Popular but Wrong: Understanding and Calibrating LLM Overconfidence through Knowledge Popularity

Anonymous ACL submission

Abstract

Large language models (LLMs) often produce incorrect answers with high confidence, yet the factors associated with such overconfidence remain insufficiently understood. In this paper, we study this problem through the lens of knowledge popularity. Using entity-centric factual QA derived from Wikidata triplets, we decompose popularity into three signals: question entity popularity, generated answer popularity, and question-generated answer co-occurrence. Our analysis reveals two consistent patterns. First, when LLMs make mistakes, their hallucinated answers are far from random: they tend to be more popular or more strongly associated with the question entity than the ground-truth answer. Second, model confidence is strongly tied to popularity-related properties of the generated answer, especially question-answer co-occurrence, even when the answer is incorrect. Together, these findings suggest a popularity-associated pathway to overconfidence: LLMs often generate popular or strongly associated incorrect alternatives, which can receive disproportionately high confidence. We further show that knowledge popularity can be used for confidence calibration. Across six models and three datasets, the three popularity signals substantially reduce the average confidence on incorrect answers from 0.765 to 0.254, while improving overall calibration by reducing ECE from 0.356 to 0.050.

1 Introduction

A trustworthy model should be well calibrated: its confidence in an answer, measured by the generation probability assigned to that answer, should reflect the likelihood that the answer is correct. However, large language models (LLMs) have been shown to be overconfident, often assigning high confidence to incorrect answers (Jiang et al., 2021; Xiong et al., 2023; Ni et al., 2024a; Yang et al., 2023). Such miscalibrated confidence poses significant risks, especially in safety-critical domains such as healthcare and finance.

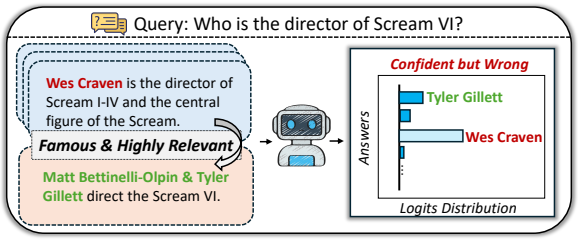

Figure 1: An example where the model generates an incorrect answer that is more popular than the correct answer, while assigning it high confidence.

Prior work has linked miscalibration to training objectives, suggesting that supervised fine-tuning and RLHF can make model predictions sharper and more confident (Desai and Durrett, 2020; Kadavath et al., 2022). Beyond training objectives, we argue that model confidence may also be affected by the knowledge it has encountered during training.

Knowledge popularity has been widely studied as an important factor in LLM factual recall. Prior work shows that LLMs are more likely to answer correctly when the relevant knowledge appears more frequently in the corpus, such as when the question entity is more popular or when the question entity and ground-truth answer co-occur more often (Mallen et al., 2023; Kandpal et al., 2023). However, these studies mainly focus on the ground-truth side, explaining when models are likely to know the correct answer. Overconfidence, in contrast, concerns the confidence assigned to the generated answer, especially when that answer is incorrect. This motivates us to shift attention from ground-truth-side popularity to generated-answer-side popularity signals. Specifically, we ask two questions: (1) When LLMs make mistakes, what kinds of answers do they generate? (2) How are generated-answer-side popularity signals associated with model confidence, especially on incorrect answers?

To answer these questions, we focus on factoid question answering (QA), which allows us to decompose generation-side popularity into question entity popularity, generated answer popularity, and question-answer co-occurrence. Specifically, we employ three representative datasets constructed by prior work (Yuksekgonul et al., 2023), covering multiple difficulty levels, with examples shown in Figure 3. Following prior studies (Mallen et al., 2023; Kandpal et al., 2023; Yuksekgonul et al., 2023), we do not directly observe models' pretraining data. Instead, we use external statistics as proxies, including Wikidata sitelinks for entity popularity and Wikipedia document co-occurrence for question-answer association strength.

We find that: 1) Hallucinated answers are far from random. When models make mistakes, they tend to generate entities that are more popular or more frequently co-occurring with the question entity than the ground-truth answer, as shown in Figure 1. This tendency becomes more pronounced in more challenging domains, suggesting that when models fail to recall the correct answer, they are more likely to generate popular or strongly associated alternatives. 2) Model confidence is consistently associated with properties of the generated answer, with question-answer co-occurrence emerging as the strongest signal in most settings, even among incorrect predictions. Notably, beyond question-answer co-occurrence, LLMs can also assign high confidence to incorrect answers simply because the generated answer itself is popular. *Together, these findings suggest a popularity-associated pathway to overconfidence: when LLMs fail to recall the correct answer, they are more likely to generate popular or strongly associated alternatives; because model confidence is also closely tied to these generated-answer-side popularity signals, these incorrect alternatives can receive disproportionately high confidence.* This pattern persists across model scales, suggesting that scaling alone may not eliminate popularity-associated overconfidence.

Although popular incorrect answers can receive high confidence, correct predictions typically exhibit even stronger and more reliable question-answer association patterns. This contrast suggests that popularity-related signals can not only help characterize spurious confidence, but also help distinguish reliable confidence from overconfidence. Based on this insight, we leverage knowledge popularity for confidence calibration. Experimental results show that among the three popularity signals, question-generated answer co-occurrence is the most effective in reducing overconfidence. Across six models and three datasets, it lowers the average confidence on incorrect answers from 0.765 to 0.265, and using all three signals further reduces it to 0.254. Moreover, on the full set of samples, the combined signals substantially improve calibration, reducing ECE from 0.356 to 0.050. To reduce reliance on external corpora, we further explore using the model itself to estimate popularity signals, which remains a reasonable alternative.

2 Related Work

Existing research on model knowledge boundary perception focuses on assessing model confidence and can be mainly classified into four categories.

Probabilistic Confidence. This line of research treats the generation probability of the answer as the confidence of the model (Guo et al., 2017; Desai and Durrett, 2020; Jiang et al., 2021; Kadavath et al., 2022; Si et al., 2022; Kuhn et al., 2023). Guo et al. (2017) examined early neural networks (e.g., ResNet (He et al., 2016)) and found them to be overconfident, proposing temperature scaling as a remedy. Later, Desai and Durrett (2020) showed that BERT-style models tend to be relatively well-calibrated, while Jiang et al. (2021) found that pre-trained language models such as T5 (Raffel et al., 2020) remained overconfident. More recent work has turned to LLMs, with studies showing that they, too, exhibit overconfidence (Si et al., 2022; Lin et al., 2022; Tian et al., 2023).

Verbalized Confidence. LLMs have been shown to express their confidence verbally (Lin et al., 2022; Yin et al., 2023; Tian et al., 2023; Xiong et al., 2023; Yang et al., 2023; Ni et al., 2024a). Some studies (Yin et al., 2023; Ni et al., 2024a) found that LLMs often fail to recognize their knowledge limitations verbally and tend to be overconfident. Xiong et al. (2023) systematically studied black-box approaches for estimating LLM confidence. Beyond prompting-based methods, some studies aim to train LLMs to verbalize more accurate confidence (Lin et al., 2022; Yang et al., 2023; Zhang et al., 2024).

Consistency-based Confidence. If the model is confident in its answer, it should maintain consistency across multiple generations. Recent studies

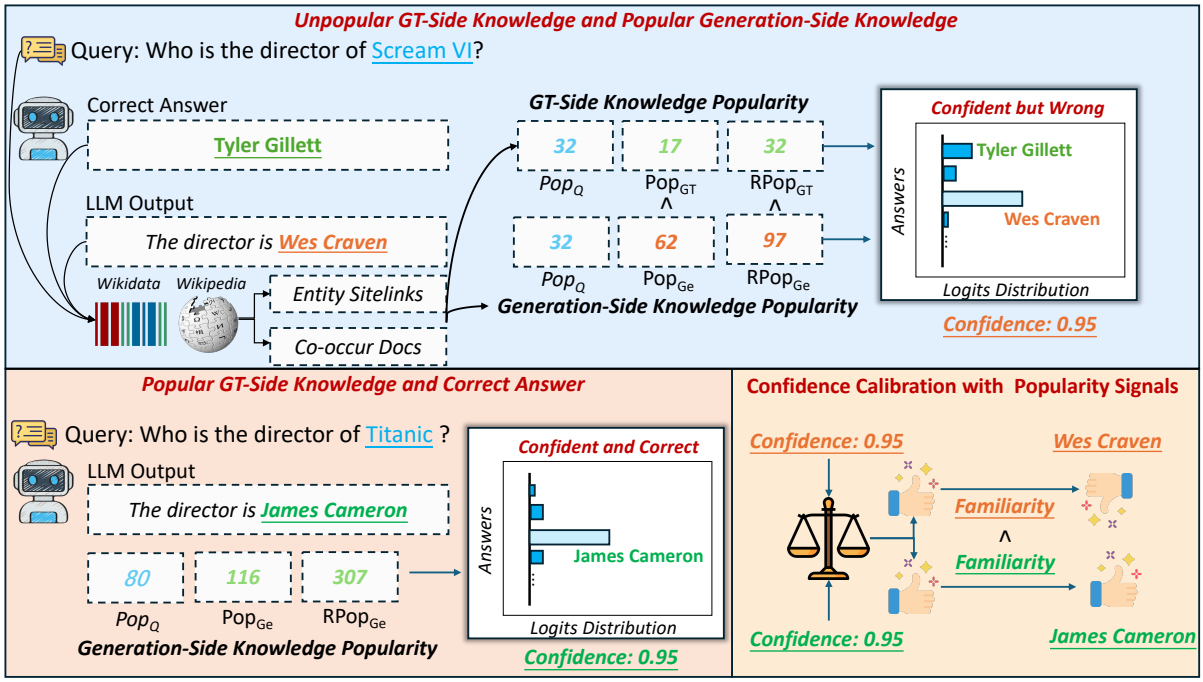

Figure 2: Overview of knowledge popularity, overconfidence, and calibration. The upper panel illustrates overconfident errors caused by popular or strongly associated incorrect answers; the lower panels show correct high-confidence predictions and popularity-aware confidence calibration.

have used self-consistency across generations as a proxy for LLM confidence (Manakul et al., 2023; Kuhn et al., 2023). Zhang et al. (2023) extended this by evaluating the consistency of answers across multiple semantically equivalent inputs and across different models. Ding et al. (2024) further adapted this approach to the multilingual setting.

Confidence Estimation via LLM Internal States. LLMs' internal states have shown to be effective in evaluating the factuality of their self-generated content (Su et al., 2024; Chen et al., 2024; Wang et al., 2024; Ni et al., 2025). Specifically, Su et al. (2024) and Chen et al. (2024) focused on internal states after generation, Wang et al. (2024) examined those before generation, and Ni et al. (2025) explored leveraging LLMs' internal states to enhance their perception of knowledge boundaries from efficiency and risk perspectives.

We focus on probabilistic confidence for the following reasons: 1) Both the model's generation probabilities and its knowledge acquisition arise from the same training objective, and are expected to align with each other. 2) Models without specialized training often struggle to verbalize confidence accurately (Ni et al., 2024b); consistency-based methods require multiple generations and incur high inference costs; and internal-state-based approaches require access to hidden representations and additional training. In contrast, probabilistic confidence is readily accessible and has been shown to perform well, especially when answers are short (Ding et al., 2024).

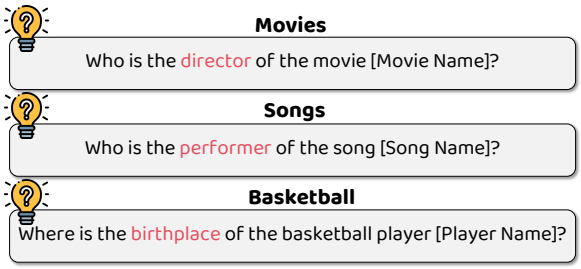

Figure 3: Question examples for each dataset.

3 Task Formulation

Entity-Centric QA. We focus on entity-centric knowledge because it allows us to measure knowledge popularity through entities. In entity-centric QA, questions and answers are derived from knowledge triples in the form of (subject, relation, object), where the question queries the relation of a given subject, and the model is expected to generate the corresponding object. Examples of knowledge triples are provided in Table 1, with their transformed question forms shown in Figure 3.

LLM Knowledge Boundary Perception. The model's perception of its knowledge boundaries is evaluated by the alignment between its confidence and actual QA performance. QA performance is measured by whether the generated answer contains the ground-truth answer, and confidence is reflected in generation probability of the answer tokens. Specifically, for a question q and a model M, the confidence c is computed as:

$$c = \frac{1}{T}\sum_{i=1}^{T} P(g_i|g_{<i}), \quad (1)$$

where $\{g_1, \ldots, g_T\}$ is the generated tokens.

Datasets	Count	Subject	Relation	Object
Movies	10,964	Movie	Directed by	Director
Songs	2,157	Song	Performed by	Performer
Basketball	13,309	Player	Birthplace	City

Table 1: Sample counts for each dataset, along with the corresponding subject, relation, and object types.

4 Experimental Setup

Datasets. Yuksekgonul et al. (2023) construct entity-centric QA datasets from Wikidata[1], using sitelinks as a proxy for entity popularity, which correlates strongly with QA performance. Following their setup, we use three datasets—Movies, Songs, and Basketball—to capture different levels of knowledge popularity. Specifically, question popularity decreases from Movies to Basketball, while ground-truth answer popularity shows the opposite trend. Table 1 summarizes the knowledge triplets and data statistics, and Figure 3 shows examples.

Entity Popularity. Following Mallen et al. (2023); Yuksekgonul et al. (2023), we define the popularity of an entity by the number of sitelinks it has—i.e., the number of Wikipedia pages in different languages that link to it.

Co-occurrence Count. As Wikipedia is the primary high-quality source for Wikidata, we estimate relation popularity based on Wikipedia content. Specifically, for each entity pair, we measure relation popularity by counting the number of documents in the Wikipedia dump[2] in which both entities co-occur.

[1] https://query.wikidata.org/sparql

[2] https://huggingface.co/datasets/wikimedia/wikipedia

Notation	Definition
Pop_{Q}	Popularity of entities in the question
Pop_{GT}	Popularity of entities in the ground-truth answer
RPop_{GT}	Co-occr. count between question and ground-truth entities
Pop_{Ge}	Popularity of entities in the generated answer
RPop_{Ge}	Co-occr. count between question and generated entities

Table 2: Definitions of notations about knowledge popularity where co-occur. means co-occurrence.

LLMs. We conduct experiments on six representative LLMs: five open-source models, Llama3-8B-Instruct (Dubey et al., 2024), Qwen2-7B-Instruct (Yang et al., 2024), and Qwen2.5-Instruct series (7B, 14B, and 32B) (Hui et al., 2024) as well as a black-box model, ChatGPT (i.e., GPT-3.5-Turbo-1106) (Achiam et al., 2023). We do not include Qwen2.5 series on Basketball for their accuracy is too low (See Table 7).

Answer Generation. For all the models, we use greedy search, selecting the token with the highest probability at each generation step.

Metrics. For each question q_i, we measure answer correctness using accuracy acc_i, where the generated answer is considered correct if it contains the ground-truth answer. The model's confidence c_i is defined as the generation probability of the answer, as described in §3.

Partial Correlation. For each signal, we compute the partial Spearman correlation coefficient to ensure reliable analysis by controlling for the effects of the other signals. For example, when analyzing how popularity influences confidence, we isolate the effect of Pop_{Ge} by controlling for RPop_{Ge} and Pop_{Q}. Details can be found in § B.3.

5 What Do LLMs Generate When They Hallucinate?

In this section, we first verify the relation between knowledge popularity and the QA performance, and then focus on what content the models generate when they hallucinate.

5.1 Ground-Truth-Side Popularity and Factual Recall

Unlike previous studies that focus on a single signal (Mallen et al., 2023; Kandpal et al., 2023), we compare the correlation strength between three different signals (i.e., Pop_{Q}, Pop_{GT}, and RPop_{GT}) and QA performance. For each signal, we control for the other two and compute its partial correlation

Table 3: Partial correlations between the three types of popularity and accuracy. We do not include Qwen2.5 series on Basketball for their accuracy is too low.

Model	$RPop_{GT}$	Pop_{Q}	Pop_{GT}
Movies			
Llama-3-8B	**+0.216**	+0.196	−0.019
Qwen2-7B	**+0.323**	+0.287	−0.039
ChatGPT	+0.083	**+0.094**	−0.026
Qwen2.5-7B	+0.285	**+0.315**	−0.005
Qwen2.5-14B	**+0.323**	+0.270	−0.038
Qwen2.5-32B	**+0.347**	+0.319	−0.041
Songs			
Llama-3-8B	**+0.391**	+0.102	−0.001
Qwen2-7B	**+0.345**	+0.220	+0.078
ChatGPT	**+0.264**	+0.010	+0.186
Qwen2.5-7B	**+0.213**	+0.196	−0.012
Qwen2.5-14B	**+0.322**	+0.219	+0.058
Qwen2.5-32B	**+0.306**	+0.231	−0.028
Basketball			
Llama-3-8B	+0.139	+0.080	**+0.259**
Qwen2-7B	+0.073	+0.000	**+0.321**
ChatGPT	**+0.245**	+0.219	+0.175

Table 4: Median popularity of ground-truth and generated answers on ChatGPT, grouped by correct and incorrect predictions. CV denotes the coefficient of variation, and Answer Unique Ratio is the fraction of unique generated answers among incorrect samples.

Metric	**Movies**	**Songs**	**Basketball**
Correct Answers			
Pop_{Q} (median)	22.0	10.0	9.0
Pop_{GT} (median)	28.0	64.0	132.0
$RPop_{GT}$ (median)	9.0	17.0	3.0
Incorrect Answers – Ground Truth			
Pop_{Q} (median)	18.0	7.0	7.0
Pop_{GT} (median)	21.0	31.5	75.0
$RPop_{GT}$ (median)	5.0	7.0	1.0
Incorrect Answers – Generated			
Pop_{Ge} (median)	5.0	39.0	186.0
$RPop_{Ge}$ (median)	2.0	4.5	1.0
Pop_{Ge} CV	1.58	0.90	0.58
Answer Unique Ratio	0.787	0.821	0.133

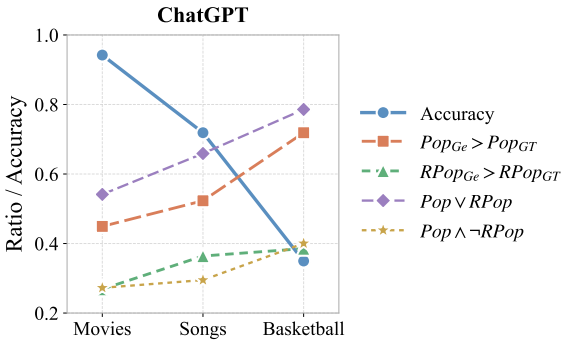

Figure 4: Comparison between generated and ground-truth entities on incorrectly answered samples.

with accuracy. The partial spearman coefficient results are shown in Table 3. The findings are: 1) Co-occurrence frequency exhibits the strongest correlation with model QA performance. 2) Relation specificity is also important. (See Figure 6). Due to space limitation, see details in § A.

5.2 Hallucinated Answers Are More Popular or Strongly Associated

We analyze the popularity of the entities generated by the model when it makes mistakes, as well as their co-occurrence frequency with the question entity, and compare them with the ground-truth answers. The results are shown in Figure 4 and Table 4. Due to space limitation, full results can be see in Table 8. Our conclusions are as follows.

LLMs tend to generate more familiar entities when they hallucinate. As shown in Figure 4, among the incorrectly answered samples, approximately 40% to 70% of the generated entities are more popular than the ground-truth entities, while 20% to 40% of the generated entities co-occur with the question entities more frequently than the correct answers do. Together, these two categories account for the majority of the incorrect samples. The former suggests that the model may increase the generation probability of an entity simply because it is more popular, assigning it a higher probability than the correct answer. For example, when asked about the director of a movie, the model does not know the correct answer and instead output a more well-known director. The latter suggests that the model can also be misled by entities that frequently co-occur with the question entity. For instance, it may answer with the producer of a movie instead of the director.

The less knowledge the model has about a domain, the more likely it is to generate more familiar answers. As shown in Figure 4, both the ratio of $RPop_{Ge} > RPop_{GT}$ and $Pop_{Ge} > Pop_{GT}$ increases from Movies to Songs to Basketball. The model's accuracy decreases progressively across these three datasets. These results suggest that, in more challenging domains, LLMs rely increasingly on popularity-related shortcuts rather than question-specific evidence.

Hallucinated answers become more concentrated as domain difficulty increases. If the

Table 5: Partial Spearman correlations between question-generated answer popularity and confidence. Bold marks the strongest positive predictor of confidence on incorrect answers. The ratio denotes how many times larger the correlation on incorrect samples is compared to that on correct samples. Ratios involving sign changes or near-zero denominators are marked "—" as they are not meaningful.

		$RPop_{Ge}$				Pop_Q				Pop_{Ge}			
Datasets	Models	Total	Correct	Incorrect	Ratio	Total	Correct	Incorrect	Ratio	Total	Correct	Incorrect	Ratio
Movies	Llama-3-8B	+0.541	+0.262	+0.267	1.0×	+0.135	+0.164	+0.045	0.3×	−0.014	+0.075	**+0.186**	2.5×
	Qwen2-7B	+0.541	+0.188	+0.179	1.0×	+0.104	+0.167	+0.046	0.3×	+0.112	+0.158	**+0.263**	1.7×
	ChatGPT	+0.172	+0.109	**+0.499**	4.6×	+0.092	+0.082	+0.060	0.7×	+0.045	+0.078	−0.112	—
	Qwen2.5-7B	+0.419	+0.178	+0.195	1.1×	+0.107	+0.172	+0.051	0.3×	+0.239	+0.156	**+0.310**	2.0×
	Qwen2.5-14B	+0.580	+0.287	+0.219	0.8×	+0.098	+0.142	+0.053	0.4×	+0.103	+0.053	**+0.262**	4.9×
	Qwen2.5-32B	+0.596	+0.254	+0.229	0.9×	+0.104	+0.133	+0.046	0.3×	+0.091	+0.074	**+0.242**	3.3×
Songs	Llama-3-8B	+0.609	+0.163	**+0.458**	2.8×	+0.107	+0.158	+0.105	0.7×	+0.026	−0.004	+0.012	—
	Qwen2-7B	+0.375	+0.065	**+0.140**	2.2×	+0.146	+0.076	**+0.084**	1.1×	+0.071	+0.048	**+0.105**	2.2×
	ChatGPT	+0.375	+0.181	**+0.656**	3.6×	+0.037	+0.050	+0.011	0.2×	+0.236	+0.255	−0.014	—
	Qwen2.5-7B	+0.357	+0.011	**+0.221**	—	+0.059	+0.002	+0.003	—	+0.158	+0.309	+0.205	0.7×
	Qwen2.5-14B	+0.451	+0.080	**+0.253**	3.1×	+0.063	+0.067	−0.008	—	+0.161	+0.243	+0.168	0.7×
	Qwen2.5-32B	+0.441	+0.059	**+0.237**	4.0×	+0.131	+0.126	+0.067	0.5×	+0.149	+0.147	**+0.194**	1.3×
Basketball	Llama-3-8B	−0.039	−0.046	−0.102	2.2×	+0.175	+0.214	**+0.158**	0.7×	−0.007	−0.177	−0.016	—
	Qwen2-7B	+0.008	+0.026	−0.024	—	+0.146	+0.015	**+0.168**	11.4×	+0.125	−0.009	**+0.099**	—
	ChatGPT	+0.188	+0.179	+0.040	0.2×	+0.255	+0.242	**+0.161**	0.7×	−0.179	−0.136	−0.100	0.7×

model relies more on popularity-based shortcuts under knowledge gaps, its outputs may become more concentrated, e.g., repeatedly generating a small set of well-known entities. To quantify this, we compute the coefficient of variation (CV) of Pop_{Ge} and the answer unique ratio. As shown in Table 4, both CV and the unique ratio decrease as domain difficulty increases, indicating that the model increasingly concentrates its outputs on a small set of highly popular entities. This concentration has implications for confidence, as discussed in § 6.

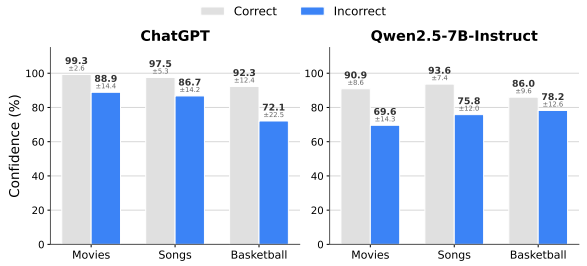

Figure 5: Mean confidence and its variation for correctly and incorrectly answered samples.

6 What Drives Confidence in Generated Answers?

In this section, we investigate the relationship between model confidence and the popularity of question-generated answer pair (Pop_Q, Pop_{Ge}, and $RPop_{Ge}$). The results are shown in Table 5. We find that from an overall perspective (Total column), **confidence is driven by question-generated answer popularity**, especially $RPop_{Ge}$. We further analyze correctly and incorrectly answered samples separately, and summarize the findings as follows.

Within correctly answered samples, uniformly high popularity and confidence leads to weak observable correlations. Although Table 5 shows weak correlations between popularity signals and confidence for correctly answered samples, this does not imply independence. Correct answers typically involve highly popular and strongly associated knowledge (See Table 4), driving confidence to consistently high with limited variance (Figure 5). When both popularity and confidence are uniformly high, their correlation is naturally hard to detect. This also helps explain why LLMs rarely exhibit conservativeness (Ni et al., 2024a; Xiong et al., 2023): correctly answered knowledge is generally popular, and models assign high confidence.

For incorrectly answered samples, model confidence is driven by question–answer pair popularity. As shown in Table 5, confidence for incorrect answers generally increases with question–answer pair popularity, though the dominant signal varies across datasets. In Movies and Songs, confidence is strongly correlated with $RPop_{Ge}$, while this effect disappears in Basketball. Moreover, the correlation between confidence and Pop_{Ge} weakens from Movies to Songs and is weakest in Basketball. As discussed in § 5.2, LLMs increasingly generate more popular and concentrated entities from Movies to Songs to Basketball. As entity popularity becomes uniformly high, selecting a more popular

entity provides diminishing additional gain in confidence, making confidence less sensitive to Pop_{Ge}. Basketball shows a different pattern: due to extremely sparse player–city co-occurrence (See Table 4), even higher RPop_{Ge} values remain small in absolute terms, so popularity differences lead to only marginal changes in confidence.

Scaling cannot mitigate the model's higher confidence in popular but incorrect answers. As shown in Table 5, within the Qwen2.5 model series, increasing model scale does not weaken the correlation between confidence and RPop_{Ge} for incorrectly answered samples in the Movies and Songs datasets. Similarly, in Basketball, confidence and Pop_{Q} remain high even for incorrect samples.

Taken together, These findings suggest a popularity-driven pathway to overconfidence: LLMs fall back on popular or strongly associated alternatives and assign high confidence due to the high popularity.

7 Calibrating Confidence with Popularity Signals

Although LLMs tend to generate more common answers when making errors, RPop_{Ge} remains substantially lower than in correctly answered samples (See Table 10). This motivates us to mitigate overconfidence via knowledge popularity.

7.1 Popularity-Aware Confidence Estimation

For a given question q, let the model's response be r and its confidence be c. We leverage the popularity signal p to map (p, c) to y, where y denotes the ground-truth confidence (i.e., correctness of the answer). We consider three types of the popularity signal p: Pop_{Q}, Pop_{Ge}, and RPop_{Ge}, as well as a combined setting that uses all three signals together. To reduce reliance on external corpora, we also explore using the LLM itself to estimate each popularity signal, denoted as $\widehat{\text{Pop}}_{\text{Q}}$, $\widehat{\text{Pop}}_{\text{Ge}}$, and $\widehat{\text{RPop}}_{\text{Ge}}$.

We learn a popularity-aware confidence estimator using a lightweight MLP, as defined below:

$$P(\hat{y} = 1) = \sigma\left(\text{MLP}\left(x\right)\right), \tag{2}$$

where σ refers to the sigmoid function, $x \in \mathcal{R}^d$ represents the input features, d is the count of input features (e.g., d=2 for c+Pop_{Q}). The MLP has three hidden layers with 128, 64, and 32 units, each followed by batch normalization, ReLU activation, and dropout (rate 0.3). We employ cross-entropy loss as the training objective:

$$\mathcal{L}_{\text{CE}} = -\sum_{i=1}^{N} y_i \log(P_i) + (1-y_i)\log(1-P_i), \tag{3}$$

where y_i is the ground-truth correctness for the i-th training sample, N is the count of training samples, and P_i denotes $P(\hat{y}_i = 1)$. Detailed training parameters can be found in § B.2.

7.2 Experimental Setup

Evaluation Metrics. To evaluate overconfidence mitigation and calibration quality, we report two calibration metrics: 1) $\textbf{Conf}_{\textbf{w}}$, the average confidence on incorrect samples, which directly measures overconfidence; 2) **Expected Calibration Error (ECE)** (Guo et al., 2017), which measures the discrepancy between predicted confidence and empirical accuracy. Beyond calibration quality, we additionally evaluate whether calibrated confidence better supports confidence-based binary decisions. We report **Alignment**, defined as the fraction of test samples whose binarized confidence prediction matches the ground-truth correctness. See Appendix B.2 for full details.

Baselines. We consider the following representative baseline methods. We exclude approaches that require access to internal model states, as they are not applicable when only output token probabilities are available.

- **Token probability** (Prob) (Jiang et al., 2021): the mean probability of generated tokens.
- **Self-consistency** (Consis) (Manakul et al., 2023): estimates confidence via semantic consistency among multiple sampled responses. If the model knows the correct answer, its outputs should be consistent. We sample 10 additional responses (temperature 1.0); for ChatGPT, we use 3 due to cost constraints. Consistency is computed using Qwen2.5-32B-Instruct.
- **Verbalized confidence** (Verbal) (Yin et al., 2023): prompts the model to explicitly judge whether it can answer the question correctly.
- **Temperature scaling** (Temp) (Guo et al., 2017): learns a single scalar temperature to rescale token-level probabilities for calibration.

7.3 Results and Analysis

Table 6 reports calibration and alignment results averaged over six open-source models.

Table 6: Conf_w (average confidence on incorrect samples), ECE, and Alignment averaged over six models. For Align., bold/underline = highest/lowest; for Conf_w and ECE, bold/underline = lowest/second-lowest. "All" denotes the use of all three types of popularity signals.

	Movies			Songs			Basketball			Avg.		
Method	Conf_w	ECE	Align.	Conf_w	ECE	Align.	Conf_w	ECE	Align.	Conf_w	ECE	Align.
Confidence alone												
Prob	0.780	0.387	81.88	0.775	0.355	80.72	0.741	0.326	68.66	0.765	0.356	77.08
Consis	0.320	0.157	79.33	0.295	0.121	83.52	**0.336**	0.173	67.16	0.317	0.150	76.67
Verbal	0.497	0.350	<u>64.97</u>	0.584	0.341	65.85	0.388	0.420	57.95	0.490	0.371	<u>62.92</u>
Prob + Temp	0.603	0.247	81.88	0.595	0.218	80.72	0.571	0.147	68.66	0.590	0.204	77.08
Corpora-based popularity												
Pop_Q	–	–	70.27	–	–	68.31	–	–	<u>55.02</u>	–	–	64.53
Pop_Ge	–	–	67.66	–	–	<u>63.20</u>	–	–	61.42	–	–	64.09
RPop_Ge	–	–	87.64	–	–	87.26	–	–	62.04	–	–	78.98
Prob + Pop_Q	0.257	0.024	83.17	0.315	0.102	81.75	0.395	<u>0.037</u>	68.71	0.322	0.054	77.88
Prob + Pop_Ge	0.260	0.025	82.73	0.340	<u>0.078</u>	81.92	<u>0.382</u>	0.049	71.57	0.327	0.050	78.74
Prob + RPop_Ge	<u>0.180</u>	0.024	89.54	**0.219**	0.084	**88.42**	0.395	0.045	69.87	<u>0.265</u>	0.051	82.61
Prob + All Pop.	**0.145**	**0.021**	**91.40**	<u>0.236</u>	0.095	87.97	0.382	**0.035**	**71.79**	**0.254**	0.050	**83.72**
Self-generated popularity												
Prob + $\widehat{\text{RPop}}_\text{Ge}$	0.259	<u>0.023</u>	83.02	0.318	0.078	82.77	0.405	0.042	68.66	0.327	**0.048**	78.15
Prob + All Pop.	0.246	0.024	83.49	0.312	**0.075**	83.57	0.384	0.048	69.99	0.314	<u>0.049</u>	79.02

Popularity signals substantially mitigate overconfidence. Adding popularity signals greatly reduces confidence on incorrect samples. Compared with vanilla Prob (Conf_w=0.765), Prob + RPop_Ge lowers Conf_w to 0.265 on average, while combining all popularity signals further reduces it to 0.254. The improvement is especially large on Movies and Songs, indicating that popularity-aware calibration effectively suppresses overconfident errors.

Popularity-aware calibration consistently improves calibration quality. Popularity signals also significantly reduce ECE. Vanilla Prob exhibits severe miscalibration (ECE=0.356), whereas popularity-enhanced methods reduce ECE to around 0.05. In contrast, temperature scaling only moderately improves ECE (0.356 $\rightarrow$ 0.204), suggesting that simple probability rescaling is insufficient for correcting overconfidence.

RPop_Ge is the most effective standalone signal. Among individual popularity features, RPop_Ge consistently yields the strongest performance. As a standalone detector, it already achieves high Alignment on Movies (87.64%) and Songs (87.26%).

Improved calibration leads to better confidence-based decisions. Better calibration further improves downstream decision accuracy. Prob + All Pop. achieves the best overall Alignment (83.72%), outperforming vanilla Prob by 6.6 points. This suggests that popularity-aware confidence estimates are more reliable for applications such as selective prediction and abstention.

Self-generated popularity signals remain effective. LLM-generated popularity estimates also improve calibration over confidence-only baselines. For example, Prob + All Pop. using self-generated signals improves average Alignment from 77.08% to 79.02% and reduces ECE from 0.356 to 0.049. However, the gains remain smaller than those achieved using external popularity signals.

8 Conclusion

This paper investigates LLM overconfidence in factual QA from the perspective of knowledge popularity. Our analysis shows that hallucinated answers are far from random: incorrect predictions are often more popular or more strongly associated with the question entity than the ground-truth answer. Moreover, model confidence is closely tied to generated-answer-side popularity signals, particularly question–answer co-occurrence, even when the generated answer is incorrect. These results suggest a popularity-associated pattern behind overconfident errors, where popular or strongly associated incorrect alternatives can receive disproportionately high confidence. Building on this observation, we show that popularity signals can also support confidence calibration, substantially reducing overconfidence and improving calibration quality.

Limitations

This work focuses on closed-book entity-centric factual QA, where knowledge popularity can be consistently measured through entity-level statistics and question–answer co-occurrence. While this setting allows us to systematically analyze the relationship between popularity and overconfidence, more complex scenarios such as multi-hop QA, open-ended generation, and reasoning-intensive tasks may involve additional factors beyond entity popularity and co-occurrence. In addition, our experiments mainly evaluate open-source LLMs, with ChatGPT included as a closed-source reference model. Due to the high cost of large-scale evaluation, we do not further test stronger proprietary models or specialized reasoning models. Extending the analysis to these settings would be a valuable direction for future work.

Ethics Statement

All datasets and models used are open-source or publicly accessible. Our analysis aims to characterize and partially mitigate overconfidence in LLM outputs, supporting more reliable downstream use; it does not introduce new harmful capabilities. Wikidata and Wikipedia are public resources used here only for entity- and document-level statistics. AI-based tools were used only for language polishing, grammar correction, and translation assistance. All original ideas, experiments, analyses, results, claims, and final manuscript content were created, verified, and reviewed by the authors.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Chao Chen, Kai Liu, Ze Chen, Yi Gu, Yue Wu, Mingyuan Tao, Zhihang Fu, and Jieping Ye. 2024. Inside: Llms' internal states retain the power of hallucination detection. *arXiv preprint arXiv:2402.03744*.

Shrey Desai and Greg Durrett. 2020. Calibration of pre-trained transformers. *arXiv preprint arXiv:2003.07892*.

Hanxing Ding, Liang Pang, Zihao Wei, Huawei Shen, and Xueqi Cheng. 2024. Retrieve only when it needs: Adaptive retrieval augmentation for hallucination mitigation in large language models. *arXiv preprint arXiv:2402.10612*.

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*.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. 2017. On calibration of modern neural networks. In *International conference on machine learning*, pages 1321–1330. PMLR.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

Binyuan Hui, Jian Yang, Zeyu Cui, Jiaxi Yang, Dayiheng Liu, Lei Zhang, Tianyu Liu, Jiajun Zhang, Bowen Yu, Keming Lu, et al. 2024. Qwen2. 5-coder technical report. *arXiv preprint arXiv:2409.12186*.

Zhengbao Jiang, Jun Araki, Haibo Ding, and Graham Neubig. 2021. How can we know when language models know? on the calibration of language models for question answering. *Transactions of the Association for Computational Linguistics*, 9:962–977.

Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, et al. 2022. Language models (mostly) know what they know. *arXiv preprint arXiv:2207.05221*.

Nikhil Kandpal, Haikang Deng, Adam Roberts, Eric Wallace, and Colin Raffel. 2023. Large language models struggle to learn long-tail knowledge. In *International conference on machine learning*, pages 15696–15707. PMLR.

Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. 2023. Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation. *arXiv preprint arXiv:2302.09664*.

Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. Teaching models to express their uncertainty in words. *arXiv preprint arXiv:2205.14334*.

Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, and Hannaneh Hajishirzi. 2023. When not to trust language models: Investigating effectiveness of parametric and non-parametric memories. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9802–9822, Toronto, Canada. Association for Computational Linguistics.

Potsawee Manakul, Adian Liusie, and Mark JF Gales. 2023. Selfcheckgpt: Zero-resource black-box hallucination detection for generative large language models. *arXiv preprint arXiv:2303.08896*.

Shiyu Ni, Keping Bi, Jiafeng Guo, and Xueqi Cheng. 2024a. When do llms need retrieval augmentation? mitigating llms' overconfidence helps retrieval augmentation. *arXiv preprint arXiv:2402.11457*.

Shiyu Ni, Keping Bi, Jiafeng Guo, Lulu Yu, Baolong Bi, and Xueqi Cheng. 2025. Towards fully exploiting llm internal states to enhance knowledge boundary perception. *arXiv preprint arXiv:2502.11677*.

Shiyu Ni, Keping Bi, Lulu Yu, and Jiafeng Guo. 2024b. Are large language models more honest in their probabilistic or verbalized confidence? *arXiv preprint arXiv:2408.09773*.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67.

Chenglei Si, Zhe Gan, Zhengyuan Yang, Shuohang Wang, Jianfeng Wang, Jordan Boyd-Graber, and Lijuan Wang. 2022. Prompting gpt-3 to be reliable. *arXiv preprint arXiv:2210.09150*.

Weihang Su, Changyue Wang, Qingyao Ai, Yiran Hu, Zhijing Wu, Yujia Zhou, and Yiqun Liu. 2024. Unsupervised real-time hallucination detection based on the internal states of large language models. *arXiv preprint arXiv:2403.06448*.

Katherine Tian, Eric Mitchell, Allan Zhou, Archit Sharma, Rafael Rafailov, Huaxiu Yao, Chelsea Finn, and Christopher D Manning. 2023. Just ask for calibration: Strategies for eliciting calibrated confidence scores from language models fine-tuned with human feedback. *arXiv preprint arXiv:2305.14975*.

Yanling Wang, Haoyang Li, Hao Zou, Jing Zhang, Xinlei He, Qi Li, and Ke Xu. 2024. Hidden question representations tell non-factuality within and across large language models. *arXiv e-prints*, pages arXiv–2406.

Miao Xiong, Zhiyuan Hu, Xinyang Lu, Yifei Li, Jie Fu, Junxian He, and Bryan Hooi. 2023. Can llms express their uncertainty? an empirical evaluation of confidence elicitation in llms. *arXiv preprint arXiv:2306.13063*.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, et al. 2024. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*.

Yuqing Yang, Ethan Chern, Xipeng Qiu, Graham Neubig, and Pengfei Liu. 2023. Alignment for honesty. *arXiv preprint arXiv:2312.07000*.

Zhangyue Yin, Qiushi Sun, Qipeng Guo, Jiawen Wu, Xipeng Qiu, and Xuanjing Huang. 2023. Do large language models know what they don't know? *arXiv preprint arXiv:2305.18153*.

Mert Yuksekgonul, Varun Chandrasekaran, Erik Jones, Suriya Gunasekar, Ranjita Naik, Hamid Palangi, Ece Kamar, and Besmira Nushi. 2023. Attention satisfies: A constraint-satisfaction lens on factual errors of language models. *arXiv preprint arXiv:2309.15098*.

Hanning Zhang, Shizhe Diao, Yong Lin, Yi Fung, Qing Lian, Xingyao Wang, Yangyi Chen, Heng Ji, and Tong Zhang. 2024. R-tuning: Instructing large language models to say 'i don't know'. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 7106–7132.

Jiaxin Zhang, Zhuohang Li, Kamalika Das, Bradley A Malin, and Sricharan Kumar. 2023. Sac3: Reliable hallucination detection in black-box language models via semantic-aware cross-check consistency. *arXiv preprint arXiv:2311.01740*.

A Knowledge Popularity Affects Model Learning

Co-occurrence frequency exhibits the strongest correlation with model QA performance. On Movies and Songs, RPop_{GT} is the dominant predictor, with partial correlations up to $r = 0.39$, substantially outperforming both question and answer popularity, which shows that factual recall is largely driven by the frequency of (question, answer) co-occurrence in pretraining corpora. On basketball, for both Llama-3-8B and Qwen2-7B, the signal most strongly correlated with accuracy is not RPop_{GT}, this may be attributed to their relatively weak performance on this dataset. As shown in Table 4, unlike the other two datasets, the co-occurrence between basketball players and their places of birth is too sparse, making it difficult for weaker models to learn effectively.

Relation specificity predicts accuracy beyond co-occurrence. We argue that co-occurrence count alone is insufficient to characterize learnability. The same frequency may correspond to either a highly focused relation (e.g., director is the dominant association of a movie) or a diffuse one where the relation is only one of many facts about the entity. We define relation specificity as $\frac{\text{RPop}_{\text{GT}}}{\text{Pop}_Q}$, where higher values indicate that the (question, answer) relation is more dominant for the question entity. To control for raw frequency, we bin examples by RPop_{GT} and compare high vs. low specificity within each bin. As shown in Figure 6, higher specificity consistently yields higher accuracy, suggesting that relation specificity is an important factor in factual learning.

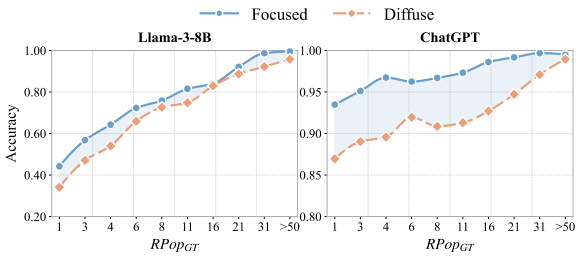

Figure 6: The impact of relation concentration on accuracy. Here, "focused" refers to the top half of samples ranked by $\text{RPop}_{\text{GT}}/\text{Pop}_Q$ within each RPop_{GT} bin, while "diffuse" refers to the bottom half of samples.

B Detailed Results

B.1 Per-Model Detection Results (Table 6)

Table 12–14 report per-model Alignment on Movies, Songs, and Basketball respectively. The pattern described in §7.2 holds consistently across all six models: RPop_{Ge} alone outperforms all confidence baselines on Movies and Songs, and Prob + RPop_{Ge} (MLP) yields the best single-feature result on Movies/Songs, while Pop_{Ge} is the effective complement on Basketball.

B.2 Training and Evaluation Details

Data preparation. For each (dataset, model) pair, we collect samples from the model's temperature-1.0 generations. We filter out samples where the model produces no answer. For Movies and Songs, we additionally discard samples where the question or generated entity has a Wikidata single-occurrence count above 6,000, as such ambiguous high-frequency entity names introduce noise. External popularity and co-occurrence values that cannot be found in Wikidata/Wikipedia are set to zero rather than discarded, reflecting the deployment scenario where not all entities appear in the knowledge base.

Class balancing. When the accuracy of a (dataset, model) pair falls outside $[0.25, 0.75]$, we undersample the majority class to achieve a 1:1 positive-to-negative ratio *before* the train/test split. This ensures that no method—including a trivial majority-class predictor—can achieve high Alignment from class imbalance alone. For models with balanced accuracy (e.g., Qwen2-7B on Movies, accuracy ≈ 0.38), no undersampling is applied.

Train/test split. The (possibly balanced) samples are randomly split 50/50 into train and test sets with stratified sampling (seed 42). All methods use the same split to ensure fair comparison.

Threshold selection. For threshold-based methods (CONF, SC, VC, and standalone popularity signals), the optimal binarization threshold is selected on the train set by scanning all unique values and choosing the one that maximizes train-set accuracy. For temperature scaling and Platt scaling, the rescaling parameter(s) are fitted on the train set, and then the optimal threshold is selected on the rescaled train-set probabilities.

MLP training. The MLP uses three hidden layers (128, 64, 32 units), each followed by batch normalization, ReLU activation, and dropout (rate 0.3). It is trained with Adam (learning rate 5×10^{-4}, weight decay 10^{-4}) and binary cross-entropy loss. From the train set, 10% is held out as a validation split (stratified, seed 42) for early stopping with patience 10. After training, the optimal binarization threshold is selected on the full train set, and Alignment is evaluated on the held-out test set.

B.3 Partial Correlation Computation

All partial correlation coefficients in this paper are computed using the rank-space OLS residual method, which yields the partial Spearman correlation:

1. **Rank transformation.** Convert each variable (predictor x, target y, and control variables $\mathbf{z} = \{z_1, \ldots, z_k\}$) to ranks.

2. **Residualize.** Regress each ranked variable on the ranked controls via ordinary least squares:

$$\hat{\beta}_x = \arg\min_{\beta} \|\text{rank}(x) - \mathbf{Z}\beta\|_2^2, \quad (4)$$

$$\hat{\beta}_y = \arg\min_{\beta} \|\text{rank}(y) - \mathbf{Z}\beta\|_2^2, \quad (5)$$

   where $\mathbf{Z} = [\text{rank}(z_1), \ldots, \text{rank}(z_k)]$ is the matrix of ranked control variables (mean-centered). The residuals

$$r_x = \text{rank}(x) - \mathbf{Z}\hat{\beta}_x, \quad (6)$$

$$r_y = \text{rank}(y) - \mathbf{Z}\hat{\beta}_y \quad (7)$$

   represent the parts of x and y not linearly explained by the controls, in the rank space.

3. **Correlate.** The partial Spearman correlation is $r = \text{Pearson}(r_x, r_y)$.

The three-variable mutual-control framework used throughout the paper selects three factors $\{f_1, f_2, f_3\}$ and, for each f_i, computes the partial

Table 7: Accuracy and average confidence for each model. We omit results for the Qwen2.5 series on Basketball in subsequent analyses due to their extremely low accuracy, which limits the reliability.

Datasets	Metrics	Llama-3-8B	Qwen2-7B	ChatGPT	Qwen2.5-7B	Qwen2.5-14B	Qwen2.5-32B
Movies	Acc	0.715	0.401	0.942	0.210	0.469	0.437
	Conf	0.904	0.817	0.986	0.754	0.877	0.877
Songs	Acc	0.364	0.255	0.719	0.117	0.247	0.185
	Conf	0.790	0.777	0.947	0.788	0.865	0.844
Basketball	Acc	0.134	0.098	0.350	0.042	0.074	0.078
	Conf	0.601	0.748	0.791	0.786	0.821	0.797

Table 8: Cross-dataset metrics for all models except ChatGPT. **Top**: median popularity on correct answers. **Middle**: median popularity on incorrect answers; CV = coefficient of variation of Pop_{Ge}. **Bottom**: ratios on incorrect answers; $P>G$ = $\text{Pop}_{\text{Ge}} > \text{Pop}_{\text{GT}}$; $R>R$ = $\text{RPop}_{\text{Ge}} > \text{RPop}_{\text{GT}}$; $P \vee R$ = either; $P \wedge \neg R$ = entity-popularity bias only.

		Correct (median)		Incorrect (median)			Incorrect (ratios)				
Model	Dataset	Pop_{GT}	RPop_{GT}	Pop_{Ge}	RPop_{Ge}	CV	Acc	$P>G$	$R>R$	$P \vee R$	$P \wedge \neg R$
Movies											
Llama-3-8B		32	11	26	0	0.97	0.715	0.557	0.101	0.579	0.478
Qwen2-7B		38	16	33	1	0.75	0.401	0.642	0.096	0.653	0.556
Qwen2.5-7B		46	24	30	0	0.82	0.210	0.561	0.047	0.568	0.521
Qwen2.5-14B		38	16	22	0	0.96	0.469	0.478	0.045	0.487	0.443
Qwen2.5-32B		38	16	24	0	0.90	0.437	0.551	0.057	0.564	0.508
Songs											
Llama-3-8B		66	26	39	1	0.83	0.364	0.448	0.159	0.517	0.358
Qwen2-7B		80	33	58	1	0.66	0.255	0.561	0.120	0.590	0.470
Qwen2.5-7B		81	52	70	1	0.60	0.117	0.596	0.095	0.615	0.520
Qwen2.5-14B		81	40	43	0	0.79	0.247	0.437	0.093	0.460	0.367
Qwen2.5-32B		76	44	57	0	0.69	0.185	0.516	0.091	0.535	0.444
Basketball											
Llama-3-8B		191	4	149	1	0.53	0.134	0.687	0.332	0.747	0.415
Qwen2-7B		223	3	191	2	0.41	0.098	0.782	0.424	0.843	0.419
Qwen2.5-7B		223	3	197	2	0.62	0.042	0.731	0.473	0.829	0.356
Qwen2.5-14B		223	4	211	2	0.47	0.074	0.788	0.452	0.839	0.387
Qwen2.5-32B		223	3	222	2	0.44	0.078	0.794	0.406	0.834	0.428

correlation between f_i and the target while controlling for $\{f_j : j \neq i\}$. This ensures that each reported coefficient captures the unique contribution of that factor after removing the linear influence of the other two. Two variable sets are used depending on type of knowledge:

- Ground-truth-side knowledge: $\{\text{RPop}_{\text{GT}}, \text{Pop}_{\text{Q}}, \text{Pop}_{\text{GT}}\}$, where RPop_{GT} is the Wikipedia co-occurrence count of the question entity and ground-truth answer, Pop_{Q} is the Wikidata sitelink count of the question entity, and Pop_{GT} is the Wikidata sitelink count of the ground-truth answer.

- Generation-side knowledge: $\{\text{RPop}_{\text{Ge}}, \text{Pop}_{\text{Q}}, \text{Pop}_{\text{Ge}}\}$, where RPop_{Ge} and Pop_{Ge} replace their ground-truth counterparts with the generated answer's equivalents, since the model can only observe its own output.

All analyses are performed independently per (dataset, model) pair without cross-model aggregation.

Table 9: Gene answer statistics by correctness. Pop_{Ge} = median popularity of generated answer; $RPop_{Ge}$ = median co-occurrence between question and generated answer; CV = coefficient of variation.

		Correct					Incorrect				
Model	Dataset	N	Pop_{Ge}	CV	$RPop_{Ge}$	CV	N	Pop_{Ge}	CV	$RPop_{Ge}$	CV
Llama-3-8B	Movies	6319	32.0	0.85	11.0	1.31	3205	17.0	1.22	0.0	4.95
	Songs	417	62.0	0.69	27.0	0.90	781	38.0	0.92	1.0	6.41
	Basketball	1771	191.0	0.47	4.0	2.34	11529	149.0	0.54	1.0	3.04
Qwen2-7B	Movies	3678	38.0	0.77	16.0	1.09	5892	31.0	0.83	1.0	4.39
	Songs	319	80.0	0.56	38.0	0.83	1011	51.0	0.75	0.0	9.77
	Basketball	1305	223.0	0.35	3.0	2.70	11995	191.0	0.42	2.0	2.87
ChatGPT	Movies	8768	27.0	0.90	9.0	1.32	828	5.0	1.58	2.0	2.34
	Songs	1034	66.0	0.63	18.0	1.10	436	39.0	0.90	4.5	1.90
	Basketball	4633	132.0	0.66	3.0	2.73	8667	186.0	0.58	1.0	3.47
Qwen2.5-7B	Movies	1749	45.0	0.71	24.0	3.88	7812	23.0	1.00	0.0	15.85
	Songs	147	87.0	0.56	56.0	0.76	1205	66.0	0.71	0.0	9.78
Qwen2.5-14B	Movies	3881	37.0	0.78	15.0	1.09	5662	15.0	1.23	0.0	17.75
	Songs	298	81.0	0.57	43.5	0.82	1055	38.0	0.94	0.0	9.83
Qwen2.5-32B	Movies	3747	38.0	0.77	16.0	1.07	5771	20.0	1.09	0.0	7.15
	Songs	241	78.0	0.58	48.0	0.78	1165	50.0	0.80	0.0	9.88

Table 10: Confidence-stratified Δ_{gcoo} per open-source model. Samples are binned by model confidence quartile (Q1 = lowest, Q4 = highest). Δ_{gcoo} = mean RPop_{Ge} of correct answers $-$ mean RPop_{Ge} of incorrect answers within each bin. Positive values indicate RPop_{Ge} carries correctness information orthogonal to confidence. "–" indicates bins with < 5 incorrect samples. Boldface marks the average row.

		Δ_{gcoo} by confidence quartile			
Model	Dataset	Q1 (low)	Q2	Q3	Q4 (high)
Movies					
Llama-3-8B		+7.6	+6.7	+3.1	+3.3
Qwen2-7B		+8.6	+11.9	+13.1	+18.2
Qwen2.5-7B		+14.8	+17.5	+32.6	+33.1
Qwen2.5-14B		+8.9	+9.0	+13.1	+11.1
Qwen2.5-32B		+8.6	+10.6	+15.1	+14.0
Avg		**+9.7**	**+11.1**	**+15.4**	**+16.0**
Songs					
Llama-3-8B		+18.2	+20.4	+4.9	−0.1
Qwen2-7B		+13.0	+28.2	+19.2	+24.6
Qwen2.5-7B		–	+48.7	+54.2	+24.6
Qwen2.5-14B		–	+30.0	+27.4	+2.8
Qwen2.5-32B		–	+32.3	+37.7	+6.1
Avg		**+15.6**	**+31.9**	**+28.7**	**+11.6**
Basketball					
Llama-3-8B		+4.0	+3.7	+3.8	+7.0
Qwen2-7B		+3.6	+1.3	+1.0	−1.6
Avg		**+3.8**	**+2.5**	**+2.4**	**+2.7**

Table 11: Confidence-stratified Δ_{gpop} per open-source model. Δ_{gpop} = mean Pop_{Ge} of correct answers − mean Pop_{Ge} of incorrect answers within each confidence bin. On Basketball, Δ_{gpop} is large and positive, reflecting that correct city answers (typically major NBA cities) have higher entity popularity than incorrect guesses. On Movies, Δ_{gpop} is near zero on average, as entity popularity alone does not strongly discriminate correct from incorrect person answers. "–" indicates bins with < 5 incorrect samples.

		Δ_{gpop} by confidence quartile			
Model	Dataset	Q1 (low)	Q2	Q3	Q4 (high)
Movies					
Llama-3-8B		−6.4	−11.6	−12.8	+7.2
Qwen2-7B		−10.2	−8.8	−12.3	−5.6
Qwen2.5-7B		+7.6	+1.3	+3.1	+3.9
Qwen2.5-14B		+7.3	+3.9	+2.7	−1.5
Qwen2.5-32B		+1.6	−1.3	+1.4	−2.5
Avg		**−0.0**	**−3.3**	**−3.6**	**+0.3**
Songs					
Llama-3-8B		+30.5	+22.1	+4.3	+39.3
Qwen2-7B		+17.7	+5.7	+13.8	+8.3
Qwen2.5-7B		–	−21.5	−13.2	−0.9
Qwen2.5-14B		–	+14.3	+11.4	+13.8
Qwen2.5-32B		–	−9.4	+1.1	+2.5
Avg		**+24.1**	**+2.2**	**+3.5**	**+12.6**
Basketball					
Llama-3-8B		+38.0	+48.0	+38.7	+11.0
Qwen2-7B		+17.7	+38.0	+25.7	+17.4
Avg		**+27.9**	**+43.0**	**+32.2**	**+14.2**

Table 12: Per-model Alignment (%) on Movies. Boldface marks the best method within each model column. All ext. = $\{\text{Pop}_{\text{Q}}, \text{Pop}_{\text{Ge}}, \text{RPop}_{\text{Ge}}\}$; All LLM = $\{\widehat{\text{Pop}}_{\text{Q}}, \widehat{\text{Pop}}_{\text{Ge}}, \widehat{\text{RPop}}_{\text{Ge}}\}$.

Method	Llama-3-8B	Qwen2-7B	ChatGPT	Qwen2.5-7B	Qwen2.5-14B	Qwen2.5-32B
Confidence alone						
Prob	83.2	81.1	78.6	82.0	81.7	84.7
Consis	74.1	81.4	70.9	82.1	83.7	83.7
Verbal	75.5	54.7	58.2	65.5	79.4	58.3
Prob + Temp	83.1	81.1	78.6	82.0	81.7	84.7
Corpora-based popularity						
Pop_{Q}	66.9	72.6	62.6	74.3	72.1	73.1
Pop_{Ge}	74.4	62.5	71.7	65.5	67.3	64.6
RPop_{Ge}	89.3	87.6	71.6	93.5	92.2	91.7
Popularity + confidence (MLP)						
Prob + Pop_{Q}	83.9	83.2	78.6	85.0	83.2	85.4
Prob + Pop_{Ge}	85.1	81.5	81.0	81.9	81.9	84.7
Prob + RPop_{Ge}	90.3	88.0	80.2	93.6	92.4	92.4
Prob + All ext.	92.9	92.0	82.5	94.7	93.2	93.7
Self-generated popularity (MLP)						
Prob + $\widehat{\text{RPop}}_{\text{Ge}}$	85.3	81.2	78.7	84.0	84.2	85.1
Prob + All LLM	85.9	81.1	75.2	86.8	85.4	87.1

Table 13: Per-model Alignment (%) on Songs. Boldface marks the best method within each model column.

Method	Llama-3-8B	Qwen2-7B	ChatGPT	Qwen2.5-7B	Qwen2.5-14B	Qwen2.5-32B
Confidence alone						
Prob	77.9	78.7	76.0	85.1	84.7	81.8
Consis	79.1	85.8	80.5	84.5	89.0	82.2
Verbal	62.5	54.3	74.9	65.5	79.4	57.9
Prob + Temp	77.9	78.7	76.2	85.1	84.7	81.8
Corpora-based popularity						
Pop_{Q}	69.8	68.8	68.0	64.2	71.8	67.4
Pop_{Ge}	67.9	62.0	71.0	54.7	67.8	55.8
RPop_{Ge}	81.6	88.6	78.3	91.2	93.4	90.5
Popularity + confidence (MLP)						
Prob + Pop_{Q}	79.8	82.1	77.1	83.8	88.7	82.2
Prob + Pop_{Ge}	79.4	79.6	76.6	86.5	85.0	83.1
Prob + RPop_{Ge}	84.4	89.5	79.3	93.9	94.0	91.3
Prob + All ext.	83.3	87.7	79.7	94.6	93.4	88.8
Self-generated popularity (MLP)						
Prob + $\widehat{\text{RPop}}_{\text{Ge}}$	80.7	79.9	76.0	87.8	88.0	83.1
Prob + All LLM	81.2	80.2	77.3	86.5	88.0	87.6

Table 14: Per-model Alignment (%) on Basketball. Boldface marks the best method within each model column.

Method	Llama-3-8B	Qwen2-7B	ChatGPT	Qwen2.5-7B	Qwen2.5-14B	Qwen2.5-32B
Confidence alone						
Prob	66.3	68.3	77.7	62.9	66.9	69.8
Consis	66.8	61.0	79.1	63.4	62.4	70.2
Verbal	62.7	51.4	55.8	65.5	50.0	62.2
Prob + Temp	66.3	68.3	77.7	63.2	66.9	70.1
Corpora-based popularity						
Pop_{Q}	57.1	51.5	69.6	48.3	52.0	51.6
Pop_{Ge}	60.0	61.1	65.2	61.6	57.5	63.1
RPop_{Ge}	66.3	61.0	65.2	56.5	59.4	63.9
Popularity + confidence (MLP)						
Prob + Pop_{Q}	67.1	67.9	78.1	62.5	65.7	68.7
Prob + Pop_{Ge}	68.5	69.0	78.3	72.5	68.2	73.4
Prob + RPop_{Ge}	71.5	69.0	78.1	63.4	66.5	71.3
Prob + All ext.	70.1	69.0	78.3	70.9	69.2	73.9
Self-generated popularity (MLP)						
Prob + $\widehat{\text{RPop}}_{\text{Ge}}$	66.3	69.0	77.5	61.6	66.1	70.0
Prob + All LLM	66.4	69.2	77.8	68.2	68.0	72.1