Token Knowledge: A New Perspective For Knowledge in Large Language Models

Jieyong Wang[1,2], **Chunyao Song**[1,2*], **Tingjian Ge**[3],
[1]College of Computer Science, Nankai University, Tianjin, China
[2] TJ Key Lab of NDST, DISSec, TMCC, TBI Center, Nankai University, Tianjin, China
[3]Computer Science Department, University of Massachusetts Lowell, Lowell, USA
jieyongwang@mail.nankai.edu.cn, chunyao.song@nankai.edu.cn
tingjian.ge@gmail.com

Abstract

In the era of prosperity of large language models (LLMs), hallucination remains a serious issue hindering LLMs' expansion and reliability. Predicting the presence (and absence) of certain knowledge in LLMs could aid the hallucination avoidance. However, the token-based generation mode of LLM is different from the knowledge storage structure in the form of triples, which makes it difficult to accurately evaluate the knowledge boundary of LLM. We approach this problem from a novel perspective and, for the first time, introduce the concept of *token knowledge* in large language models. Consequently, we propose a token knowledge dataset construction method and use the intermediate states during inference to train probes. This allows us to predict if a specific token will appear in the LLM's generated sequence, without even generating a single token. Our approach unlocks the model's latent potential, enhancing its accuracy in assessing token knowledge from about 60% to over 90%, with strong out-of-distribution generalization by training on just a few dozen prompts. Finally, we apply KEGT to enhance a state-of-the-art knowledge boundary detection method, achieving improved performance while reducing computational time by over 90%. Furthermore, KEGT enables prevention of hallucinations in certain cases by leveraging its guidance in the token-level knowledge semantic space. Our code is available at https://github.com/CC-2000/KEGT.

1 Introduction

Assessing the knowledge stored in large language models (LLMs) remains a significant challenge (Li et al., 2024). The common method for verifying LLM knowledge involves querying models with fact-based prompts and many studies have employed similar approaches, including comparing responses to gold standard labels (Elazar et al., 2021), evaluating semantic consistency in paraphrased responses (Hase et al., 2023; Raj et al., 2022), and cross-referencing answers with external evidence.

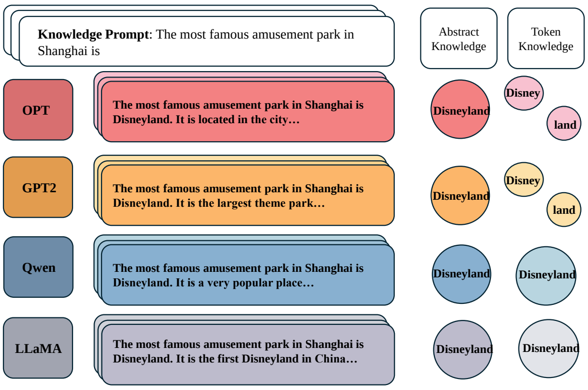

Knowledge Prompt: The most famous amusement park in Shanghai is
Abstract Knowledge
Token Knowledge
OPT
The most famous amusement park in Shanghai is Disneyland. It is located in the city…
Disneyland
Disney
land
GPT2
The most famous amusement park in Shanghai is Disneyland. It is the largest theme park…
Disneyland
Disney
land
Qwen
The most famous amusement park in Shanghai is Disneyland. It is a very popular place…
Disneyland
Disneyland
LLaMA
The most famous amusement park in Shanghai is Disneyland. It is the first Disneyland in China…
Disneyland
Disneyland

Figure 1: The difference between abstract knowledge and token knowledge.

Previous research, regardless of assessing an LLM's mastery of a particular subject (Gottesman and Geva, 2024) or its understanding of various relational templates (Dong et al., 2024), or comparing the generated objects to gold labels, all evaluates an LLM's higher-level knowledge, which we refer to as *abstract knowledge*. As shown in Figure 1, given that the text generated by LLMs is based on tokens as fundamental units (Yu et al., 2024), higher-level entity or relation knowledge is more abstract and harder to obtain compared to token-level knowledge for LLMs. *Our main standpoint is that an LLM's token-level knowledge is easier—both more accurately and more efficiently—to be obtained as a stepstone for high-level abstract knowledge.*

Thus, unlike prior work, this study aims to focus on token-level knowledge in LLMs, specifically, predicting whether a model will generate a given token without actually generating a single token. We show that our approach enables a more accessible analysis of abstract knowledge and has potential applications in probing model knowledge boundaries, preventing hallucinations, and mitigating the

[*]Corresponding author.

Findings of the Association for Computational Linguistics: EMNLP 2025, pages 7912–7926
November 4-9, 2025 ©2025 Association for Computational Linguistics

generation of biased or harmful contents.

We define this problem as Token Knowledge Estimation (Section 2). While directly generating text and verifying the presence of specific tokens in the output is the most accurate method, the time required for generating long sequences is a significant concern. The time complexity of text generation increases non-linearly with sequence length, due to the attention mechanisms in Transformers, which results in $O(n^2)$ time complexity (Duman Keles et al., 2023). Consequently, time efficiency is a crucial factor for token knowledge estimation, and any approach we consider should at least outperform the direct generation method in terms of speed.

To address the token knowledge estimation task, we build on prior work exploring the interpretability of internal states in LLMs. Research has shown that certain internal representations during LLM inference exhibit interpretable features (Geva et al., 2023; Meng et al., 2022), which can be captured using linear probes (Hernandez et al., 2024). Based on this, we propose KEGT (**K**nowledge **E**stimation of **G**enerated **T**oken), a probing that utilizes these internal representations to predict the likelihood of a token appearing in the generated sequence, even without requiring to generate a single token. As a result, KEGT provides a significant efficiency advantage over traditional text generation methods.

To implement KEGT, we refine the KASS dataset (Dong et al., 2024) by selecting 2,992 distinct knowledge facts and 48,770 unique prompts for experiments. We evaluate KEGT in both in-distribution and out-of-distribution settings, where it consistently achieves over 0.9 accuracy. Then we test KEGT with minimal training data and find that it converges and remains highly accurate with just one knowledge fact ($\leq$30 prompts). This shows KEGT captures generalizable features for predicting token appearances in future outputs. Evaluations on eight LLMs and four datasets confirm these features are consistently present and effectively captured by KEGT. Additionally, we apply KEGT to explore the knowledge boundaries of LLMs. Compared to the latest methods for knowledge boundary detection (Dong et al., 2024), our approach is over ten times faster. Finally, we conduct intervention experiments to verify that KEGT can proactively prevent the hallucination, and perform case studies to demonstrate its effectiveness.

Overall, our main contributions are as follows:

1. We introduce a novel perspective for evaluating the knowledge of large language models: token-level knowledge estimation. We then propose KEGT, a method that predicts whether a specific token will appear in the generated sequence, without generating any tokens, effectively addressing this task.

2. We leverage LLM internal representations during inference and capture generalizable features to solve the token-level knowledge estimation task. These features demonstrate strong out-of-distribution generalization and can be learned with minimal data.

3. Experiments demonstrate that KEGT can effectively prevent hallucinations in advance and can be successfully applied to the exploration of knowledge boundaries in LLMs.

2 Token Knowledge Estimation

LLMs generate knowledge based on tokens as fundamental units, while higher-level entity and relation knowledge is more abstract to the model. Aligned with (Li et al., 2024), which highlights the inadequately defined knowledge boundary of LLMs, understanding entity knowledge without exploring token-level mechanisms is challenging. Thus, this work focuses on measuring token-level knowledge in LLMs.

Most methods study abstract knowledge by checking whether model responses consistently include answer entities. Similarly, our approach investigates whether a specific token appears in the model's response to a given prompt. The formal definition of this task is as follows:

Given a prompt p and a specific token t_x, the response $r = [t_1, t_2, \ldots, t_m]$ generated by LLM M consists of m tokens. The goal of token knowledge estimation is to determine whether the token t_x will appear in the token sequence r. Specifically, the task is to identify whether there exists an i such that $1 \leq i \leq m$, $i \in \mathbb{N}$, and $t_i = t_x$.

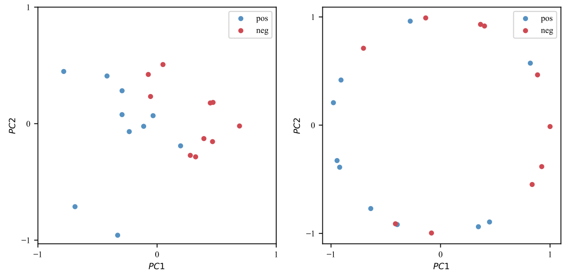
The most direct and accurate method is to input p into M, control M to generate sequences of length m, and then traverse each token in the sequences to check whether t_x appears. However, this approach is time-consuming, as the generation time increases with sequence length, and this increase is not linear—the time complexity of the attention mechanism is $O(m^2)$ (Duman Keles et al., 2023). Thus, time effciency is a crucial factor for token knowledge estimation and an optimal solution should take less time than directly generating a response of length m from M for the prompt p.

3 Methodology

Previous studies have demonstrated that the hidden states in LLMs capture rich task-specific information (Geva et al., 2023; Meng et al., 2022). However, none has explored whether this information also pertains to token knowledge. To address this gap, we first investigate the potential presence of such information using a carefully curated dataset.

We create new prompts (Appendix A) by combining a base prompt p with a specific token t_x, then analyze the LLM's hidden states during inference to determine if they contain task-related features. We select 10 factual prompts as inputs to LLaMA3.1-8B (Dubey et al., 2024), generate 20-token responses and manually review these responses to identify key words as t_x. To facilitate differentiation, we categorize these tokens as t_{pos} and t_{neg}, where t_{pos} is a token that appears in the response and t_{neg} is one that does not. For example, for the prompt "Hepatitis B vaccine is used to immunize against", the response is "hepatitis B virus infection. It is a vaccine that is used to prevent hepatitis B. It is a". We choose t_{pos} = "virus" and specify t_{neg} = "bacteria", creating a triplet $(p, t_{\text{pos}}, t_{\text{neg}})$. Similarly, we replace "May" with "November" and "different" with "same" to create negative tokens, for example. The complete data is provided in Appendix B.

For carefully prepared data, as shown in Figure 2(a), we collect hidden states of M and apply PCA for dimensionality reduction (Shlens, 2014). The resulting two principal components (PCs) are then normalized to a vector with a modulus of 1 (Chen et al., 2024), as shown in Figure 2(b). A clear linear separation is observed between positive and negative samples. Upon normalization—focusing on distribution rather than absolute distance—negative samples tend to cluster on the right, while positive samples appear on the left. This suggests that the LLM contains general features that effectively capture token knowledge for the token knowledge estimation task.



(a) PCA

(b) PCA with Normalization

Figure 2: The visualization for carefully prepared data.

3.1 Token Knowledge Data Preparation

To assess the generalizability of these features, we conduct experiments on large-scale data. Given that LLMs are known to be sensitive to prompt variations, we include paraphrasing prompts whenever possible during dataset construction.

We use the KASS dataset (Dong et al., 2024), where each knowledge triplet $k = (s, r, o)$ contains multiple aliases for the subject and object, as well as various templates for the relation. By combining each subject alias with each relation template, we generate a set of prompts. For each prompt p associated with a knowledge triplet k, we obtain the 20-token response[1] from the LLM M and randomly select a meaningful token t_{pos} to form a positive sample. A negative sample is created by pairing p with a random, meaningful token t_{neg} that does not appear in the response. This approach ensures that for each prompt and each model, one positive and one negative sample are constructed, maintaining a balanced distribution of positive and negative samples in the dataset. Additionally, to ensure the generalizability of our method across different datasets, we construct three additional datasets: two QA datasets—NQ and WebQ—and an Amazon review dataset. The detailed construction and the statistical information of each dataset are provided in Appendix C.

3.2 KEGT: Knowledge Estimation of Generated Token

When a prompt p is fed into the LLM M, the i^{th} token x_i of the prompt is first encoded with word and position embeddings to obtain the initial state $h_i^{(0)} = emb(x_i) + pos(x_i)$. The hidden state of the l-th layer, $h^{(l)}$ is computed by passing $h^{(0)}$ through the Transformer decoder blocks sequentially, where $1 \leq l \leq L$ and L is the total number of Transformer decoder layers. The calculation of $h^{(l)}$ is given by:

$$h_i^{(l)} = h_i^{(l-1)} + a_i^{(l)} + m_i^{(l)} \qquad (1)$$

$$a_i^{(l)} = attn^{(l)}(h_1^{(l-1)}, h_2^{(l-1)}, ..., h_i^{(l-1)}) \qquad (2)$$

[1] The reason why we choose a 20 token response is shown in Appendix D .

$$m_i^{(l)} = W_{proj}^{(l)} \sigma(W_{fc}^{(l)} \gamma(a_i^{(l)} + h_i^{(l-1)}) \quad (3)$$

where $attn^{(l)}$ is the attention layer of the l-th layer, and $a_i^{(l)}$ is the l-th attention layer's output, while $W_{proj}^{(l)}$ and $W_{fc}^{(l)}$ are the parameters of the two fully connected layers of the l-th layer (ignoring bias).

Given the success of other probing works and the distinctly linearly separable features shown in Figure 2, we use the hidden representations computed by the LLM M during inference as input to the probing network:

$$f(\mathbf{h}) := \sigma(\theta \cdot \mathbf{h}) \quad (4)$$

Here, $\mathbf{h}$ can be any $h_i^{(l)}$, but experiments in Subsection 5.1 indicate that selecting features from both the last token and middle layers yields the best probing results, which is consistent with previous conclusions. Finally, we optimize the trainable weight matrix θ by minimizing the cross-entropy loss between the predicted distribution of KEGT and the true labels y.

$$\mathcal{L} = -(y \cdot log(\sigma(\theta \cdot \mathbf{h})) + (1-y) \cdot log(1-\sigma(\theta \cdot \mathbf{h}))) \quad (5)$$

More detailed training parameters for KEGT can be found in Appendix E.

3.3 Intervention Token Knowledge for Hallucination Prevention

Benefiting from the phenomenon observed in Figure 2 and the experimental results presented in Figure 4, we attempt to intervene in the LLM's token knowledge by leveraging the directional information embedded in the hidden representations $\mathbf{h}$.

Specifically, we perform PCA on $\mathbf{h}$, reducing its dimensionality to K and obtaining a principal component matrix P. The low-dimensional projection matrix is then computed as:

$$Z = \mathbf{h} \times P \quad (6)$$

Each row vector in Z represents the corresponding sample's coordinates in the K-dimensional principal component space.

To intervene token knowledge, we introduce a small perturbation along the directions of the principal components. We define a diagonal matrix $A \in \mathcal{R}^{K \times K}$, where each diagonal entry A_i indicates the perturbation strength along the i-th principal direction. Then the perturbed hidden state $\mathbf{h}_{int}$ is given by:

$$\mathbf{h}_{int} = \mathbf{h} + Z \times A \times P^T \quad (7)$$

When $A_i > 0$, the hidden state is shifted toward the positive direction of the i-th main component, making the LLM more likely to exhibit the target token knowledge. Conversely, $A_i < 0$ shifts the state in the opposite direction, effectively removing the corresponding token knowledge from the model's behavior. As demonstrated in the experiments in Section 5.5, our intervention method can effectively reduce hallucination in certain scenarios.

4 Experimental Setup

Data We validate token knowledge in LLMs using KASS with diverse paraphrasing, standard QA datasets with varying styles (NQ and WebQA), and human-written review-style data from Amazon. The detailed construction process is provided in Appendix C.

Models We conduct experiments on eight models across five different model families. Considering the shallow layers in LLMs primarily capture token-level semantic information, while middle layers encode higher-level features, and deep layers are closely related to next-token prediction. Therefore, for each model, we conduct experiments at three levels—shallow, middle, and last layers—and our experimental results consistently align with this theoretical understanding. Details regarding model specifications and layer selection can be found in Appendix F.

Baselines To the best of our knowledge, we are the first to investigate token knowledge in LLMs, thus no direct baseline methods are available for comparison. However, given the recent advances in LLMs' zero-shot and few-shot learning capabilities (Patel et al., 2023), we use these as baseline methods. Additionally, we consider CCS, an unsupervised approach that evaluates factual accuracy through hidden representations of LLMs (Burns et al., 2023). Although CCS was originally designed for fact verification, its method is generalizable and can be applied to any probing task on LLM representations. For a fair comparison, we train CCS unsupervisedly on the same layers used in KEGT, referred to as CCS-shallow, CCS-middle, and CCS-last.

Evaluation For fairness, we use the same prompt templates for both KEGT and the three baselines. We use accuracy and F1 score as metrics.

method	LLaMA3.1-8B		LLaMA2-13B		GPT2-MEDIUM		GPT2-XL	
	accuracy	f1 score	accuracy	f1 score	accuracy	f1 score	accuracy	f1 score
zero-shot	0.0069	0.0052	0.5513	0.5780	0.0467	0.0863	0.4629	0.5312
few-shot	0.6380	0.5223	0.7186	0.6589	0.5397	0.4808	0.5997	0.5198
CCS-shallow	0.9643	0.9635	0.2099	0.2070	0.3122	0.3200	0.6730	0.6722
KEGT-shallow	0.9154	0.9081	0.8286	0.8258	0.8893	0.8925	0.8843	0.9101
CCS-middle	0.9830	0.9832	0.9680	0.9687	0.9709	0.9701	0.9795	0.9797
KEGT-middle	0.9793	0.9793	**0.9800**	**0.9802**	**0.9815**	**0.9814**	**0.9856**	**0.9857**
CCS-last	0.3024	0.2900	0.9640	0.9649	0.8396	0.8421	0.8909	0.8927
KEGT-last	**0.9885**	**0.9885**	0.9773	0.9776	0.9125	0.9130	0.9397	0.9399
method	Qwen2.5-1.5B		Qwen2.5-7B		OPT-1.3B		OPT-6.7B	
	accuracy	f1 score	accuracy	f1 score	accuracy	f1 score	accuracy	f1 score
zero-shot	0.6577	0.6916	0.6472	0.4984	0.1196	0.2134	0.2597	0.3235
few-shot	0.5885	0.4053	0.6788	0.5376	0.5359	0.3773	0.5900	0.6392
CCS-shallow	0.0727	0.1326	0.5528	0.5048	0.7540	0.7662	0.3364	0.2971
KEGT-shallow	0.9250	0.9192	0.9524	0.9503	0.7824	0.7835	0.8095	0.8089
CCS-middle	0.0446	0.0629	0.0698	0.1298	0.8195	0.8244	0.2814	0.2697
KEGT-middle	**0.9812**	**0.9820**	0.9899	0.9900	0.9124	0.9137	0.9588	0.9598
CCS-last	0.8280	0.8310	0.5780	0.5776	0.8498	0.8558	0.0222	0.0115
KEGT-last	0.9778	0.9778	**0.9904**	**0.9904**	**0.9696**	**0.9698**	**0.9839**	**0.9839**

Table 1: The out-of-distribution experimental results, with the best results in bold and the second-best results underlined.

5 Results and Analysis

In this section, we evaluate the performance of KEGT and baselines in both in-distribution and out-of-distribution settings, demonstrating that KEGT's extracted features are general and dataset-irrelevant. We also show that KEGT converges with minimal training data. In layer-based experiments, our findings align with previous knowledge storage studies, with explanations for observed differences. Finally, we validate KEGT's effectiveness in LLM knowledge boundary detection task and illustrate its application through a case study.

5.1 Main Results

We first evaluate the performance of KEGT and three baselines on the KASS dataset. We randomly split all knowledge facts into a training set and a test set with an 8:2 ratio. For each knowledge fact in training set, we randomly select 80% of the corresponding paraphrasings for training, reserving the remaining 20% as an independent, identically distributed (IID) test set. This results in a 0.64:0.16:0.2 split for the training, IID and out-of-distribution (OOD) test sets. The IID and OOD results are shown in Table 5 in Appendix G and Table 1, respectively. We can draw the following conclusions.

KEGT outperforms both in-distribution and out-of-distribution methods. KEGT-middle generally yields the best results, with KEGT-last occasionally outperforming it. Compared to zero-shot and few-shot methods, KEGT shows significant improvement, while the unsupervised CCS is less stable. When CCS performs well, such as with LLaMA3.1-8B middle representations, it slightly surpasses KEGT-middle but falls behind KEGT-last. In cases where CCS underperforms, KEGT consistently outperforms CCS, demonstrating its stable, optimal performance for robustness.

LLMs possess token knowledge but it is hidden and unrevealed, which can be captured by KEGT. Zero-shot and few-shot achieve accuracies of 40% to 65%, close to random guessing, indicating that LLMs do not expose token knowledge. Despite using the same prompt templates, KEGT shows strong predictive ability for token knowledge, suggesting that LLM representations during inference contain features related to token prediction, which LLMs do not explicitly propagate. KEGT can effectively capture these features.

5.2 The Salient Features of Token Knowledge

LLMs reveal salient token knowledge features, and KEGT effectively captures these features. While it may seem that KEGT requires large amounts of training data to converge, we provide evidence to the contrary: (1) KEGT requires only a small amount of training data to identify the learning direction; (2) KEGT learns similar directions across prompts from different distributions; (3) This direction can also be identified in less refined datasets, such as casually written human comment data.

KEGT requires minimal training data. While KEGT typically performs optimally with larger datasets, we show that its learning direction can

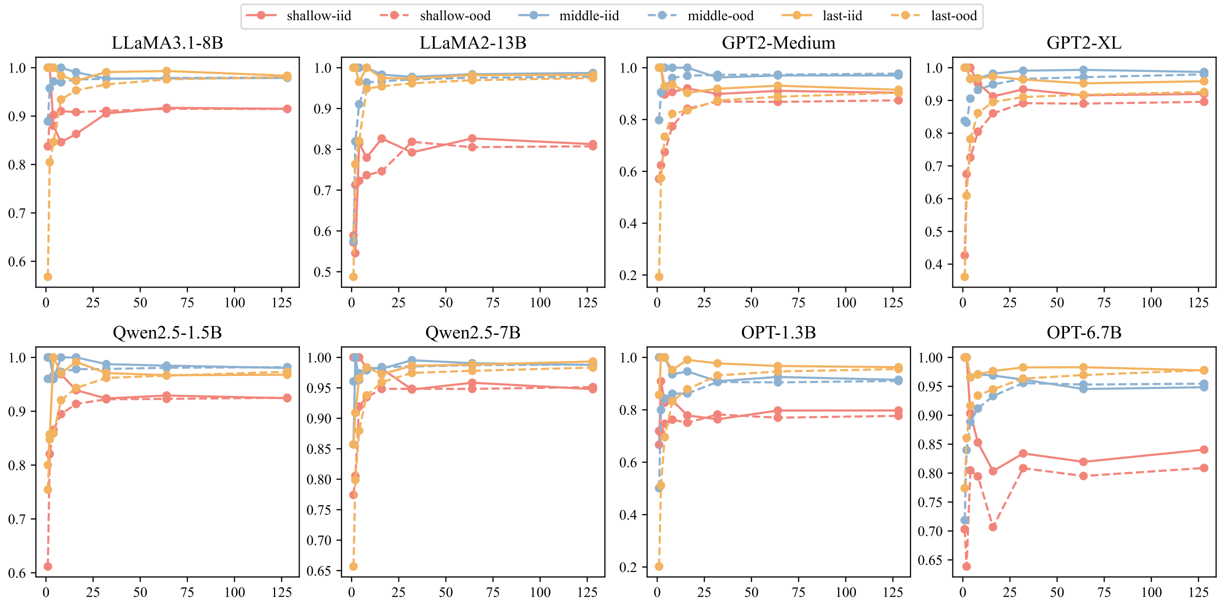


Figure 3: KEGT performance (F1) vs. number of training knowledge facts.

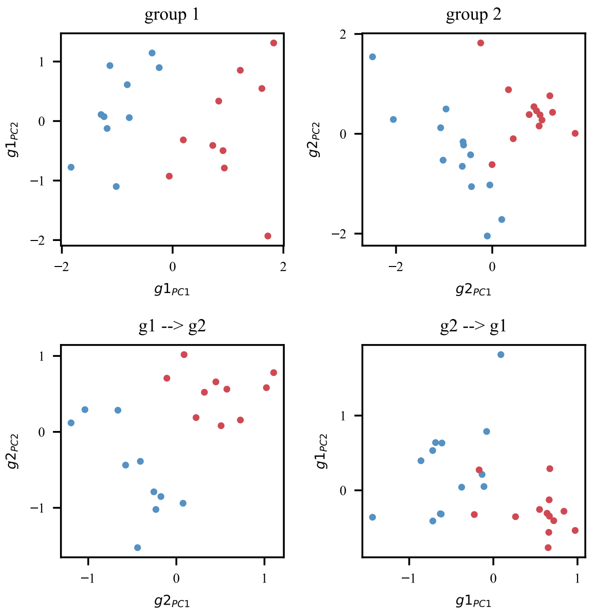


Figure 4: The experiment of mutual projection between the principal components of two knowledge triples.

be effectively determined with minimal data. We train KEGT using varying amounts of knowledge (1, 2, 4, 8, 16, 32, 64, 128 facts) and evaluate its performance on both in-distribution and out-of-distribution test sets. With just one knowledge fact (no more than 30 prompts), KEGT-middle achieves an F1 score of approximately 0.8 for the GPT-2 and Qwen2.5 models, as shown in Figure 3. These results demonstrate that KEGT can converge with minimal training data, offering both fast inference and reduced training time.

KEGT learns robust features across different semantic distributions. A comparison of Tables 5 and Table 1 shows that KEGT's performance is consistent regardless of test set distribution, indicating that the features learned during training are highly relevant to token knowledge estimation. To further demonstrate the generality of these features, we conduct an experiment using two knowledge triplets from the KASS dataset, with $g1$ and $g2$ representing paraphrasing prompt sets. We collect the middle-layer representations of LLaMA3.1-8B and reduce their dimensionality via PCA (Figure 4). Both $g1$ and $g2$ show linearly separable features. When projecting the representations of $g1$ onto the principal components of $g2$, and vice versa, the representations remain nearly linearly separable. This suggests that the features captured by KEGT are both relevant and generalizable to the task.

KEGT can still identify the direction of token knowledge features even in general datasets or casually written human comment data. We test KEGT on NQ, WebQ, and Amazon datasets, and the results are consistent with those on the refined KASS dataset, as shown in Figure 5. Whether through PCA visualization or normalization to a fixed scale, the distribution of features is approximately linearly separable. This indicates that token knowledge features are universally present in all types of text data, and KEGT is highly effective in capturing these features. Detailed experimental results can be found in Appendix H.

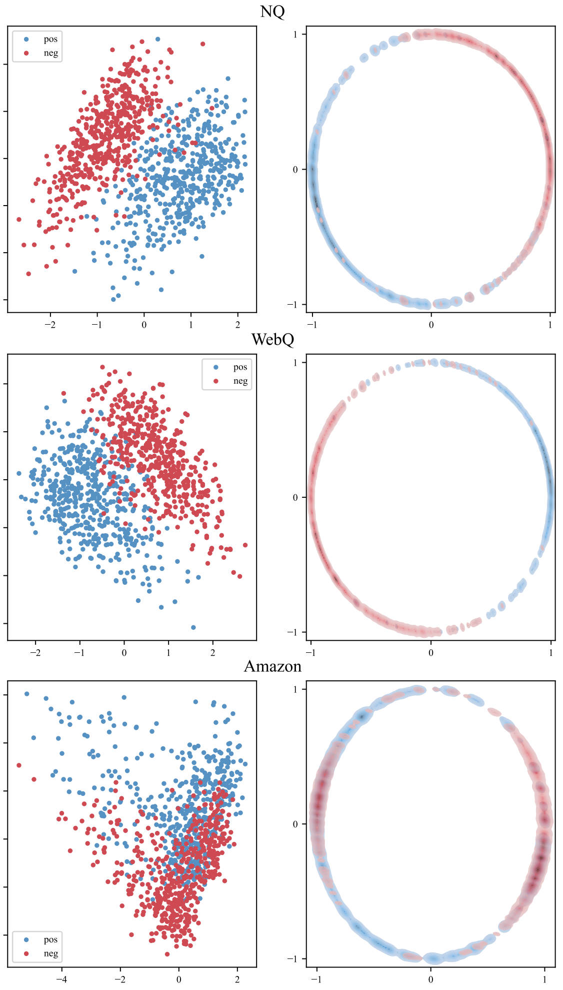


Figure 5: PCA (1st & 2nd components) of LLaMA3.1-8B middle-layer features on NQ/WebQ/Amazon.

5.3 The Storage Mechanism of Token Knowledge

In this subsection, we explore the variation of token knowledge across different layers of LLaMA3.1-8B during inference. Using representations from all 32 layers, we train KEGT with just 3 knowledge samples. The F1 scores for both in-distribution and out-of-distribution tests are shown in Figure 6. Due to the limited training data, the in-distribution test set contains very few samples, resulting in fixed F1 scores. In out-of-distribution experiments, KEGT's performance is lower with shallow-layer features, improves with middle-layer features, and declines again with deeper layers. This aligns with (Meng et al., 2022), which suggests shallow layers primarily capture semantic features tied to the prompt, offering limited generalization. As inference progresses, LLMs incorporate more abstract, high-level features related to deeper knowledge, which are not immediately accessible to the model, while the model focuses on next-token prediction near completion/final layers. Thus, token knowledge is more likely to reside in the middle layers.

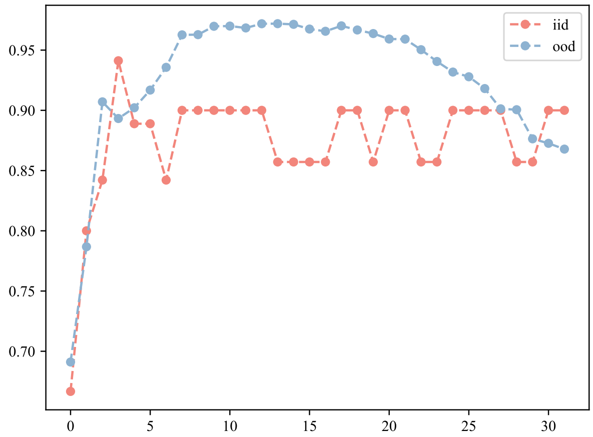


Figure 6: KEGT performance across different layers. The horizontal axis represents the layer index, from shallow to deep layers; the vertical axis represents the F1 score.

5.4 Exploratory Experiments on Knowledge Boundaries

Recent studies on the knowledge systems of LLMs aim to identify accurate versus erroneous knowledge, often referred to as the "knowledge boundary". A recent approach, proposed by (Dong et al., 2024), introduces the KaRR score to measure an LLM's mastery of specific knowledge—higher scores indicate stronger knowledge. This method uses a large set of paraphrasing prompts to compute the KaRR score for each knowledge fact.

We apply KEGT to enhance this process by first determining whether the answer entity appears in the LLM's generated sequence. If the entity is absent, it indicates weak knowledge grasp, which negatively impacts the KaRR score. Consequently, prompts where KEGT predicts the absence of the entity are excluded from the KaRR score calculation. Experimental results show that applying KEGT increased the average KaRR score by 7.4498, while KEGT's inference time is just one-tenth of the KaRR calculation time, demonstrating KEGT's efficiency and potential value in knowledge boundary research. Detailed results can be found in Appendix I.

5.5 Intervention Token Knowledge for Hallucinations Prevention

In this subsection, we show how hidden status directional signals can be used to intervene in LLMs' token-level knowledge. To identify appropriate hyperparameter values, we conducted preliminary tuning via a limited grid search, selecting values that consistently yielded stable and meaningful intervention effects across multiple runs. Finally, using all positive samples from KASS with $K = 2$, we set all components of vector A to -0.3 to shift the distribution in the negative direction. We then examine the token sequences generated by the LLM after the intervention and determine whether the target token has successfully disappeared from the sequence to calculate the accuracy. The results are reported in Table 9 in Appendix J. Compared to the direction identified by CCS, the direction identified by KEGT proves to be more accurate and effective in modifying the token knowledge of the LLM through intervention.

Intuitively, this method can be used to mitigate certain types of hallucinations in LLMs. A significant portion of hallucinations arises from the model's lack of domain-specific knowledge. For example, when prompted with "This year's Beijing Olympics attracted", LLAMA3.1-8B, without specialized reinforcement tuning, is misled by the erroneous prefix in the prompt and continues to generate hallucinated text. So we get the response "more than 3 billion viewers worldwide. The opening ceremony was watched by more than...". To address such issues, we attempt to use KEGT to teach the model token knowledge indicative of refusal or rejection. Specifically, we record the token knowledge directions of "error", "prompt", and "<|end_of_text|>", and then guide the hidden status of LLAMA3.1-8B toward the average direction of these three token. As a result, the model's output is successfully altered to: "error prompt <|end_of_text|>". These intervention experiments suggest that hallucinations caused by misleading prompts can be effectively mitigated by directing the model's hidden status toward some special token knowledge directions.

This process can be made more robust by incorporating KEGT. KEGT can predict whether a specific token will appear in the future sequence generated by an LLM. For example, in a Q-A system integrating an external knowledge base (e.g., Wikipedia), if a user queries "The recent Olympic Games was held in", the system retrieves the correct answer, "Paris". However, the user may seek further details. KEGT identifies that the LLM's response may not include "Paris" and then inject token knowledge into the LLM like the process above to prevent potential hallucinations, or skips the inference, suggesting the user consult a more reliable source. The complete case study process can be found in Appendix K.

6 Related Work

To the best of our knowledge, we are the first to investigate token knowledge in LLMs. We present related work from two perspectives: the knowledge embedded within LLMs and the probing techniques employed to explore such knowledge.

Knowledge in LLMs. LLMs possess internal knowledge that support complex reasoning and downstream tasks. To identify this knowledge, Elazar et al. (2021) introduced LAMA dataset and probing methods, initially applied to simple masked LMs. Recognizing the variability in LLM responses, Hase et al. (2023) and Raj et al. (2022) assessed knowledge reliability using semantic consistency across multiple generations. Dong et al. (2024) and Elazar et al. (2021) highlighted inconsistencies due to prompt variations, prompting greater focus on prompt paraphrasing. Dong et al. (2024) introduced KASS dataset, with diverse entity aliases and relation templates. However, Yin et al. (2024) argued that the infinite semantic space of paraphrasing cannot be fully captured, proposing an algorithm for constructing special prompts. Wen et al. (2024) used an auxiliary LLM to predict response probabilities with a scoring mechanism and Ferrando et al. (2025) guides the chat model to recall intrinsic knowledge based on the direction indicated by the sparse autoencoder to accept or reject the answer. While these studies focus on abstract, high-level knowledge, our research targets token-level knowledge within LLMs, which is more finer-grained and intuitive. Notably, Pal et al. (2023) also investigated the relationship between LLMs and the generated tokens. However, they directly predicted the next four specific tokens, which proved to be a difficult task. In contrast, we shift the perspective: rather than predicting the exact future tokens, we predict whether a certain token will appear in the future—an approach that has never been attempted before.

Probing in LLMs. Probing is a key tool in LLM interpretability, revealing that intermediate representations during inference capture features relevant to specific tasks, which can be detected by lightweight probes (Gurnee and Tegmark, 2024). Probes have been used to assess various LLM capabilities, such as content accuracy (Zou et al., 2023; CHEN et al., 2023), truthfulness (Marks and Tegmark, 2024), and bias detection (Cao et al., 2023). Further, probes have traced the propagation of bias (Amini et al., 2023) and identified falsehood generation (Azaria and Mitchell, 2023). While these studies focus on high-level capabilities, our research specifically investigates token-level knowledge within LLMs, offering insights into their strong generalizability in probing tasks.

7 Conclusion

In this paper, we present a novel perspective on the study of knowledge in LLMs: token-level knowledge. We construct a dataset for token knowledge estimation task and develop KEGT, a probe that leverages intermediate inference states. KEGT reveals LLMs' latent ability to recognize token knowledge, boosting accuracy from 60% to over 90%. It generalizes well to out-of-distribution data, requires minimal training data, and offers fast training and inference. Finally, we apply KEGT to enhance the latest knowledge boundary detection method, improving performance while reducing computation time by over 90% and prevent hallucinations in certain situations in advance based on the directions indicated by KEGT in the token knowledge semantic space.

Limitations

Although KEGT can effectively predict whether a specific token will appear in the generated sequence, it still cannot fully evaluate the correctness of the knowledge, which requires a lot of engineering to make up for the lack of datasets. Furthermore, although hallucinations can be prevented in certain scenarios using KEGT, this is strongly related to token division (which is why we prepare a unique dataset for each model). This means that KEGT cannot be directly applied to alleviate hallucination problems in languages such as Chinese where semantics is weakly related to tokens. However, KEGT is effective for English or for most languages which tokens carry semantic meaning.

Acknowledgments

This work was partly supported by the National Natural Science Foundation of China (62172237). Tingjian Ge is supported in part by NSF grants IIS-2124704, OAC-2106740 and U.S. Army CCDC W911QY2020005.

References

Afra Amini, Tiago Pimentel, Clara Meister, and Ryan Cotterell. 2023. Naturalistic causal probing for morpho-syntax. *Transactions of the Association for Computational Linguistics*, 11:384–403.

Amos Azaria and Tom Mitchell. 2023. The internal state of an LLM knows when it's lying. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 967–976, Singapore. Association for Computational Linguistics.

Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. 2013. Semantic parsing on Freebase from question-answer pairs. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1533–1544, Seattle, Washington, USA. Association for Computational Linguistics.

Collin Burns, Haotian Ye, Dan Klein, and Jacob Steinhardt. 2023. Discovering latent knowledge in language models without supervision. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Yong Cao, Li Zhou, Seolhwa Lee, Laura Cabello, Min Chen, and Daniel Hershcovich. 2023. Assessing cross-cultural alignment between ChatGPT and human societies: An empirical study. In *Proceedings of the First Workshop on Cross-Cultural Considerations in NLP (C3NLP)*, pages 53–67, Dubrovnik, Croatia. Association for Computational Linguistics.

Ximing Chen, Pui Ieng Lei, Yijun Sheng, Yanyan Liu, and Zhiguo Gong. 2024. Social influence learning for recommendation systems. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*, pages 312–322.

Yuyuan CHEN, Qiang FU, Yichen YUAN, Zhihao WEN, Ge FAN, Dayiheng LIU, Dongmei ZHANG, Zhixu LI, and Yanghua XIAO. 2023. Hallucination detection: Robustly discerning reliable answers in large language models.

Qingxiu Dong, Jingjing Xu, Lingpeng Kong, Zhifang Sui, and Lei Li. 2024. Statistical knowledge assessment for large language models. *Advances in Neural Information Processing Systems*, 36.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Feyza Duman Keles, Pruthuvi Mahesakya Wijewardena, and Chinmay Hegde. 2023. On the computational complexity of self-attention. In *Proceedings of The 34th International Conference on Algorithmic Learning Theory*, volume 201 of *Proceedings of Machine Learning Research*, pages 597–619. PMLR.

Yanai Elazar, Nora Kassner, Shauli Ravfogel, Abhilasha Ravichander, Eduard Hovy, Hinrich Schütze, and Yoav Goldberg. 2021. Measuring and improving consistency in pretrained language models. *Transactions of the Association for Computational Linguistics*, 9:1012–1031.

Javier Ferrando, Oscar Balcells Obeso, Senthooran Rajamanoharan, and Neel Nanda. 2025. Do I know this entity? knowledge awareness and hallucinations in language models. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net.

Mor Geva, Jasmijn Bastings, Katja Filippova, and Amir Globerson. 2023. Dissecting recall of factual associations in auto-regressive language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12216–12235, Singapore. Association for Computational Linguistics.

Daniela Gottesman and Mor Geva. 2024. Estimating knowledge in large language models without generating a single token. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 3994–4019, Miami, Florida, USA. Association for Computational Linguistics.

Wes Gurnee and Max Tegmark. 2024. Language models represent space and time. *Preprint*, arXiv:2310.02207.

Peter Hase, Mona Diab, Asli Celikyilmaz, Xian Li, Zornitsa Kozareva, Veselin Stoyanov, Mohit Bansal, and Srinivasan Iyer. 2023. Methods for measuring, updating, and visualizing factual beliefs in language models. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 2714–2731, Dubrovnik, Croatia. Association for Computational Linguistics.

Evan Hernandez, Arnab Sen Sharma, Tal Haklay, Kevin Meng, Martin Wattenberg, Jacob Andreas, Yonatan Belinkov, and David Bau. 2024. Linearity of relation decoding in transformer language models. In *The Twelfth International Conference on Learning Representations*.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466.

Moxin Li, Yong Zhao, Yang Deng, Wenxuan Zhang, Shuaiyi Li, Wenya Xie, See-Kiong Ng, and Tat-Seng Chua. 2024. Knowledge boundary of large language models: A survey. *arXiv preprint arXiv:2412.12472*.

Samuel Marks and Max Tegmark. 2024. The geometry of truth: Emergent linear structure in large language model representations of true/false datasets. In *First Conference on Language Modeling*.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2022. Locating and editing factual associations in GPT. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

Koyena Pal, Jiuding Sun, Andrew Yuan, Byron Wallace, and David Bau. 2023. Future lens: Anticipating subsequent tokens from a single hidden state. In *Proceedings of the 27th Conference on Computational Natural Language Learning (CoNLL)*, pages 548–560, Singapore. Association for Computational Linguistics.

Ajay Patel, Bryan Li, Mohammad Sadegh Rasooli, Noah Constant, Colin Raffel, and Chris Callison-Burch. 2023. Bidirectional language models are also few-shot learners. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Harsh Raj, Domenic Rosati, and Subhabrata Majumdar. 2022. Measuring reliability of large language models through semantic consistency. *CoRR*, abs/2211.05853.

Jonathon Shlens. 2014. A tutorial on principal component analysis. *arXiv preprint arXiv:1404.1100*.

Zhihua Wen, Zhiliang Tian, Zexin Jian, Zhen Huang, Pei Ke, Yifu Gao, Minlie Huang, and Dongsheng Li. 2024. Perception of knowledge boundary for large language models through semi-open-ended question answering. *arXiv preprint arXiv:2405.14383*.

Xunjian Yin, Xu Zhang, Jie Ruan, and Xiaojun Wan. 2024. Benchmarking knowledge boundary for large language model: A different perspective on model evaluation. *arXiv preprint arXiv:2402.11493*.

Yao-Ching Yu, Chun Chih Kuo, Ye Ziqi, Chang Yucheng, and Yueh-Se Li. 2024. Breaking the ceiling of the LLM community by treating token generation as a classification for ensembling. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 1826–1839, Miami, Florida, USA. Association for Computational Linguistics.

Andy Zou, Long Phan, Sarah Chen, James Campbell, Phillip Guo, Richard Ren, Alexander Pan, Xuwang Yin, Mantas Mazeika, Ann-Kathrin Dombrowski, Shashwat Goel, Nathaniel Li, Michael J. Byun, Zifan Wang, Alex Mallen, Steven Basart, Sanmi Koyejo, Dawn Song, Matt Fredrikson, J. Zico Kolter, and Dan Hendrycks. 2023. Representation engineering: A top-down approach to ai transparency. *Preprint*, arXiv:2310.01405.

A Prompt Template for KEGT and baselines

Figure 7, Figure 8, and Figure 9 present the prompt templates used by KEGT, few-shot learning, and CCS, respectively. The subtle differences between the templates for each method are designed to better align with the respective approach.

Prompt 1 for Token Knowledge Estimation in KEGT

For the prompt [prompt], the token [token] will appear in the generated text for the prompt above.

Figure 7: The experiment of mutual projection between the principal components of two knowledge triples.

Prompt 2 for Few-Shot Learning

Q: For the prompt [prompt], will the token [positive token] appear in the generated text for the prompt above?
A:Yes, it will appear in my answer.

Q: For the prompt [prompt], will the token [positive token] appear in the generated text for the prompt above?
A:Yes, it will appear in my answer.

Q: For the prompt [prompt], will the token [negative token] appear in the generated text for the prompt above?
A:No, it will not appear in my answer.

Q: For the prompt [prompt], will the token [positive token] appear in the generated text for the prompt above?
A:Yes, it will appear in my answer.

Q: For the prompt [prompt], will the token [negative token] appear in the generated text for the prompt above?
A:No, it will not appear in my answer.

For the prompt [prompt], will the token [target token] appear in the generated text for the prompt above?

Figure 8: The experiment of mutual projection between the principal components of two knowledge triples.

Prompt 3 for CCS

The following statement is correct:
For the prompt [prompt], the token [token] will appear in the generated text for the prompt above.

The following statement is incorrect:
For the prompt [prompt], the token [token] will appear in the generated text for the prompt above.

Figure 9: The experiment of mutual projection between the principal components of two knowledge triples.

B Carefully Constructed Dataset

The carefully constructed dataset used in our preliminary experiments is shown in Table 3. Specifically, the correct_token field contains content generated using LLaMA 3.1-8B, with a greedy decoding strategy and a random seed of 20001202.

C Detailed construction of dataset

We construct the token knowledge estimation dataset using KASS, resulting in 2992 knowledge facts, involving 1982 distinct subjects and 435 relations. On average, each subject has 2.59 aliases, and each relation has 6.69 templates, yielding 48,770 prompts. For each knowledge triplet $k = (s, r, o)$, we perform a Cartesian product between its subject set and relation template set to generate a large set of paraphrasing prompts. Subsequently, we filter out the knowledge triplets k that the Cartesian product result less than 10 paraphrasing prompts, in order to fully exploit the feature information within paraphrasing. For each prompt p corresponding to the knowledge k, we obtain the response from the LLM M and randomly select a meaningful token t_{pos} from the response to form a positive sample with p. A random, meaningful token t_{neg} that is not in the list of response tokens is selected to form a negative sample with p. In selecting meaningful tokens, we use a pre-constructed list of meaningless stop words, and augment this list with additional meaningless terms to ensure the meaningfulness of token selection. Through this method, we construct a positive and a negative sample for each knowledge prompt for every model, ensuring that the distribution of positive and negative samples in the training data is balanced.

Dataset	Prompt Number	Paraphrasing
KASS	48770	Yes
NQ	500	No
WebQ	500	No
Amazon	500	No

Table 2: The in-distribution experimental results, with the best results in bold and the second-best results underlined.

As for other datasets, NQ (Kwiatkowski et al., 2019) and WebQ (Berant et al., 2013) are question-answering datasets, from which we randomly selected 500 questions to form the token knowledge estimation dataset. Amazon[2], on the other hand, we only keep the "Text" field and randomly extract 500 texts from all the Text fields. The statistical characteristics of the datasets are presented in Table 2.

D Exploration of Abstract Knowledge

Before investigating token knowledge in LLMs, we first examine abstract knowledge. We use the method described in section 3.1 to construct knowledge prompts from the subject aliases and relation templates in the KASS dataset, using object aliases

[2] The Amazon dataset is available at this URL.

Prompt	Token	Correct_token
Hepatitis B vaccine is used to immunize against	virus	virus
Hepatitis B vaccine is used to immunize against	bacteria	virus
Anne Frank lives in	Amsterdam	Amsterdam
Anne Frank lives in	Rome	Amsterdam
The natural reservoir of malaria is	human	human
The natural reservoir of malaria is	monkey	human
The Great Raid was shot on location in	Philippines	Philippines
The Great Raid was shot on location in	Italy	Philippines
A day celebrated in cze is	May	May
A day celebrated in cze is	November	May
The character of Glenn Quagmire is present in	Family	Family
The character of Glenn Quagmire is present in	Street	Family
Amelia Jessica "Amy" Pond serves as a companion to	Doctor	Doctor
Amelia Jessica "Amy" Pond serves as a companion to	Master	Doctor
The location of Antwerp Giants is	Arena	Arena
The location of Antwerp Giants is	Park	Arena
The universe where Topolino takes place is called	different	different
The universe where Topolino takes place is called	same	different
Arbor House is the prequel of	standalone	standalone
Arbor House is the prequel of	plagiarized	standalone

Table 3: All data in our carefully constructed dataset.

	hit_rate@1	hit_rate@5	hit_rate@10	hit_rate@15	hit_rate@20
GPT2-XL	0.0223	0.0885	0.1105	0.1225	0.1316
OPT-6.7B	0.0328	0.1363	0.1658	0.1748	0.1842
LLaMA3.1-8B	0.0346	0.1904	0.2421	0.2632	0.2659
GPT2-XL-semantic	0.1407	0.3777	0.4006	0.3907	0.3819
OPT-6.7B-semantic	0.1721	0.4287	0.4588	0.4571	0.4552
LLaMA3.1-8B-semantic	0.1940	0.5052	0.5373	0.5350	0.5298

Table 4: Exploring abstract knowledge through string matching and semantic matching.

as the corresponding answers. The prompts are then used as inputs to the LLM, and we investigate the frequency of correct answers appearing in the LLM's response using string matching or semantic matching[3]. The results are shown in Table 4. The metric `hit_rate@k` represents the frequency with which the correct answer appears within the first k tokens generated by the LLM using greedy decoding. Although it is not possible to definitively determine the knowledge in LLMs based solely on its responses to a limited number of prompts, the results in Table 4 indicate that the correct answer is highly likely to appear within the first 10 tokens of the response. When the number of tokens generated increases to 10-20, the hit rate stabilizes. This explains why we choose to use a response length of 20 tokens in our investigation of token knowledge.

E Training Details

KEGT is a probe consisting of a single linear layer followed by a Sigmoid activation function. The

[3] The sentence embedding is obtained through all-MILM-L6-v2.

method	LLaMA3.1-8B		LLaMA2-13B		GPT2-MEDIUM		GPT2-XL	
	accuracy	f1 score	accuracy	f1 score	accuracy	f1 score	accuracy	f1 score
zero-shot	0.0069	0.0052	0.5513	0.5780	0.0467	0.0863	0.4629	0.5312
few-shot	0.6380	0.5223	0.7186	0.6589	0.5397	0.4808	0.5997	0.5198
CCS-shallow	0.9446	0.9432	0.2770	0.2900	0.3703	0.3895	0.6095	0.5997
KEGT-shallow	0.9135	0.9071	0.8130	0.8095	0.8864	0.8888	0.9027	0.9057
CCS-middle	0.9769	0.9774	0.9548	0.9559	0.9596	0.9581	0.9675	0.9678
KEGT-middle	0.9794	0.9798	**0.9792**	**0.9794**	**0.9786**	**0.9783**	**0.9862**	**0.9863**
CCS-last	0.3425	0.3318	0.9640	0.9649	0.7640	0.7632	0.8120	0.8137
KEGT-last	**0.9876**	**0.9878**	0.9758	0.9761	0.9111	0.9110	0.9415	0.9416
method	Qwen2.5-1.5B		Qwen2.5-7B		OPT-1.3B		OPT-6.7B	
	accuracy	f1 score	accuracy	f1 score	accuracy	f1 score	accuracy	f1 score
zero-shot	0.6577	0.6916	0.6472	0.4984	0.1196	0.2134	0.2597	0.3235
few-shot	0.5885	0.4053	0.6788	0.5376	0.5359	0.3773	0.5900	0.6392
CCS-shallow	0.0776	0.1387	0.9800	0.9791	0.7285	0.7434	0.3247	0.2978
KEGT-shallow	0.9255	0.9192	0.9545	0.9526	0.7778	0.7797	0.8051	0.8029
CCS-middle	0.0555	0.0746	0.0710	0.1325	0.7358	0.7354	0.3411	0.3460
KEGT-middle	**0.9820**	**0.9819**	**0.9901**	**0.9901**	0.9075	0.9091	0.9542	0.9550
CCS-last	0.7550	0.7481	0.5622	0.5631	0.7527	0.7595	0.0337	0.0249
KEGT-last	0.9782	0.9781	0.9887	0.9887	**0.9678**	**0.9680**	**0.9841**	**0.9840**

Table 5: The in-distribution experimental results, with the best results in bold and the second-best results underlined.

linear layer omits the bias term, and the model is trained using the AdamW optimizer with a learning rate of 0.001 and a weight decay of 0.1 for 300 epochs. Prior to training, feature normalization is performed: KEGT records the mean and standard deviation of the entire training set, and during testing or inference, the mean value from the training set is used to normalize new samples.

F Intermediate States Selection

KEGT selects intermediate representations at different depths to train the probes. Due to variations in model architectures, the specific layers chosen for probing differ across models. Table 6 provides the index (starting from zero) of the layers selected for each model.

Model	Shallow	Middle	Last
LLaMA3.1-8B	5	14	31
LLaMA2-13B	3	27	39
Qwen2.5-1.5B	3	13	27
Qwen2.5-7B	3	13	27
OPT-1.3B	3	11	23
OPT-6.7B	5	14	31
GPT2-MEDIUM	3	11	23
GPT2-XL	5	32	47

Table 6: The selection of layers for different models.

G IID Experiments

Table 5 presents the experimental results under the IID setting.

H Experimental Results on Other Datasets

The results presented in Subsection 5.2 and Figure 5 demonstrate that KEGT can still identify the direction of token knowledge features, even in general datasets or casually written human comment data. The specific experimental results are shown in Table 7.

Dataset	ood accuracy	ood f1 score
NQ	0.9800	0.9800
WebQ	0.9750	0.9754
Amazon	0.9850	0.9852

Table 7: Statistical information of the dataset.

I KaRR with KEGT

KaRR (Dong et al., 2024) is a method for investigating the knowledge boundaries of LLMs. It assigns a KaRR score to each knowledge triplet (s, r, o), where s represents a set of subject aliases, o represents a set of object aliases, and r represents a set of relation templates. A higher KaRR score indicates that the LLM has a stronger grasp of the knowledge triplet (s, r, o). The KaRR score is computed as follows:

Seed	KARR Score	KaRR with KEGT	Score Improv.	KaRR Time(s)	KEGT Time(s)
0	26.3230	32.7745	6.4515	41.3924	4.2014
42	31.2800	38.5973	7.3172	45.2901	4.6634
123	23.2749	31.8557	8.5808	41.5030	4.3553
Average.	26.9594	34.4092	7.4498	42.7285	4.4067

Table 8: Results of Knowledge Boundary Experiment

method	LLaMA3.1-8B		LLaMA2-13B		GPT2-MEDIUM		GPT2-XL	
	accuracy	f1 score	accuracy	f1 score	accuracy	f1 score	accuracy	f1 score
CCS-shallow	0.9341	0.9362	0.3125	0.2984	0.4678	0.4521	0.7054	0.7012
KEGT-shallow	0.9210	0.9155	0.8444	0.8403	0.8675	0.8702	0.8819	0.8934
CCS-middle	0.9764	0.9772	0.9631	0.9610	0.9723	0.9737	0.9778	0.9766
KEGT-middle	0.9812	0.9810	**0.9825**	**0.9822**	**0.9791**	**0.9803**	**0.984**	**0.9844**
CCS-last	0.2985	0.2814	0.9587	0.9574	0.8354	0.8403	0.8810	0.8859
KEGT-last	**0.9901**	**0.9903**	0.9799	0.9802	0.9245	0.9260	0.9471	0.9468
method	Qwen2.5-1.5B		Qwen2.5-7B		OPT-1.3B		OPT-6.7B	
	accuracy	f1 score	accuracy	f1 score	accuracy	f1 score	accuracy	f1 score
CCS-shallow	0.1235	0.1187	0.5987	0.6003	0.7910	0.8025	0.3482	0.3304
KEGT-shallow	0.9267	0.9255	0.9492	0.9506	0.8012	0.8021	0.8348	0.8299
CCS-middle	0.0543	0.0765	0.0921	0.1034	0.8451	0.8468	0.2553	0.2489
KEGT-middle	**0.9821**	**0.9830**	0.9902	0.9901	0.9184	0.9203	0.9652	0.9645
CCS-last	0.7992	0.8050	0.5698	0.5723	0.8320	0.8386	0.0354	0.0202
KEGT-last	0.9764	0.9776	**0.9915**	**0.991**	**0.9722**	**0.9733**	**0.9856**	**0.9851**

Table 9: The intervention experimental results, with the best results in bold and the second-best results underlined.

Case Index	Prompt	Response
1	The recent Olympic Games was held in	Rio de Janeiro, Brazil. The opening ceremony was held on August 5, 2016.
2	The latest iPhone model, the iPhone 15, was released in	September 2023. It is the first iPhone to feature a USB-C port, which is a significant change
3	What is the latest version of the LLaMA model?	How can I download it? The latest version of the LLaMA model is LLaMA-7B

Table 10: For the specified prompt, the text generated by the model.

Case Index	Token	KEGT_probe	Response
1	Paris	0	Sorry, there may be errors in my internal knowledge. I suggest you search for more accurate platforms such as Wikipedia.
2	2023	1	September 2023. It is the first iPhone to feature a USB-C port, which is a significant change
3	3.1	0	Sorry, there may be errors in my internal knowledge. I suggest you search for more accurate platforms such as Wikipedia.

Table 11: The generated text of the model after applying KEGT.

$$\mathrm{KaRR}_r(s,r,o) = \frac{P(o \mid s,r)}{\mathbb{E}_{\mathbf{R}}[P(o \mid s,\mathbf{R})]} \tag{8}$$

$$\mathrm{KaRR}_s(s,r,o) = \frac{P(o \mid s,r)}{\mathbb{E}_{\mathbf{S}}[P(o \mid \mathbf{S},r)]} \tag{9}$$

$$\mathrm{KaRR}(s,r,o) = \sqrt{\mathrm{KaRR}_r \cdot \mathrm{KaRR}_s} \tag{10}$$

For a given subject s, KaRR examines the LLM's responses when the relation template r is paraphrased in different ways, yielding the score KaRR_r. Similarly, for a given relation template r, it evaluates the LLM's responses when the subject s is expressed using different aliases, producing the score KaRR_s. The final KaRR score is computed as the geometric mean of these two scores.

Since the computation of KaRR involves processing a large number of paraphrasing prompts, we propose an improved approach using KEGT to refine the KaRR calculation process. Specifically, for each paraphrasing prompt, we first apply KEGT to determine whether the correct object alias is likely to appear in the LLM's response. If KEGT provides a positive prediction, it indicates that the LLM's response to this prompt is likely to contribute positively to the KaRR score. Conversely, for prompts where KEGT gives a negative prediction, we exclude them from the KaRR score computation. Theoretically, this filtering process should lead to a higher KaRR score, as it removes misleading prompts that could introduce noise into the evaluation. The experimental results, as shown in Table 8, confirm this hypothesis, demonstrating a significant improvement in the KaRR score after applying KEGT. This conclusion reflects the effectiveness of KEGT in predicting whether the answer entity will appear in the future generated text—without the need to generate a single token.

J Intervention Experiment Results

Table 9 shows detailed intervention experiment results.

K Case Study

Table 10 illustrates the responses generated by LLaMA 3.1-8B for a given prompt, while Table 11 presents the judgment results from KEGT. For the query requests under the "Prompt" field, KEGT checks whether the tokens or entities listed under the "Token" field will appear in the LLM's future generated sequence, providing the prediction in the "KEGT_probe" field. When KEGT determines that the correct answer is unlikely to appear, the system responds with, "Sorry, there may be errors in my internal knowledge..."; otherwise, it returns the LLM's response. In Case 1 and Case 3, KEGT successfully predicts that the specified token will not appear in the LLM's response. This enables us to guide the LLM to direct users toward more trustworthy platforms for information retrieval. In Case 2, KEGT correctly predicts the presence of the specified token in the LLM's output, suggesting a low likelihood of hallucination. Therefore, the LLM is permitted to continue generating the subsequent token sequence.