

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

Shiyu Ni^{1,2,3} Keping Bi^{1,2,3*} Jiafeng Guo^{1,2,3} Xueqi Cheng^{1,2,3}

¹State Key Laboratory of AI Safety

²Institute of Computing Technology, Chinese Academy of Sciences

³University of Chinese Academy of Sciences

{nishiyu23z, bikeping, guojiafeng, cxq}@ict.ac.cn

 Code

Abstract

Large language models (LLMs) often produce incorrect answers with high confidence, yet the factors associated with such overconfidence remain insufficiently understood. We study this problem through the lens of knowledge popularity. Using entity-centric factual QA derived from Wikidata triplets, we characterize popularity through question entity popularity, answer popularity, and question-answer co-occurrence. We find two consistent patterns. First, hallucinated answers are far from random: compared with ground-truth answers, they tend to be more popular or more frequently associated with the question entity. Second, confidence is strongly tied to the popularity of generated answers: even among incorrect predictions, more popular answers or those with higher question-answer co-occurrence receive higher confidence. Together, these findings suggest that popular but wrong alternatives may contribute to overconfidence. We further show that popularity-related signals can mitigate overconfidence and improve overall confidence estimation. Across six models and three datasets, incorporating knowledge popularity reduces average confidence on incorrect answers from 0.765 to 0.254 and overall ECE from 0.356 to 0.050, while improving Alignment from 77.08% to 83.72%.

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

*Corresponding author

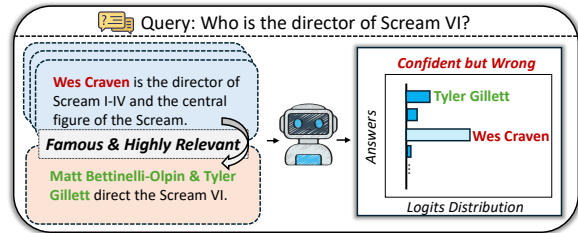


Figure 1: An example where the model generates an incorrect answer that is more popular than the correct answer, while assigning it high confidence.

such as healthcare and finance. Despite extensive evidence of LLM overconfidence, why incorrect answers can receive high confidence remains insufficiently understood.

Overconfidence can be viewed as a mismatch between the correctness of a generated answer and the confidence assigned to it. We suspect that one source of this mismatch lies in the frequency structure of the knowledge learned by LLMs. Repeated exposure to an entity may make it more likely to be generated, but such popularity does not guarantee that the entity is correct for the queried relation. This motivates us to study overconfidence through the lens of knowledge popularity.

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 required knowledge appears more frequently in the corpus (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. We therefore 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 among incorrect answers?

To answer these questions, we focus on factoid question answering (QA), which allows us to characterize generation-side popularity through question entity popularity, generated answer popularity, and question-answer co-occurrence. We employ three representative datasets constructed by prior work (Yuksekgonul et al., 2023), with examples shown in Figure 3. As the models’ training data are not accessible, we follow prior studies (Mallen et al., 2023; Kandpal et al., 2023; Yuksekgonul et al., 2023) and use external statistics as proxies, including Wikidata sitelinks for entity popularity and Wikipedia document co-occurrence for question-answer co-occurrence frequency.

Our results reveal a systematic popularity-related pattern behind overconfident errors. When factual recall fails, LLMs do not generate arbitrary wrong answers; instead, their errors tend to gravitate toward entities that are more popular or co-occur more frequently with the question entity than the ground-truth answer. More importantly, popularity is also related to how strongly models trust these errors: even among incorrect predictions, answers with stronger popularity signals tend to receive higher confidence. *Together, these findings suggest that popular but wrong alternatives may contribute to overconfidence.* The pattern is particularly evident when models struggle with the task and persists across model scales.

Although popular incorrect answers can receive high confidence, correct predictions typically exhibit even stronger and more reliable question-answer association patterns. We therefore leverage these signals for confidence calibration. Across six models and three datasets, incorporating knowledge popularity substantially mitigates overconfidence, reducing the average confidence on incorrect answers from 0.765 to 0.254, while improving overall confidence calibration, with ECE decreasing from 0.356 to 0.050 and Alignment increasing from 77.08% to 83.72%. 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 uses the probability assigned to an answer as a measure of model confidence (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 modern 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 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 been 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

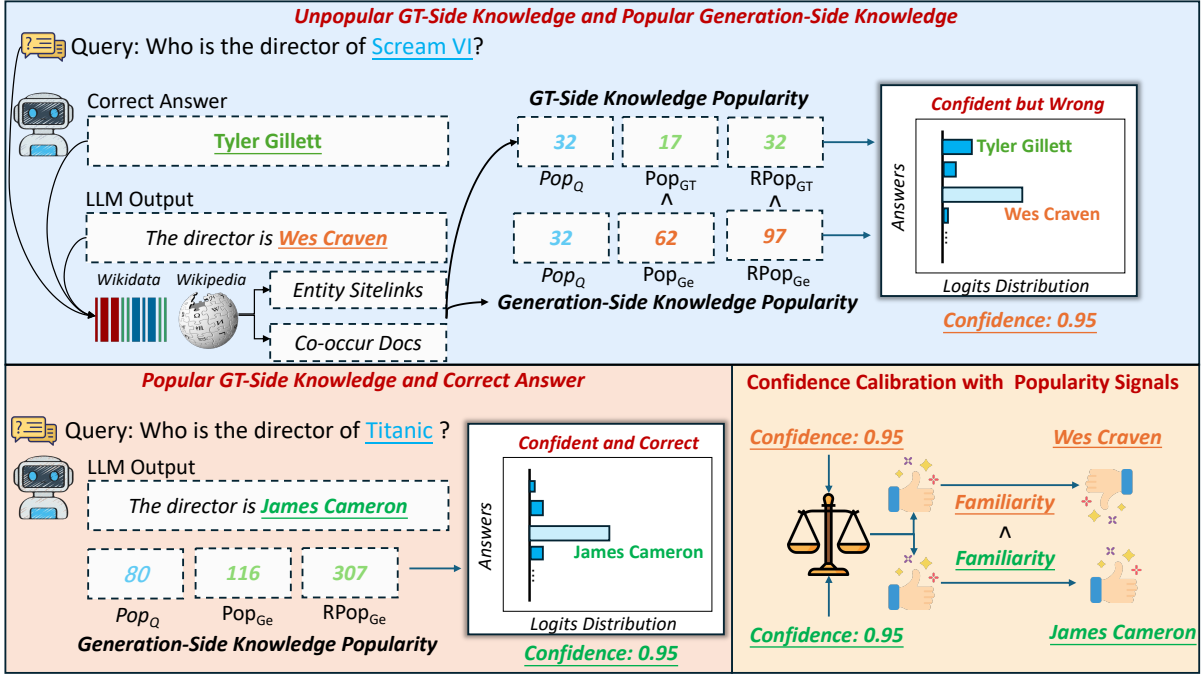


Figure 2: Overview of knowledge popularity, overconfidence, and calibration. The upper panel illustrates overconfident errors involving popular or frequently co-occurring incorrect answers; the lower panels show correct high-confidence predictions and popularity-aware confidence calibration.

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).

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.

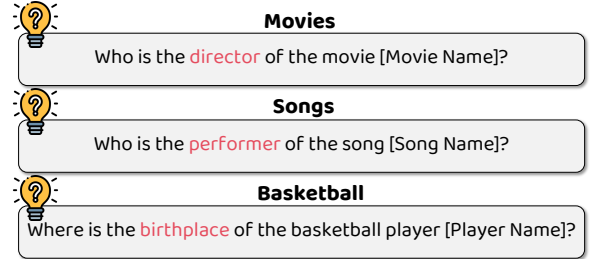


Figure 3: Question examples for each dataset.

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 the generation probabilities of the answer tokens. Specifically, for a question q and a model M , the confidence c is computed as $c = \frac{1}{T} \sum_{i=1}^T P(g_i | g_{<i})$, where $\{g_1, \dots, g_T\}$ are the generated tokens.

4 Experimental Setup

Datasets. Yuksekgonul et al. (2023) constructed entity-centric QA datasets from Wikidata¹, using sitelinks as a proxy for entity popularity, which cor-

¹<https://query.wikidata.org/sparql>

relates 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 dump² in which both entities co-occur.

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

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 2: Definitions of the popularity notation.

Notation	Definition
Pop _Q	Popularity of entities in the question
Pop _{GT}	Popularity of entities in the ground-truth answer
RPop _{GT}	Co-occurrence count between question and ground-truth entities
Pop _{Ge}	Popularity of entities in the generated answer
RPop _{Ge}	Co-occurrence count between question and generated entities

LLMs. We conduct experiments on six representative LLMs: five open-source models—Llama 3-8B-Instruct (Dubey et al., 2024), Qwen2-7B-Instruct (Yang et al., 2024a), and the Qwen2.5-Instruct series (7B, 14B, and 32B) (Yang et al., 2024b)—as well as a black-box model, ChatGPT (i.e., GPT-3.5-Turbo-1106) (Achiam et al., 2023). We do not include the Qwen2.5 series on Basketball because their accuracy is too low (see Table 9).

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

²<https://huggingface.co/datasets/wikimedia/wikipedia>

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.4.

5 What Do LLMs Generate When They Hallucinate?

In this section, we first examine the relationship between knowledge popularity and QA performance and then focus on what 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 strengths of the correlations 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 with accuracy. The partial Spearman correlation coefficients are shown in Table 8. 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 limitations, details are provided 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 3. Due to space limitations, the full results are provided in Table 10. Our conclusions are as follows.

LLMs tend to generate more familiar entities when they hallucinate. As shown in Figure 4,

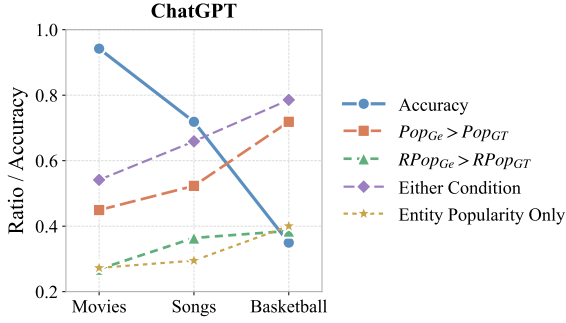


Figure 4: QA accuracy and failure modes across datasets for ChatGPT. Accuracy uses all samples, whereas failure-mode ratios use incorrect answers only. *Either Condition* means that either popularity comparison holds; *Entity Popularity Only* means that only $Pop_{Ge} > Pop_{GT}$ holds.

Table 3: 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
<i>Correct Answers</i>			
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
<i>Incorrect Answers – Ground Truth</i>			
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
<i>Incorrect Answers – Generated</i>			
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

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 may not know the correct answer and instead output a better-known director. The latter suggests that the model can also be misled by entities that frequently co-occur with the question

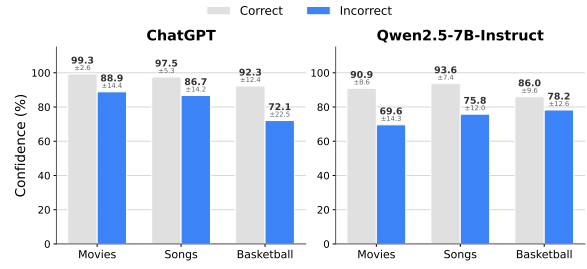


Figure 5: Mean confidence and its variation for correctly and incorrectly answered samples.

entity. For instance, it may answer with the producer of a movie instead of the director.

Substantial domain knowledge gaps are associated with stronger popularity-biased generation.

As shown in Figure 4, both popularity-bias ratios are substantially higher on Basketball, where model accuracy is lowest. Although the differences between Movies and Songs are modest, the contrast with Basketball suggests that, when models have substantially less knowledge about a domain, their hallucinations are more likely to favor answers that are globally popular or strongly associated with the question. Table 3 reports detailed popularity and question-answer association statistics for the ground-truth and generated answers, together with the CV and answer unique ratio to characterize the concentration of generated-answer distributions.

6 What Drives Confidence in Generated Answers?

In this section, we investigate the relationship between model confidence and the popularity signals associated with the question and generated answer (Pop_Q, Pop_{Ge}, and RPop_{Ge}). The results are shown in Table 4. Overall (Total column), we find that **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 lead to weak observable correlations. Although Table 4 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 3), driving confidence to consistently high levels with limited variance (Figure 5). When both popularity and confidence are

Table 4: 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 than on correct samples. Ratios involving sign changes or near-zero denominators are marked “—” as they are not meaningful.

Datasets	Models	RPop _{Ge}				Pop _Q				Pop _{Ge}			
		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×

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 4, 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 3), 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 4, within the Qwen2.5 model series, increasing model scale does not weaken the correlation between confidence and RPop_{Ge} for incor-

rectly 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-related pathway to overconfidence: when the correct question-answer association is weak, LLMs tend to generate more familiar but incorrect alternatives, and these alternatives also tend to receive high confidence.

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 12). This motivates us to mitigate overconfidence using knowledge-popularity signals.

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 a popularity signal p to map (p, c) to y , where y denotes the ground-truth confidence (i.e., the correctness of the answer). We consider three types of popularity signals: Pop_Q, Pop_{Ge}, and RPop_{Ge}, as well as a combined setting that uses all three signals. To reduce reliance on external corpora, we also explore using the LLM itself to estimate each popularity signal, denoted as $\widehat{\text{Pop}}_Q$, $\widehat{\text{Pop}}_{Ge}$, and $\widehat{\text{RPop}}_{Ge}$.

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

$$P(\hat{y} = 1) = \sigma(\text{MLP}(x)), \quad (1)$$

where σ denotes the sigmoid function, $x \in \mathbb{R}^d$ represents the input features, and d is the number of input features (e.g., $d = 2$ for $c + \text{Pop}_Q$). 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:

$$\mathcal{L}_{\text{CE}} = - \sum_{i=1}^N [y_i \log P_i + (1 - y_i) \log(1 - P_i)], \quad (2)$$

where y_i is the ground-truth correctness for the i -th training sample, N is the number 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) **Conf_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.
- **Platt scaling (Platt)** (Platt et al., 1999): fits a logistic mapping from token probability to answer correctness.
- **Isotonic regression (Isotonic)** (Zadrozny and Elkan, 2002): learns a non-parametric monotonic mapping from token probability to answer correctness.

Training and evaluation protocol. All methods are evaluated independently for each (dataset, model) pair. To prevent extreme QA accuracy on a dataset from causing Alignment to be unduly influenced by the majority class, we balance each (dataset, model) pair whose accuracy falls outside $[0.25, 0.75]$ by undersampling the majority class to a 1:1 ratio. We then apply a 50/50 stratified train/test split (seed 42). All parameter fitting and threshold selection use only the training split, and the held-out test split is used solely for evaluation. For MLP-based methods, we select the checkpoint using a 10% validation subset drawn from the training split. Full details are provided in Appendix B.2.

7.3 Results and Analysis

Table 5 reports calibration and alignment results averaged over six models.

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 improves both calibration and confidence-based decisions. Both standard supervised calibration and popularity-aware methods substantially reduce ECE. Platt scaling and isotonic regression reduce ECE from 0.356 with vanilla Prob to 0.105 and 0.038, respectively, but leave Alignment unchanged at 77.08%. Popularity-aware methods similarly achieve ECE around 0.05, while Prob + All Pop. improves Alignment from 77.08% to 83.72%. This contrast suggests that supervised calibration primarily corrects the probability scale, whereas popularity

Table 5: 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.

Method	Movies			Songs			Basketball			Avg.		
	Conf_w	ECE	Align.	Conf_w	ECE	Align.	Conf_w	ECE	Align.	Conf_w	ECE	Align.
<i>Confidence and calibration baselines</i>												
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
Prob + Platt	0.307	0.082	81.88	0.398	0.173	80.72	0.414	0.060	68.66	0.373	0.105	77.08
Prob + Isotonic	0.258	0.024	81.88	0.269	0.055	80.72	0.388	<u>0.035</u>	68.66	0.305	0.038	77.08
<i>Corpora-based popularity</i>												
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	0.037	68.71	0.322	0.054	77.88
Prob + Pop _{Ge}	0.260	0.025	82.73	0.340	0.078	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
<i>Self-generated popularity</i>												
Prob + RPop _{Ge}	0.259	<u>0.023</u>	83.02	0.318	0.078	82.77	0.405	0.042	68.66	0.327	<u>0.048</u>	78.15
Prob + All Pop.	0.246	0.024	83.49	0.312	<u>0.075</u>	83.57	0.384	0.048	69.99	0.314	0.049	79.02

signals additionally improve discrimination between correct and incorrect answers. We disentangle these contributions in detail in §7.4.

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%).

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.

7.4 Further Analysis

We further conduct fine-grained analyses to better understand the source of the calibration gains and their robustness across evaluation settings. In addition to ECE and Alignment, we report **Brier score** (Glenn et al., 1950), which measures sample-level probabilistic prediction error, and **AUROC** (Fawcett, 2006), which measures discrimination between correct and incorrect answers.

To isolate the contribution of popularity features from that of supervised nonlinear fitting, we introduce MLP(*c*), which takes confidence alone as

Table 6: Fine-grained comparison of calibration baselines and the capacity-matched ablation. The main setting follows the same evaluation protocol as Table 5; the natural setting retains the original class distribution.

Method	ECE↓	Brier↓	AUROC↑	Align.↑
<i>Balanced class distribution</i>				
Prob	0.356	0.322	0.839	77.08
Prob + Temp	0.204	0.227	0.839	77.08
Prob + Platt	0.105	0.172	0.839	77.08
Prob + Isotonic	0.038	0.156	0.837	77.08
MLP(<i>c</i>)	0.052	0.158	0.840	77.08
MLP(<i>c</i> + RPop _{Ge})	0.050	0.123	0.877	82.62
<i>Natural class distribution</i>				
Prob	0.513	0.421	0.836	86.29
Prob + Temp	0.345	0.258	0.836	86.29
Prob + Platt	0.054	0.108	0.836	86.29
Prob + Isotonic	0.017	0.101	0.835	86.23
MLP(<i>c</i>)	0.031	0.103	0.835	86.29
MLP(<i>c</i> + RPop _{Ge})	0.041	0.081	0.876	89.71

input while using the same architecture, objective, data split, and optimization procedure as our popularity-aware estimator, MLP(*c* + RPop_{Ge}).

Popularity mainly improves correctness discrimination. As shown in Table 6, the standard calibration baselines from Table 5, together with the capacity-matched ablation, show that supervised recalibration already accounts for most of the ECE reduction: MLP(*c*) reduces ECE from 0.356 to 0.052, comparable to 0.050 after adding RPop_{Ge}. In contrast, MLP(*c*) leaves AUROC nearly unchanged

Table 7: Cross-model and cross-dataset transfer under the natural class distribution.

Method	ECE↓	Brier↓	AUROC↑	Align.↑
<i>Cross-model transfer</i>				
MLP(<i>c</i>)	0.114	0.126	0.830	83.21
MLP(<i>c</i> + RPop _{Ge})	0.106	0.104	0.861	86.00
<i>Cross-dataset transfer</i>				
MLP(<i>c</i>)	0.137	0.144	0.828	80.37
MLP(<i>c</i> + RPop _{Ge})	0.147	0.139	0.845	81.14

(0.839→0.840) and Alignment at 77.08%, whereas adding RPop_{Ge} improves AUROC to 0.877 and Alignment to 82.62%. This suggests that the main additional benefit of popularity lies in improving correctness discrimination across samples, rather than further rescaling confidence values.

Popularity remains effective under the natural class distribution. We further evaluate all methods while retaining the original class distributions in both the training and test sets. As shown in Table 6, MLP(*c* + RPop_{Ge}) improves AUROC from 0.835 to 0.876 and Alignment from 86.29% to 89.71% over MLP(*c*). It also reduces the Brier score from 0.103 to 0.081 while maintaining a low ECE of 0.041. These results show that the additional discrimination provided by RPop_{Ge} also holds under the natural data distribution.

Popularity-aware calibration transfers across models and datasets. We further evaluate whether the learned calibration rule generalizes across models and datasets. For cross-model transfer, adding RPop_{Ge} improves AUROC from 0.830 to 0.861 and Alignment from 83.21% to 86.00%. For cross-dataset transfer, which also corresponds to transfer across relation types, AUROC improves from 0.828 to 0.845 and Alignment from 80.37% to 81.14%. Although the gains become smaller under cross-dataset transfer, RPop_{Ge} still provides useful transferable information beyond model confidence. Details are provided in the Appendix B.3.

8 Conclusion

This paper investigates LLM overconfidence in factual QA through the lens of knowledge popularity. We find that hallucinated answers are far from random: incorrect predictions are often more popular than the ground-truth answer or more strongly associated with the question entity. Moreover, model confidence is closely related to answer-side popularity signals, particularly

question–answer co-occurrence, even when the generated answer is incorrect. Together, these findings reveal a popularity-associated pattern of overconfidence, in which popular or strongly associated incorrect alternatives can receive high confidence. We further show that these popularity signals can be leveraged for confidence calibration, substantially mitigating overconfidence while improving overall calibration quality.

Limitations

Our study has several limitations. First, we do not have access to the actual training data of the evaluated LLMs, so our popularity measures serve only as proxies for how frequently particular knowledge may have been encountered during training. Second, our findings are correlational rather than causal: they reveal systematic associations between popularity and overconfidence, but do not establish that popularity itself causes overconfident predictions. Finally, our analysis focuses on closed-book entity-centric factual QA, where entity popularity and question-answer co-occurrence can be measured consistently. Whether these patterns extend to more complex settings, such as multi-hop QA, open-ended generation, and reasoning-intensive tasks, remains an important direction for future work.

Ethics Statement

All datasets and models used are open-source or publicly accessible. This work aims to understand and mitigate LLM overconfidence and does not introduce new harmful capabilities. Wikidata and Wikipedia are used only for entity- and document-level statistics. AI-based tools were used solely for language polishing, grammar correction, and translation; all ideas, experiments, analyses, results, and claims were produced and verified by the authors.

Acknowledgements

This work was funded by the National Natural Science Foundation of China (NSFC) under Grants No. 62302486 and No. 62441229, the Innovation Project of ICT CAS under Grants E561010 and No. E361140, and the Strategic Priority Research Program of the CAS under Grants No. XDB0680102.

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*.
- Tom Fawcett. 2006. An introduction to roc analysis. *Pattern recognition letters*, 27(8):861–874.
- W Brier Glenn et al. 1950. Verification of forecasts expressed in terms of probability. *Monthly weather review*, 78(1):1–3.
- 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.
- 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*.
- John Platt et al. 1999. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in large margin classifiers*, 10(3):61–74.
- 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. 2024a. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiayi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. 2024b. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*.

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 Yuksekogonul, 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*.

Bianca Zadrozny and Charles Elkan. 2002. Transforming classifier scores into accurate multiclass probability estimates. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 694–699.

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*.

Table 8: Partial correlations between the three types of popularity and accuracy. We do not include the Qwen2.5 series on Basketball because their accuracy is too low.

Model	RPop _{GT}	Pop _Q	Pop _{GT}
<i>Movies</i>			
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
<i>Songs</i>			
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
<i>Basketball</i>			
Llama-3-8B	+0.139	+0.080	+0.259
Qwen2-7B	+0.073	+0.000	+0.321
ChatGPT	+0.245	+0.219	+0.175

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. This result suggests that factual recall is strongly associated with the frequency of (question, answer) co-occurrence in pretraining corpora. On Basketball, the signal most strongly correlated with accuracy is not RPop_{GT} for either Llama-3-8B or Qwen2-7B. This result may be attributed to their relatively weak performance on this dataset. As shown in Table 3, 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 $\frac{RPop_{GT}}{Pop_Q}$, 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

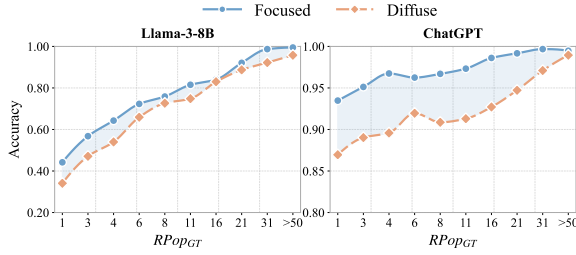


Figure 6: The impact of relation concentration on accuracy. Here, “focused” refers to the top half of samples ranked by $RPop_{GT}/Pop_Q$ within each $RPop_{GT}$ bin, while “diffuse” refers to the bottom half of samples.

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.

B Detailed Results

B.1 Per-Model Detection Results

Tables 14–16 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 and Songs, while Pop_{Ge} provides an effective complementary signal 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

whose QA accuracy already falls within this range (e.g., Qwen2-7B on Movies, accuracy ≈ 0.38), no undersampling is applied.

Evaluation settings. The main setting applies the conditional class-balancing procedure described above before the train/test split. The natural-distribution setting retains the original class distribution without undersampling. In both settings, the calibration methods and MLPs are fitted on the corresponding training split and evaluated on the held-out test split.

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.

Calibration fitting and threshold selection. All fitting and threshold selection are performed independently for each (dataset, model) pair using only the training split; the held-out test split is used solely for evaluation. For threshold-based methods (Prob, Consis, Verbal, and standalone popularity signals), the optimal binarization threshold is selected by scanning all unique training-set values and choosing the one that maximizes training-set accuracy. For temperature scaling, Platt scaling, and isotonic regression, the calibration mapping is first fitted on the training split, after which the optimal threshold is selected on the corresponding calibrated training-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. For each (dataset, model) pair, 10% of its training split is held out as a validation subset (stratified, seed 42). We retain the checkpoint with the lowest validation BCE loss and stop training after 10 consecutive epochs without improvement. After training, the optimal binarization threshold is selected on the full training split, and Alignment is evaluated on the held-out test split. In transfer experiments, checkpoint selection uses only the validation subset of the source pair; no target-test samples are used for training or model selection.

B.3 Full Calibration Transfer Results

For cross-dataset transfer, each method is fitted on the training split of a source dataset and evaluated

Table 9: Accuracy and average confidence for each model. We omit results for the Qwen2.5 series on Basketball in subsequent analyses because their extremely low accuracy limits the reliability of the estimates.

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 10: Accuracy and popularity-bias ratios across datasets for all models except ChatGPT. Ratios are computed on incorrect predictions. “Either” indicates that either $\text{Pop}_{\text{Ge}} > \text{Pop}_{\text{GT}}$ or $\text{RPop}_{\text{Ge}} > \text{RPop}_{\text{GT}}$ holds; “Pop only” indicates that only the former holds.

Dataset	Model	Acc.	Ratios on incorrect predictions			
			$\text{Pop}_{\text{Ge}} > \text{Pop}_{\text{GT}}$	$\text{RPop}_{\text{Ge}} > \text{RPop}_{\text{GT}}$	Either	Pop only
Movies	Llama-3-8B	0.715	0.557	0.101	0.579	0.478
	Qwen2-7B	0.401	0.642	0.096	0.653	0.556
	Qwen2.5-7B	0.210	0.561	0.047	0.568	0.521
	Qwen2.5-14B	0.469	0.478	0.045	0.487	0.443
	Qwen2.5-32B	0.437	0.551	0.057	0.564	0.508
Songs	Llama-3-8B	0.364	0.448	0.159	0.517	0.358
	Qwen2-7B	0.255	0.561	0.120	0.590	0.470
	Qwen2.5-7B	0.117	0.596	0.095	0.615	0.520
	Qwen2.5-14B	0.247	0.437	0.093	0.460	0.367
	Qwen2.5-32B	0.185	0.516	0.091	0.535	0.444
Basketball	Llama-3-8B	0.134	0.687	0.332	0.747	0.415
	Qwen2-7B	0.098	0.782	0.424	0.843	0.419
	Qwen2.5-7B	0.042	0.731	0.473	0.829	0.356
	Qwen2.5-14B	0.074	0.788	0.452	0.839	0.387
	Qwen2.5-32B	0.078	0.794	0.406	0.834	0.428

on the test split of a different target dataset for the same model. Results are macro-averaged over 36 ordered source–target pairs. For cross-model transfer, each method is fitted on a source model and evaluated on a different target model within the same dataset, with results macro-averaged over 90 ordered pairs. All fitted parameters, feature scalers, and decision thresholds come exclusively from the source training split; target labels are used only for evaluation. The main setting applies the conditional balancing procedure in Appendix B.2, whereas the natural-distribution setting retains the original class distribution for both source and target pairs.

B.4 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 $\mathbf{z}$ =

$\{z_1, \dots, z_k\}$) to ranks.

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

$$\hat{\beta}_x = \arg \min_{\beta} \|\text{rank}(x) - \mathbf{Z}\beta\|_2^2, \quad (3)$$

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

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

$$r_x = \text{rank}(x) - \mathbf{Z}\hat{\beta}_x, \quad (5)$$

$$r_y = \text{rank}(y) - \mathbf{Z}\hat{\beta}_y \quad (6)$$

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

Table 11: Generated-answer statistics by correctness. Pop_{Ge} and $RPop_{Ge}$ denote the median entity popularity and question–answer association strength, respectively. CV is reported only for Pop_{Ge} because $RPop_{Ge}$ is highly zero-inflated, making its CV unstable.

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

$\{f_1, f_2, f_3\}$ and, for each f_i , computes the partial 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 the type of knowledge:

- Ground-truth-side knowledge:
 $\{RPop_{GT}, Pop_Q, Pop_{GT}\}$, 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:
 $\{RPop_{Ge}, Pop_Q, Pop_{Ge}\}$, 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.

Table 12: Difference in mean question-generated answer association between correct and incorrect predictions within confidence quartiles for open-source models. Positive values indicate that correct answers have stronger question-answer associations than incorrect answers at similar confidence levels. “-” denotes fewer than five incorrect examples; Avg is the mean over available models.

Model	$\Delta\text{RPop}_{\text{Ge}}$ by confidence quartile			
	Q1 (low)	Q2	Q3	Q4 (high)
<i>Movies</i>				
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
<i>Songs</i>				
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
<i>Basketball</i>				
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 13: Difference in mean generated-answer entity popularity between correct and incorrect predictions within confidence quartiles for open-source models. Positive values indicate that correct answers are more popular than incorrect answers at similar confidence levels. “-” denotes fewer than five incorrect examples; Avg is the unweighted mean over available models.

Model	$\Delta\text{Pop}_{\text{Ge}}$ by confidence quartile			
	Q1 (low)	Q2	Q3	Q4 (high)
<i>Movies</i>				
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
<i>Songs</i>				
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
<i>Basketball</i>				
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 14: Per-model Alignment (%) on Movies. Boldface marks the best method within each model column. All ext. = {Pop_Q, Pop_{Ge}, RPop_{Ge}}; All LLM = {Pop_Q, Pop_{Ge}, RPop_{Ge}}.

Method	Llama-3-8B	Qwen2-7B	ChatGPT	Qwen2.5-7B	Qwen2.5-14B	Qwen2.5-32B
<i>Confidence alone</i>						
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
<i>Corpora-based popularity</i>						
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
<i>Popularity + confidence (MLP)</i>						
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
<i>Self-generated popularity (MLP)</i>						
Prob + RPop _{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 15: 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
<i>Confidence alone</i>						
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
<i>Corpora-based popularity</i>						
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
<i>Popularity + confidence (MLP)</i>						
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
<i>Self-generated popularity (MLP)</i>						
Prob + RPop _{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 16: 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
<i>Confidence alone</i>						
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
<i>Corpora-based popularity</i>						
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
<i>Popularity + confidence (MLP)</i>						
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
<i>Self-generated popularity (MLP)</i>						
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