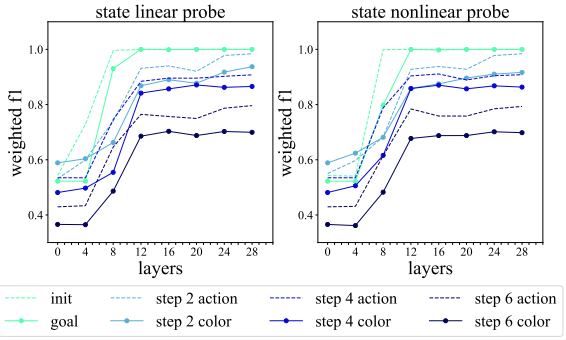

Figure 7: State probe in Vicuna-7b.

Using the linear probe as an example, we consider six steps and six colors of blocks. For *Current Block States*, the input to the probe is a hidden layer representation h of the model. The output is a 12x8 matrix representing probabilities, where 12 denotes the colors of the blocks above and below each color block, and 8 represents 6 colors plus sky and table. For *Future Decisions*, the input to the probe is h. The output is six predicted colors from steps 1 to 6, we only consider future steps in our evaluation. We split the training and test sets in a 4:1 ratio for 400 samples. For the evaluation, we calculate the weighted F1 accuracy for *Current Block States* and accuracy for *Future Decisions*.

Results and Analysis As shown in Figure 6 and Figure 7, the horizontal axis represents the layers probed, while the vertical axis represents the mean accuracy of the probe test. Different colored lines represent the probed spans of states and steps. (1) We observe that as the number of layers increases, the accuracy of the probe gradually improves. This indicates that the early layers of the model are enriching the representation of the current state. (2) The black line (step 6) in the figure has a lower accuracy compared to the light blue line (step 2), demonstrating that as the planning steps progress, the models are difficult to maintain the represen-

tations of the current placement of the blocks. (3) By comparing the linear probe in Figure 6 and the nonlinear probe in Figure 7, we find that both have the same trend, indicating that the model internally stores the current state in a linear manner. A similar trend in *Future Decisions* is shown in Figure 8 and Figure 9 for actions. It reveals that look-ahead decisions are enriched in the early layers.

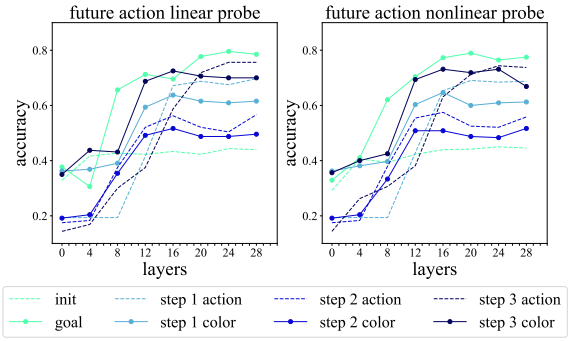

Figure 8: Action probe in Llama-2-7b-chat-hf.

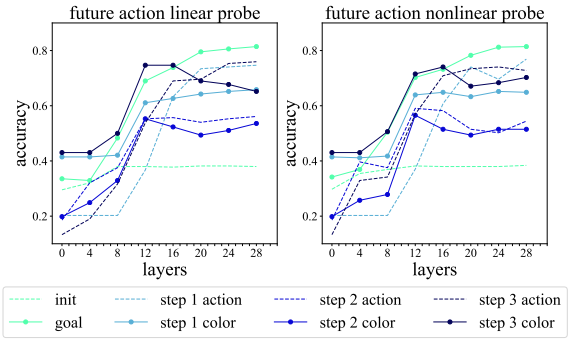

Figure 9: Action probe in Vicuna-7b.

Supplementary Analysis As shown in Figure 10 and Figure 12. They illustrate the accuracy of future decisions based on the current step. Each column represents the current step, while the rows represent the max accuracy of the probe in predicting future answers. We observe the following : (1) For the sixth row and first column, the probe can predict the future sixth step with an accuracy of 0.51 at the first step. This indicates that the model stores information about future decisions in advance, supporting the hypothesis of forward planning. (2) For each row, the values increase from left to right. For example, the accuracy in the fifth column of the sixth row is higher than that in the first column. This means the model is more certain about the output of the sixth step at the fifth step compared to the first step, demonstrating that the model has difficulty in planning over long distances. (3) The accuracy for the first column, representing the prediction accuracy for the next five steps after the initial step, shows a declining trend, indicating that the model stores future decision information in advance, supporting the hypothesis of look-ahead planning decisions existence hypothesis.

6.2 Internal Representations Facilitate Future Decision-making

In this section, we further verify the causal effect of planning information at different steps. We test the causality between planning information in the previous history t_a and decisions in step t_b, where $t_a < t_b$. Specifically, we compare whether the information from step t_1 contributes to the planning in step t_2. If the model is greedy in its planning, there should be no decision information in t_a that can help make better decisions in t_b. Therefore, we set the key of MHSA in historical decision t to 0 to study the causal effect of historical information on future predictions.

Experiments For each step t, we have a history $H_t = [a_1, a_2, ..., a_{t-1}]$, where each step span a_i contains color tokens.

(1) Mask all steps: First, identify all color tokens in H_t, and set the keys to 0 for these colors in each layer of MHSA, resulting in the masked historical information H'_t. The main goal is to stop past decision information from affecting the current decision of the last token. Obtain the decision probability y'_t based on H'_t in t step.

(2) Make one step visible: Based on H'_t, make only the color at position i visible, while masking the other positions, resulting in $H''_{t,i}$. Use $H''_{t,i}$ for prediction, Obtain the decision probability $y''_{t,i}$.

(3) Calculate one step effect: Compare the decision probabilities obtained from masking all steps and from making one step visible to calculate the effect of a single step. The larger this value, the greater the impact of step i on step t:

$$\text{Impact}_{i,t} = y''_{t,i} - y'_t \tag{9}$$

Results and Analysis As illustrated in the Figure 11 and Figure 15, the columns represent the steps visible during prediction, the rows represent the steps being predicted, and the values inside represent the contribution of step t to step i. (1) For example, in the second column of the sixth row, the model can increase the probability of inferring the correct decision in the sixth step by 0.24 just by

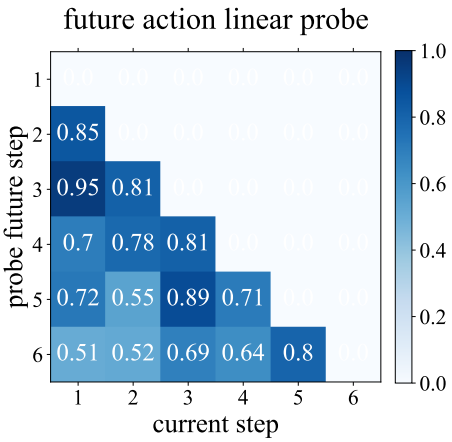

Figure 10: Future action linear probe in Llama-2-7b-chat-hf.

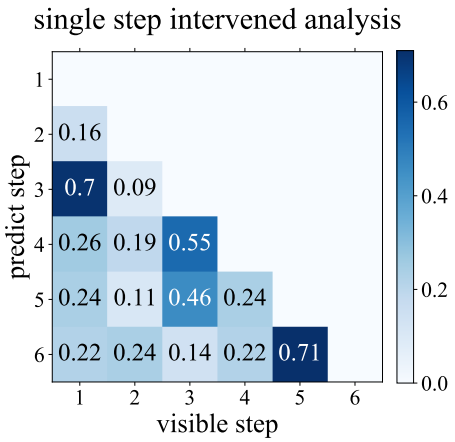

Figure 11: Single step intervened analysis in Vicuna-7b.

using the information from the second step. This indicates that the model is not greedy and is not limited to only preparing for the next step, which causally proves the conclusion of look-ahead planning. (2) Observing the values in each column, for instance, the maximum value in the fifth row is 0.46, located in the third column. This represents that the third step is the most important for predicting the fifth step. It is found that the most important steps for prediction tend to be later steps, indicating that the look-ahead planning ability of LLMs is still relatively preliminary.

7 Related work

LLM-Based Agents With the emergence of large language models, researchers begin to use them as intelligent agents (Xi et al., 2023; Wang et al., 2024a). Significantly, ReAct (Yao et al., 2022) innovatively combines CoT reasoning with agent actions. Some tasks utilize the planning capabilities of large language models through prompt engineering methods (Huang et al., 2022; Hao et al., 2023; Yao et al., 2024; Zhang et al., 2024). Other researchers enhance the planning capabilities of large language models through training methods (Zeng et al., 2023; Chen et al., 2023; Wang et al., 2023b; Yu et al., 2024). Some researchers construct benchmarks to evaluate the planning ability of large language models (Shridhar et al., 2020; Wang et al., 2022; Zhou et al., 2023; Deng et al., 2024; Xu et al., 2023; Qin et al., 2023).

Mechanistic Interpretability Recent works study mechanistic interpretability in factual associations, in-context Learning, and arithmetic reasoning tasks from the perspective of information flow (Geva et al., 2023; Wang et al., 2023a; Stolfo et al., 2023; Jin et al., 2024; Yuan et al., 2024). Researchers also study Othello (Li et al., 2022; Nanda et al., 2023), chess (Karvonen, 2024) and Blocksword (Wang et al., 2024b) in transformer. However, research on the mechanistic interpretation of LLMs performing planning tasks is still unexplored. Our work conducts a study from the perspective of information flow and representation.

Look-Ahead Pal et al. (2023); Wu et al. (2024a); Jenner et al. (2024) demonstrate that it is possible to decode future tokens from the hidden representations at previous token positions. In task planning, a model needs to have look-ahead capabilities. However, it is not clear whether LLMs use similar mechanisms when planning. Our work focuses on the look-ahead mechanisms in planning in LLMs.

8 Conclusion

In this paper, we investigate the mechanisms of look-ahead planning in LLMs through the perspectives of information flow and internal representations. We demonstrate **Look-Ahead Planning Decisions Existence Hypothesis**. Our findings indicate that internal representations of LLMs encode a few short-term future decisions to some extent when planning is successful. These look-ahead decisions are enriched in the early layers, with their accuracy diminishing as the number of planning steps increases. We demonstrate that MHSA mainly extracts information from the spans of goal states and recent steps. Additionally, the output of MHSA in the middle layers at the final token can partially decode the correct decisions.

Limitation

Although our work provides an in-depth analysis and explanation of look-ahead planning mechanisms of large language models, there are several limitations. First, our analytical methods require access to the internal parameters and representations of open-source models. Although black-box large language models such as ChatGPT possess strong planning capabilities, we cannot access their internal parameters, making it challenging to interpret the most advanced language models. Second, our research primarily focuses on the planning mechanisms in Blocksworld. However, many other planning tasks, such as commonsense planning (e.g., "how to make a meal"), lack standard answers, making it difficult to evaluate the correctness of the planning and conduct quantitative analysis. We leave these limitations for future work.

Acknowledgements

This work was supported by the National Key R&D Program of China (No. 2022ZD0160503), the National Natural Science Foundation of China (No. 62176257). This work was also supported by the China Postdoctoral Science Foundation under Grant Number 2024M753500.

References

Roy F Baumeister, Kathleen D Vohs, and Gabriele Oettingen. 2016. Pragmatic prospection: How and why people think about the future. *Review of general psychology*, 20(1):3–16.

Fei-Long Chen, Du-Zhen Zhang, Ming-Lun Han, Xiu-Yi Chen, Jing Shi, Shuang Xu, and Bo Xu. 2023. Vlp: A survey on vision-language pre-training. *Machine Intelligence Research*, 20(1):38–56.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality.

Xiang Deng, Yu Gu, Boyuan Zheng, Shijie Chen, Sam Stevens, Boshi Wang, Huan Sun, and Yu Su. 2024. Mind2web: Towards a generalist agent for the web. *Advances in Neural Information Processing Systems*, 36.

Mor Geva, Jasmijn Bastings, Katja Filippova, and Amir Globerson. 2023. Dissecting recall of factual associations in auto-regressive language models. *arXiv preprint arXiv:2304.14767*.

Shibo Hao, Yi Gu, Haodi Ma, Joshua Jiahua Hong, Zhen Wang, Daisy Zhe Wang, and Zhiting Hu. 2023. Reasoning with language model is planning with world model. *arXiv preprint arXiv:2305.14992*.

Wenlong Huang, Pieter Abbeel, Deepak Pathak, and Igor Mordatch. 2022. Language models as zero-shot planners: Extracting actionable knowledge for embodied agents. In *International Conference on Machine Learning*, pages 9118–9147. PMLR.

Erik Jenner, Shreyas Kapur, Vasil Georgiev, Cameron Allen, Scott Emmons, and Stuart Russell. 2024. Evidence of learned look-ahead in a chess-playing neural network. *arXiv preprint arXiv:2406.00877*.

Zhuoran Jin, Pengfei Cao, Hongbang Yuan, Yubo Chen, Jiexin Xu, Huaijun Li, Xiaojian Jiang, Kang Liu, and Jun Zhao. 2024. Cutting off the head ends the conflict: A mechanism for interpreting and mitigating knowledge conflicts in language models. *arXiv preprint arXiv:2402.18154*.

Adam Karvonen. 2024. Emergent world models and latent variable estimation in chess-playing language models. *arXiv preprint arXiv:2403.15498*.

Kenneth Li, Aspen K Hopkins, David Bau, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. 2022. Emergent world representations: Exploring a sequence model trained on a synthetic task. *arXiv preprint arXiv:2210.13382*.

Xiao Liu, Hao Yu, Hanchen Zhang, Yifan Xu, Xuanyu Lei, Hanyu Lai, Yu Gu, Hangliang Ding, Kaiwen Men, Kejuan Yang, et al. 2023. Agentbench: Evaluating llms as agents. *arXiv preprint arXiv:2308.03688*.

Neel Nanda, Andrew Lee, and Martin Wattenberg. 2023. Emergent linear representations in world models of self-supervised sequence models. *arXiv preprint arXiv:2309.00941*.

Koyena Pal, Jiuding Sun, Andrew Yuan, Byron C Wallace, and David Bau. 2023. Future lens: Anticipating subsequent tokens from a single hidden state. *arXiv preprint arXiv:2311.04897*.

Yujia Qin, Shihao Liang, Yining Ye, Kunlun Zhu, Lan Yan, Yaxi Lu, Yankai Lin, Xin Cong, Xiangru Tang, Bill Qian, et al. 2023. Toolllm: Facilitating large language models to master 16000+ real-world apis. *arXiv preprint arXiv:2307.16789*.

Mohit Shridhar, Xingdi Yuan, Marc-Alexandre Côté, Yonatan Bisk, Adam Trischler, and Matthew Hausknecht. 2020. Alfworld: Aligning text and embodied environments for interactive learning. *arXiv preprint arXiv:2010.03768*.

Alessandro Stolfo, Yonatan Belinkov, and Mrinmaya Sachan. 2023. A mechanistic interpretation of arithmetic reasoning in language models using causal mediation analysis. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 7035–7052.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Karthik Valmeekam, Matthew Marquez, Alberto Olmo, Sarath Sreedharan, and Subbarao Kambhampati. 2024. Planbench: An extensible benchmark for evaluating large language models on planning and reasoning about change. *Advances in Neural Information Processing Systems*, 36.

Karthik Valmeekam, Matthew Marquez, Sarath Sreedharan, and Subbarao Kambhampati. 2023. On the planning abilities of large language models-a critical investigation. *Advances in Neural Information Processing Systems*, 36:75993–76005.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Lean Wang, Lei Li, Damai Dai, Deli Chen, Hao Zhou, Fandong Meng, Jie Zhou, and Xu Sun. 2023a. Label words are anchors: An information flow perspective for understanding in-context learning. *arXiv preprint arXiv:2305.14160*.

Lei Wang, Chen Ma, Xueyang Feng, Zeyu Zhang, Hao Yang, Jingsen Zhang, Zhiyuan Chen, Jiakai Tang, Xu Chen, Yankai Lin, et al. 2024a. A survey on large language model based autonomous agents. *Frontiers of Computer Science*, 18(6):186345.

Ruoyao Wang, Peter Jansen, Marc-Alexandre Côté, and Prithviraj Ammanabrolu. 2022. Scienceworld: Is your agent smarter than a 5th grader? *arXiv preprint arXiv:2203.07540*.

Siwei Wang, Yifei Shen, Shi Feng, Haoran Sun, Shang-Hua Teng, and Wei Chen. 2024b. Alpine: Unveiling the planning capability of autoregressive learning in language models. *arXiv preprint arXiv:2405.09220*.

Xiao Wang, Guangyao Chen, Guangwu Qian, Pengcheng Gao, Xiao-Yong Wei, Yaowei Wang, Yonghong Tian, and Wen Gao. 2023b. Large-scale multi-modal pre-trained models: A comprehensive survey. *Machine Intelligence Research*, 20(4):447–482.

Wilson Wu, John X Morris, and Lionel Levine. 2024a. Do language models plan ahead for future tokens? *arXiv preprint arXiv:2404.00859*.

Zhengxuan Wu, Atticus Geiger, Aryaman Arora, Jing Huang, Zheng Wang, Noah D Goodman, Christopher D Manning, and Christopher Potts. 2024b. pyvene: A library for understanding and improving pytorch models via interventions. *arXiv preprint arXiv:2403.07809*.

Zhiheng Xi, Wenxiang Chen, Xin Guo, Wei He, Yiwen Ding, Boyang Hong, Ming Zhang, Junzhe Wang, Senjie Jin, Enyu Zhou, et al. 2023. The rise and potential of large language model based agents: A survey. *arXiv preprint arXiv:2309.07864*.

Zhiheng Xi, Yiwen Ding, Wenxiang Chen, Boyang Hong, Honglin Guo, Junzhe Wang, Dingwen Yang, Chenyang Liao, Xin Guo, Wei He, et al. 2024. Agentgym: Evolving large language model-based agents across diverse environments. *arXiv preprint arXiv:2406.04151*.

Qiantong Xu, Fenglu Hong, Bo Li, Changran Hu, Zhengyu Chen, and Jian Zhang. 2023. On the tool manipulation capability of open-source large language models. *arXiv preprint arXiv:2305.16504*.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. 2024. Tree of thoughts: Deliberate problem solving with large language models. *Advances in Neural Information Processing Systems*, 36.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2022. React: Synergizing reasoning and acting in language models. *arXiv preprint arXiv:2210.03629*.

Fangxu Yu, Lai Jiang, Haoqiang Kang, Shibo Hao, and Lianhui Qin. 2024. Flow of Reasoning: Efficient Training of LLM Policy with Divergent Thinking. *arXiv e-prints*, arXiv:2406.05673.

Hongbang Yuan, Pengfei Cao, Zhuoran Jin, Yubo Chen, Daojian Zeng, Kang Liu, and Jun Zhao. 2024. Whispers that shake foundations: Analyzing and mitigating false premise hallucinations in large language models. *arXiv preprint arXiv:2402.19103*.

Aohan Zeng, Mingdao Liu, Rui Lu, Bowen Wang, Xiao Liu, Yuxiao Dong, and Jie Tang. 2023. Agenttuning: Enabling generalized agent abilities for llms. *arXiv preprint arXiv:2310.12823*.

Li Zhang, Peter Jansen, Tianyi Zhang, Peter Clark, Chris Callison-Burch, and Niket Tandon. 2024. Pddlego: Iterative planning in textual environments. *arXiv preprint arXiv:2405.19793*.

Longtao Zheng, Rundong Wang, Xinrun Wang, and Bo An. 2023. Synapse: Trajectory-as-exemplar prompting with memory for computer control. In *The Twelfth International Conference on Learning Representations*.

Shuyan Zhou, Frank F Xu, Hao Zhu, Xuhui Zhou, Robert Lo, Abishek Sridhar, Xianyi Cheng, Yonatan Bisk, Daniel Fried, Uri Alon, et al. 2023. Webarena: A realistic web environment for building autonomous agents. *arXiv preprint arXiv:2307.13854*.

A Additional Results

Future action nonlinear probe in Llama-2-7b-chat-hf is shown in Figure 12. Future action linear probe in Vicuna-7b is shown in Figure 13. Future action nonlinear probe in Vicuna-7b is shown in Figure 14, Single step intervened analysis in Llama-2-7b-chat-hf is shown in Figure 15. Data statistics is shown in Table 1. Information flow of last token in Vicuna-7b is shown in Figure 16.

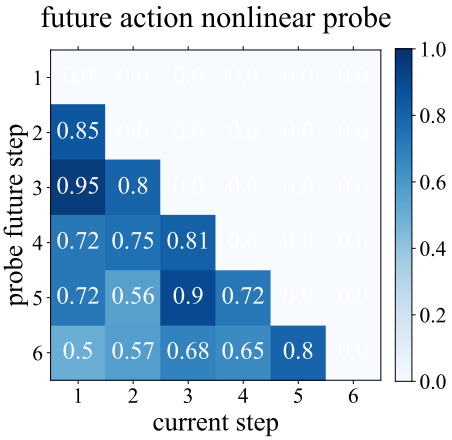

Figure 12: Future action nonlinear probe in Llama-2-7b-chat-hf.

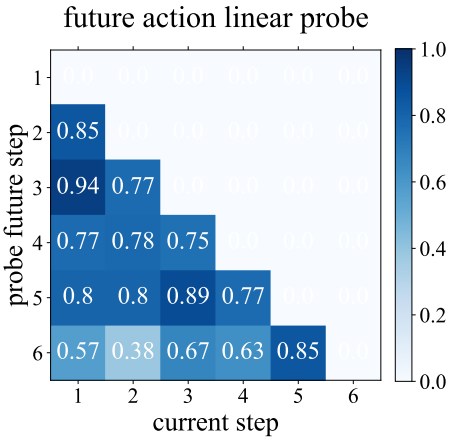

Figure 13: Future action linear probe in Vicuna-7b

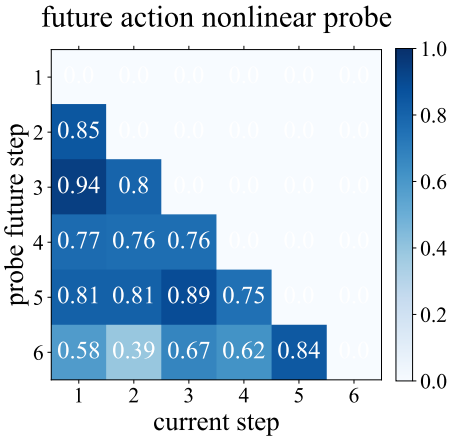

Figure 14: Future action nonlinear probe in Vicuna-7b

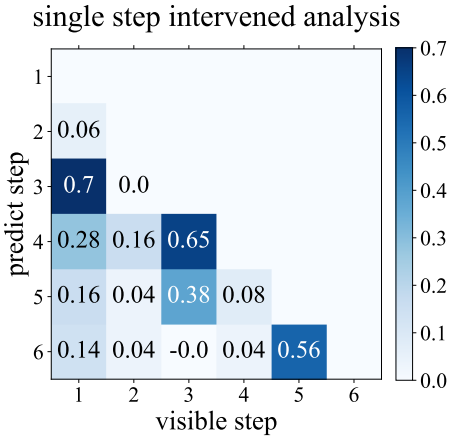

Figure 15: Single step intervened analysis in Llama-2-7b-chat-hf.

LEVEL	L1	L2	L3	Total
Train Size				
4 blocks	3	17	25	45
5 blocks	1	23	121	145
6 blocks	3	48	326	377
Total	7	88	472	567
Test Size				
4 blocks	24	60	80	164
5 blocks	34	115	268	417
6 blocks	57	232	709	998
Total	115	407	1057	1579

Table 1: Blocksworld dataset statistics

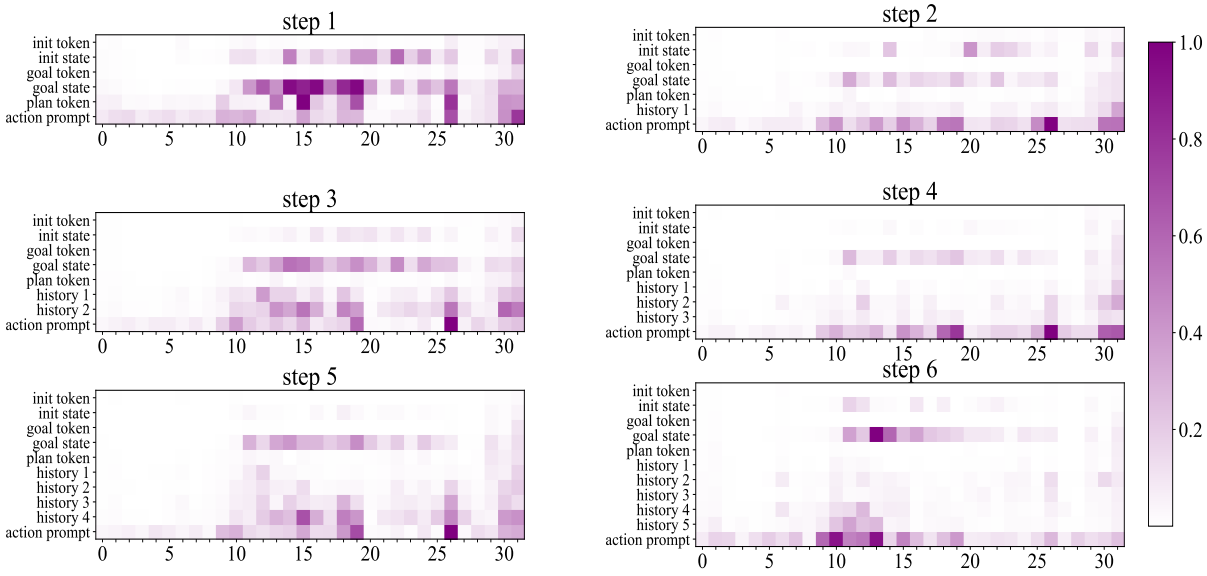

Figure 16: Information flow of last token in Vicuna-7b.