What Models Know, How Well They Know It: Knowledge-Weighted Fine-Tuning for Learning When to Say "I Don't Know"

Anonymous ACL submission

Abstract

While large language models (LLMs) demonstrate strong capabilities across diverse user queries, they still suffer from hallucinations, often arising from knowledge misalignment between pre-training and fine-tuning. To address this misalignment, we reliably estimate a fine-grained, instance-level knowledge score via multi-sampled inference. Using the knowledge score, we scale the learning signal according to the model's existing knowledge, while encouraging explicit "I don't know" responses for out-of-scope queries. Experimental results show that this approach allows the model to explicitly express uncertainty when it lacks knowledge, while maintaining accuracy on questions it can answer. Furthermore, we propose evaluation metrics for uncertainty, showing that accurate discrimination between known and unknown instances consistently improves performance.

1 Introduction

LLMs perform effectively on many instruction-based tasks but still struggle with persistent hallucinations, often generating confident yet incorrect content. Recent studies (Cohen et al., 2024; Gekhman et al., 2024; Kang et al., 2025) identify a key cause of these hallucinations as knowledge misalignment between the parametric knowledge acquired during pre-training and the supervised fine-tuning datasets. When fine-tuning datasets contain knowledge that was rarely or never encountered during pre-training, models are compelled to produce labeled tokens even for facts they inherently do not "know." As a result, models tend to generate answers for unanswerable queries instead of explicitly acknowledging the absence of required knowledge, which directly leads to hallucinations. This suggests that fine-tuning on unknown data makes the model more likely to generate hallucinations.

Recent studies encourage uncertainty expression by training models to distinguish known from unknown data. **R-Tuning** (Zhang et al., 2024) trains the model by categorizing instances into known and unknown based on estimated knowledge, encouraging linguistic expressions of uncertainty for unknown instances. However, prior works (Gekhman et al., 2024; Zhong et al., 2025) show that the degree to which a model knows a fact substantially influences how the model learns, indicating that knowledge is better represented as a graded quantity rather than a binary one. **IDK-Tuning** (Cohen et al., 2024) and **SEAL** (Huang et al., 2025) instead allocate probability mass to a special refusal token to reject predictions that are inconsistent with the model's knowledge distribution. However, generating a refusal token during decoding enforces token-level decisions, leading to an inherent trade-off between accuracy and refusal behavior.

Question	SFT Response	KWT Response
Which genus contains more species: Fallopia or Nicandra?	Fallopia	Fallopia
Where was the host of Pressure Pad born?	Glasgow, Scotland	Glasgow, Scotland <IDK>

Table 1: Comparison of response behaviors with and without instance-level knowledge awareness. SFT generates confident but incorrect answers, whereas KWT generates <IDK> to express uncertainty. Blue and red text indicate correct and incorrect answer content, respectively.

Kadavath et al. (2022) suggests that LLMs exhibit latent self-knowledge through internal signals such as $\mathrm{P(True)}$ and $\mathrm{P(IK)}$. However, determining whether a model truly knows a fact based on a single response to a question can be unreliable. Therefore, our goal is to estimate the model's knowledge in a fine-grained manner and to train it to explicitly express "I don't know" when relevant knowledge is absent. Before introducing our method, we briefly illustrate why distinguishing knowledge during training is important. As shown

in Table 1, SFT generates answers regardless of whether the required knowledge is known, whereas KWT expresses uncertainty by generating <IDK> when the knowledge is absent. Specifically, we estimate knowledge by performing multi-sampled inference over the training data and computing a knowledge score based on the accuracy of generated responses. This score is then used as a fine-tuning signal, allowing the training strength to vary according to the model's existing knowledge. As a result, the model adjusts its training based on what it already knows and explicitly expresses uncertainty when its knowledge is insufficient.

We evaluate our approach on diverse question-answering datasets spanning general, medical, and scientific knowledge. Across the in-domain evaluations, our knowledge-weighted fine-tuning achieves accuracy comparable to standard supervised fine-tuning, while improving uncertainty expression on unknown queries. To further test generalization, we also evaluate out-of-domain datasets that explicitly separate answerable and unanswerable queries. In out-of-domain evaluations, our method improves the model's ability to express uncertainty, enabling more reliable handling of questions beyond its parametric knowledge.

Our main contributions are as follows:

- We show that estimating how well the model knows each instance is important for reducing hallucinations. To this end, we conduct experiments using multi-sampled inference and multiple answer-matching metrics to measure the model's knowledge more reliably.
- We demonstrate that how the model leverages this knowledge score during fine-tuning (e.g., through knowledge-dependent weighting strategies) plays a critical role in improving both uncertainty expression and overall performance.
- We introduce evaluation metrics to systematically measure knowledge awareness and uncertainty expression.

2 Related Works

2.1 Mitigating Hallucination

Approaches to reducing hallucinations in LLMs can be grouped into three categories. First, some methods use additional supervision. Tian et al. (2024) assigns hallucination scores and constructs preference pairs for DPO training, while Li et al. (2025) builds an uncertainty-aware MRC dataset that labels answerability to enable more reliable instruction tuning. Second, decoding-based methods reduce hallucinations by comparing multiple logit sources. Li et al. (2023b) contrasts expert and amateur models, Chuang et al. (2024) compares logits from intermediate layers, and Shi et al. (2024) contrasts logits with and without contextual evidence to select more factual next tokens. Third, test-time scaling approaches regenerate or select outputs at inference. Methods such as Madaan et al. (2023) and Mündler et al. (2024) iteratively refine initial answers using self-feedback, while other approaches (Chen et al., 2023; Wang et al., 2024b,a) generate multiple candidate responses and choose the most reliable one using self-consistency-style selection.

2.2 Knowledge-Aware Tuning

FT-TOP (Ghosal et al., 2024) reports that filtering the training data to retain only samples that the pre-trained model is estimated to already know leads to improved model performance. Ren et al. (2024) find that instruction fine-tuning is most effective when training data aligns with a model's existing internal knowledge, and that injecting inconsistent knowledge can degrade performance. **KaFT** (Zhong et al., 2025) detects conflicts between a model's parametric knowledge and supervision data and assigns discrete weights to improve domain-specific multiple-choice QA. Since KaFT operates in a multiple-choice setting, correctness is directly determined by option selection, without addressing semantic matching challenges in free-form generation.

We focus on reducing hallucinations by addressing knowledge misalignment between pre-training and fine-tuning. We study open-ended question answering by estimating instance-level knowledge as a fine-grained value to guide fine-tuning. By using a special token to express uncertainty on unknown instances, our method supports uncertainty-aware responses while remaining compatible with standard answer generation.

3 Method

We conduct experiments on question–answering datasets. Each fine-tuning sample is represented as (q, r^*, k), where q is the input question, r^* is the gold response, and k is an optional supporting knowledge paragraph when such information is available. We denote the pre-trained model as M_{base}. Given a question q, M_{base} generates a pre-

Metric	KP-Acc	KP-F1	KNP-Acc	KNP-F1
EM	63.2	57.2	71.2	55.8
Rouge	82.4	85.8	83.8	81.6
LLM-as-a-judge	93.4	94.8	93.8	93.4

Table 2: Agreement of automatic metrics (EM, ROUGE, LLM) with human evaluation under knowledge-provided (KP) and knowledge-not-provided (KNP) settings.

diction $\hat{r}$, which is compared against the gold response r^* to estimate the model's knowledge of the instance. When available, the knowledge paragraph k is used for correctness evaluation under the LLM-as-a-judge framework (Zheng et al., 2023).

To obtain a reliable estimate of the model's knowledge, we infer multiple responses rather than relying on a single $\hat{r}$, and assess their consistency with the gold response via multi-sampled inference. The resulting fine-grained knowledge score is used to modulate the fine-tuning signal, so that each instance is trained in proportion to the model's prior knowledge. For instances with insufficient knowledge, we additionally encourage the model to generate a special <IDK> token. As a result, the model learns to answer confidently when sufficient knowledge is present, while explicitly refusing to answer by producing <IDK> when the required knowledge is absent.

3.1 Knowledge Estimation

Even if M_{base} has parametric knowledge of a fact, it may not reliably express that knowledge during generation. Although "not generated" and "not known" are conceptually distinct, following prior works (Gekhman et al., 2024; Yang et al., 2024; Zhang et al., 2024), we operationalize knowledge as the ability to produce a correct answer under typical inference conditions, and treat failures to do so as effectively unknown for fine-tuning. We compute the model's knowledge not as a binary known-or-unknown classification, but as a fine-grained score derived through the following steps.

Multi-sampled few-shot inference for robust knowledge estimation. Because pre-trained models may not reliably follow instructions, we probe knowledge using 3-shot prompting. For each question, we sample three demonstration QA pairs to form a 3-shot prompt and then sample one response. We repeat this process with independently resampled demonstrations $S = 5$ times, yielding responses under diverse prompting conditions. Inspired by self-consistency (Wang et al., 2023), this produces a more robust estimate of the model's parametric knowledge.

Measuring correctness by comparing sampled responses with gold answers. Each sampled response is evaluated against the gold answer to determine whether it is correct. We attempt to compare the two responses using the following three methods: Exact Match (EM) to detect strictly correct predictions but may miss semantically correct answers with different surface forms. Following Kuhn et al. (2023), we deem a response correct if its ROUGE score exceeds a certain threshold, allowing lexically varied but semantically aligned answers to be counted as correct. Lastly, we use an LLM-as-a-judge, which compares the response and the gold answer to determine semantic correctness. The prompts for evaluating the LLM are described in Appendix A.2.

Consequently, we compute a fine-grained knowledge score by averaging the correctness of the sampled responses. Given generated responses ($\{\hat{r}_i\}_{i=1}^{S}$) and a matching function ($f(\cdot)$) that determines whether a response matches the gold response, the knowledge score is defined as:

$$\text{KnowledgeScore}(q) = \frac{1}{S}\sum_{i=1}^{S} f\left(\hat{r}_i, r^*\right) \quad (1)$$

To assess the reliability of these automatic correctness metrics, we obtain human judgments from a single annotator for 100 questions (500 responses) drawn from both a knowledge-provided dataset (HaluEval) and a knowledge-not-provided dataset (MedQA). For the knowledge-provided dataset, the relevant knowledge paragraph is included in the prompt when evaluating responses with the LLM-as-a-judge method. As shown in Table 2, LLM-as-a-judge exhibits the highest agreement with human judgments, followed by ROUGE and EM, across both datasets. The ROUGE threshold is selected based on the configuration that yields the highest agreement with human judgments. These results indicate that LLM-as-a-judge is the most suitable matching function. Unless otherwise specified, we therefore compute knowledge scores using the LLM-as-a-judge metric and evaluate all reported performance metrics using the same criterion.

3.2 Knowledge-Weighted Fine-Tuning

We assign sample-specific training weights using the instance-level knowledge scores, whose dis-

	EM			Rouge			LLM		
KnowledgeScore (KS)	**HaluEval**	**MedQA**	**SciQ**	**HaluEval**	**MedQA**	**SciQ**	**HaluEval**	**MedQA**	**SciQ**
0	66.75	87.98	40.00	49.83	80.28	25.84	47.74	67.34	14.38
0.2	9.34	6.60	12.66	12.59	10.26	11.97	13.13	15.74	9.07
0.4	7.24	2.63	9.58	9.41	4.45	9.80	8.80	7.69	9.21
0.6	5.85	1.56	9.21	8.48	2.63	10.75	8.55	4.75	10.79
0.8	5.68	0.87	10.52	8.76	1.58	13.21	9.34	2.77	15.93
1.0	5.15	0.34	18.02	10.94	0.80	28.44	12.45	1.71	40.62

Table 3: Distribution (%) of questions across knowledge score bins, computed from multi-sampled model responses generated by Llama 3.2-3B and evaluated using EM, ROUGE, and LLM-based metrics.

tribution is reported in Table 3. Specifically, w_i denotes the training weight assigned to the i-th sample, which modulates the contribution of each instance to the overall loss during fine-tuning:

$$\mathcal{L} = -\frac{1}{NM}\sum_{i=1}^{N} w_i \sum_{t=1}^{M} \log p(x_t^{(i)} \mid x_{<t}^{(i)}). \tag{2}$$

where $x^{(i)} = (x_1^{(i)}, \ldots, x_M^{(i)})$ is the target token sequence of the i-th training sample, N denotes the total number of training samples, and M denotes the number of target tokens in each sample.

To understand how the knowledge score relates to effective fine-tuning, we explore three weighting strategies: (1) **Familiarity (F)** weighting (w^f), which increases the contribution of samples the model already knows well; (2) **Reverse-Familiarity (RF)** weighting (w^{rf}), which down-weights well-known samples; and (3) **Uniform (U)** weighting (w^u), which keeps all weights identical. Specifically, we use:

$$w_i^f = \frac{S \times \mathrm{KS}_i + 1}{S + 1}, \tag{3}$$

$$w_i^{rf} = 1 - \frac{S \times \mathrm{KS}_i}{S + 1}, \tag{4}$$

$$w_i^u = 1.0, \tag{5}$$

where KS_i represents the knowledge score of the i-th sample and S denotes the number of sampled responses. We adopt the F strategy as our default weighting strategy. The effect of different weighting strategies is discussed in Section 4.8.

For samples with a knowledge score of zero, which indicates that the model has no knowledge of the underlying fact, we append a special <IDK> token (representing "I don't know") to the end of the target response to encourage explicit uncertainty. In Appendix D, this approach is shown to be simple yet effective, with appending <IDK> yielding better performance than replacing the entire response with <IDK>. Unlike full replacement, our method preserves performance on answerable questions while still promoting appropriate uncertainty expression. At inference time, users can choose to allow the <IDK> token to enable explicit uncertainty expression, or suppress it to use the model as a standard answer-generating LLM.

4 Experiments

4.1 Implementation Details

We conduct all experiments using the pretrained Llama 3.2-3B (Grattafiori et al., 2024). For KWT (ROUGE), we use ROUGE-L F1 as the matching metric. The correctness thresholds are set to 0.35, 0.6, and 0.6 for HaluEval, MedQA, and SciQ, respectively. When estimating knowledge scores via multi-sampled inference, we use a temperature of 0.7 to encourage diverse generations. The LLM-as-a-judge model used in our experiments is the instructed Gemma-3 12B (Gemma Team and Google DeepMind, 2025). All models are trained under the same experimental settings for 3 epochs. To demonstrate that our findings generalize across models, we additionally conduct experiments on Qwen 2.5-3B (Qwen Team, 2025) in Appendix C.

4.2 Datasets

We empirically evaluate our method across diverse open-domain QA datasets, including HaluEval (Li et al., 2023a) for capturing general hallucination scenarios, MedQA (Jin et al., 2021) for medical knowledge assessment, and SciQ (Welbl et al., 2017) for scientific commonsense evaluation. For HaluEval, we use the provided 10,000 samples and split them into training and test sets with an 8:2 ratio. Additionally, we evaluated on explicitly answerable/unanswerable datasets such as NEC (Liu et al., 2024), RefuNQ (Liu et al., 2024), and Self-Aware (Yin et al., 2023), verifying the model's generalized capability to distinguish and appropriately express uncertainty across out-of-domain.

4.3 Evaluation Metrics

Evaluating a model's ability to appropriately respond with uncertainty is non-trivial. Standard accuracy only measures answer correctness and does not reward correct abstention. Kalai et al. (2025) show that standard accuracy-based evaluation inherently penalizes abstention and reward confident guessing, which reinforces hallucination-like behavior rather than discouraging it. This motivates the use of evaluation metrics that explicitly account for appropriate uncertainty expression, and we therefore adopt the following metrics to capture different aspects of uncertainty-aware behavior. All evaluation metrics described below are computed based on the four outcome categories (A, B, C, D) defined in Table 4.

Normalized Area Under the Product Curve (nAUPC). Uncertainty-Aware Accuracy (UA-Acc) measures a model's ability to appropriately abstain on incorrect answers by assigning partial credit:

$$\text{UA-Acc}(\alpha) = \frac{\alpha B + C}{A + B + C + D} \tag{6}$$

where α controls the reward for abstention on incorrect answers.

Since UA-Acc does not penalize abstention on correct answers, we define the complementary Certainty-Aware Accuracy (CA-Acc) as:

$$\text{CA-Acc}(\alpha) = \frac{C - \alpha A}{A + B + C + D} \tag{7}$$

where α penalizes unnecessary abstention.

Varying α induces a continuum of operating points that reflect different balances between uncertainty expression and certainty. To summarize this trade-off with a single scalar, we define the nAUPC as:

$$\begin{aligned}\text{nAUPC} &= \frac{1}{\alpha_{\max}} \int_0^{\alpha_{\max}} \text{UA-Acc}(\alpha)\,\text{CA-Acc}(\alpha)\,d\alpha \\ &\approx \frac{1}{N\,\alpha_{\max}} \sum_{k=1}^{K} \Big(\text{UA-Acc}(\alpha_{k-1})\text{CA-Acc}(\alpha_{k-1}) \\ &\quad + \text{UA-Acc}(\alpha_k)\text{CA-Acc}(\alpha_k)\Big)(\alpha_k - \alpha_{k-1})\end{aligned} \tag{8}$$

where $N = 200$ is a normalization constant accounting for trapezoidal integration and percentage scaling, and $\{\alpha_k\}_{k=0}^{K}$ are uniformly spaced values in $[0, \alpha_{\max}]$. This definition assigns a high score only when both uncertainty-aware abstention (UA-Acc) and certainty preservation (CA-Acc) are simultaneously high across the range of α. We set $\alpha_{\max} = 1.0$ in all experiments.

	Correct	Incorrect
Response w/ IDK	A	B
Response w/o IDK	C	D

Table 4: Confusion matrix of responses based on correctness and IDK usage.

Abstain False Positive Rate (A-FPR). The proportion of cases in which the model includes <IDK> in its response even though the answer is correct. Such cases indicate unnecessary uncertainty expression that may confuse users.

$$\text{A-FPR} = \frac{A}{A + C} \tag{9}$$

IDK Precision. IDK Precision quantifies how frequently an <IDK> prediction is justified, measured as the fraction of <IDK> responses that correspond to incorrect answers.

$$\text{IDK-Precision} = \frac{B}{A + B} \tag{10}$$

4.4 Compared Methods

SFT is standard fine-tuning where M_{base} is trained without distinguishing whether each sample is known to the model. In **FT-TOP**, M_{base} is fine-tuned only on samples it is estimated to already know. As a result, neither method is capable of producing IDK responses. **R-Tuning** is a method that trains M_{base} by appending an uncertainty expression to the response on samples it does not know. For a fair comparison with our method, this uncertainty expression is replaced with the <IDK> token during our experiments. **SEAL** trains M_{base} by comparing the predicted token distribution with the gold token at each step. When the top-1 prediction does not match the gold token, it reallocates a portion of the one-hot target probability to a special <IDK> token, enabling the model to express uncertainty when its knowledge is unreliable.

4.5 Results on In-Domain QA

Table 6 presents the results of our method and baselines. KWT performance consistently reflects the quality of the underlying knowledge estimation. KWT (LLM) achieves the strongest uncertainty-aware performance, with higher nAUPC, lower A-FPR, and higher IDK Precision than KWT (ROUGE) and KWT (EM), demonstrating that more reliable instance-level knowledge estimation leads to better-calibrated uncertainty-aware behavior. Appending an <IDK> token to samples with zero knowledge score enables precise uncertainty

Model	HaluEval	MedQA	SciQ
SFT	31.0	20.4	62.7
FT-TOP	29.5	17.2	63.9
R-Tuning	30.3	19.7	64.6
KWT (EM)	30.7	16.9	62.1
KWT (Rouge)	30.3	16.9	63.8
KWT (LLM)	29.8	18.5	62.6

Table 5: Accuracy comparison across datasets, where accuracy is defined as $(A + C)/(A + B + C + D)$.

Dataset	Model	nAUPC ↑	A-FPR ↓	IDK Precision ↑
HaluEval	R-Tuning	2.7	50.2	80.4
	KWT (EM)	8.8	25.4	86.7
	KWT (Rouge)	10.2	14.5	88.9
	KWT (LLM)	**11.3**	**11.41**	**92.1**
MedQA	R-Tuning	-2.0	75.3	82.9
	KWT (EM)	1.3	49.3	87.6
	KWT (Rouge)	2.1	37.7	88.5
	KWT (LLM)	**4.3**	**21.6**	**91.2**
SciQ	R-Tuning	9.9	44.4	47.6
	KWT (EM)	26.4	19.0	62.1
	KWT (Rouge)	34.0	11.0	66.8
	KWT (LLM)	**38.1**	**3.7**	**78.3**

Table 6: Comparison of uncertainty-aware behavior on the in-domain datasets. **Bold** indicates the best performance in each column.

expression without sacrificing accuracy on answerable queries.

Acc. Table 5 shows that KWT incurs only a slight decrease in standard accuracy compared to SFT. This result indicates that uncertainty-aware training introduces only a modest trade-off in answer correctness. A more detailed analysis of the impact of <IDK> supervision on standard accuracy is provided in Appendix 6. Under the uncertainty-aware evaluation in Table 6, KWT (LLM) attains the highest nAUPC with lower A-FPR and higher IDK Precision than other methods, suggesting that instance-level knowledge weighting improves uncertainty expression without degrading standard accuracy compared to R-Tuning.

IDK expression. KWT consistently achieves lower A-FPR and higher IDK Precision than R-Tuning, indicating reduced unnecessary abstention while producing <IDK> responses when appropriate. This improvement stems from knowledge-weighted fine-tuning, which better distinguishes answerable from unanswerable queries.

4.6 Results on Out-of-Domain QA

We further evaluate our method on out-of-domain datasets. Since all models are fine-tuned on the same in-domain data with an identical base model, differences in performance primarily reflect the effect of the fine-tuning strategy rather than domain-specific knowledge advantages.

Dataset	Model	nAUPC ↑	A-FPR ↓	IDK Precision ↑
RefuNQ	R-Tuning	-1.9	73.3	81.6
	KWT (EM)	2.2	45.8	86.0
	KWT (Rouge)	5.4	22.1	90.7
	KWT (LLM)	**5.7**	**19.7**	**91.0**
SelfAware	R-Tuning	-2.0	70.7	74.7
	KWT (EM)	4.6	42.3	80.5
	KWT (Rouge)	8.7	19.6	85.0
	KWT (LLM)	**9.9**	**17.1**	**87.5**

Table 7: Comparison of uncertainty-aware behavior on the out-of-domain datasets: RefuNQ and SelfAware.

As shown in Table 7, KWT exhibits similar trends on out-of-domain datasets. In particular, KWT maintains comparable accuracy while achieving improved uncertainty-aware behavior, indicating that knowledge-weighted fine-tuning generalizes effectively beyond the training distribution. Detailed analysis on explicitly unanswerable queries and the corresponding IDK expression is deferred to Appendix B.

4.7 Comparison to Token-Level Abstention

Table 8 compares our method with SEAL, a token-level uncertainty expression approach. Since SEAL treats any response containing <IDK> as incorrect, it enforces $A = 0$, trivially yielding an A-FPR of 0 and an IDK Precision of 100, which makes these metrics uninformative for comparison.

This behavior also limits the interpretability of nAUPC, as nAUPC largely mirrors standard accuracy trends in SEAL rather than reflecting calibrated uncertainty-aware trade-offs. Under standard accuracy evaluation, SEAL consistently underperforms KWT across in-domain datasets and exhibits substantial performance degradation on out-of-domain benchmarks, indicating limited generalization beyond the training distribution.

4.8 Effect of Different Weighting Strategies

We examine not only how the knowledge score should be estimated, but also how it should be utilized during fine-tuning. We conduct experiments using the alternative weighting strategies introduced in Section 3.2.

Table 9 reports the results of the reverse-familiarity (RF) and uniform (U) weighting strategies. While emphasizing low-knowledge samples can provide limited benefits in certain settings, RF

Domain	Dataset	Acc ↑	nAUPC ↑
ID	HaluEval	27.9 (–1.9)	11.3 (0.0)
	MedQA	18.3 (–0.2)	5.8 (+1.5)
	SciQ	62.6 (0.0)	40.2 (+2.1)
OOD	RefuNQ	17.5 (–4.0)	5.4 (-0.3)
	SelfAware	27.0 (–3.9)	11.1 (+1.2)

Table 8: Performance of SEAL compared to KWT on in-domain (ID) and out-of-domain (OOD) datasets. Values in parentheses indicate differences from KWT (LLM).

Dataset	Model	nAUPC ↑	A-FPR ↓	IDK Precision ↑
HaluEval	KWT-RF	9.2 (-2.1)	25.5 (+14.1)	87.1 (–5.0)
	KWT-U	9.9 (-1.4)	22.3 (+10.9)	88.1 (–4.0)
MedQA	KWT-RF	2.1 (-2.2)	45.9 (+24.3)	86.5 (–4.7)
	KWT-U	2.6 (-1.7)	41.8 (+20.2)	86.5 (–4.7)
SciQ	KWT-RF	33.9 (-4.2)	11.1 (+7.4)	70.3 (–8.0)
	KWT-U	38.1 (0.0)	8.0 (+4.3)	71.9 (–6.4)

Table 9: Comparison of alternative weighting strategies (KWT-RF and KWT-U) relative to the KWT baseline.

and U weighting consistently reduce nAUPC. A similar trend is observed in Table 5, where SFT is often comparable to or slightly better than FT-TOP, suggesting that some low-knowledge instances still provide useful training signal. As shown in Table 3, a large fraction of training samples receive zero knowledge scores, and up-weighting such samples can improve answer accuracy in some cases.

However, RF and U consistently result in higher A-FPR and lower IDK Precision across datasets, indicating excessive uncertainty expression even when the model can answer correctly. Overall, these results suggest that emphasizing low-knowledge samples yields modest nAUPC gains at the cost of degraded uncertainty expression. In contrast, the familiarity (F) weighting strategy achieves a better balance between answer accuracy and well-calibrated uncertainty expression.

5 Analysis of IDK expression in KWT

5.1 Effect of External Knowledge on IDK expression

We investigate how the model's IDK expression changes when external knowledge is explicitly provided at inference time. When relevant knowledge is provided in the prompt, the model should be able to answer questions that would otherwise be unanswerable based solely on its parametric knowledge. To incorporate the external knowledge, we modify the test-time prompts following the format described in Appendix A.1.

Dataset	Model	nAUPC ↑	A-FPR ↓	IDK Precision ↑
HaluEval	KWT + k	57.8 (+46.5)	3.5 (–7.9)	68.6 (–23.5)
SciQ	KWT + k	62.9 (+24.8)	1.4 (–2.3)	81.4 (+3.1)

Table 10: Effect of knowledge-based prompting on KWT. Here, k denotes external knowledge provided in the input.

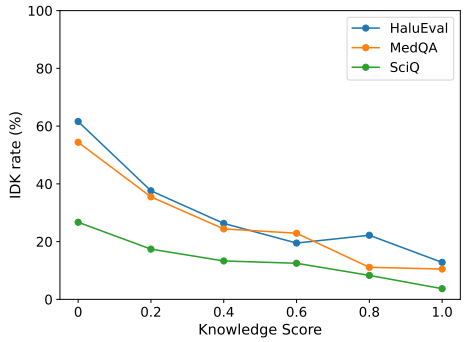


Figure 1: Relationship between instance-level knowledge scores and the frequency of <IDK> responses on the test dataset.

Table 10 presents the results when knowledge is added to the input. The results show a substantial increase in nAUPC, along with a sharp decrease in A-FPR. IDK Precision drops on HaluEval but improves on SciQ. Since KWT is trained only on SFT data, it may not fully understand or follow instructions describing the provided knowledge. This demonstrates that KWT leverages not only its internal knowledge but also externally provided information when generating responses, leading to improved accuracy and more informed decisions on whether to output <IDK>. Nevertheless, the model exhibits large shifts in <IDK> production once the knowledge is included in the prompt.

5.2 Generalization of IDK expression

Figure 1 shows the relationship between instance-level knowledge scores and the frequency of <IDK> responses on the test dataset. For each test instance, we estimate a knowledge score using the same procedure described in Section 3.1, and group instances according to their scores. For each group, we report the proportion of responses that include the <IDK> token.

As the knowledge score increases, the frequency of <IDK> responses consistently decreases. This indicates that KWT, which is trained to express uncertainty on low-knowledge training instances,

Backbone	Method	HaluEval	MedQA	SciQ	Avg
Llama3.2-3B	SFT	**31.0**	20.4	62.7	38.0
	KWT (non-IDK)	30.5	**21.1**	**65.9**	**39.2**
	KWT	29.8	18.5	62.6	37.0
Qwen2.5-3B	SFT	29.5	19.3	70.7	39.8
	KWT (non-IDK)	**30.6**	**21.3**	71.8	**41.2**
	KWT	29.8	17.8	**72.4**	40.0

Table 11: Standard accuracy of SFT, KWT, and KWT (non-IDK). Removing <IDK> supervision while keeping familiarity-based weighting improves average accuracy across both backbones.

Model	HaluEval	MedQA	SciQ	Avg
SFT	3.54	3.06	5.14	3.91
SEAL	3.71	3.20	**5.11**	4.01
KWT	**3.30**	**2.56**	5.20	**3.69**

Table 12: Token-level KL divergence between the base model and each fine-tuned model. Lower values indicate closer alignment to the base model's predictive distribution.

exhibits the same behavior on previously unseen test data. The monotonic relationship indicates that the IDK expression induced during fine-tuning is consistently expressed on the test set according to the base model's knowledge strength.

6 Impact of <IDK> Supervision on Accuracy Trade-off

To analyze the impact of <IDK> supervision on standard accuracy, we conduct an ablation study in which <IDK> is removed from the target responses while retaining the familiarity-based weighting strategy (denoted as KWT (non-IDK)). We compare this setting with standard SFT and the full KWT model, as summarized in Table 11.

Across both backbones, removing <IDK> supervision consistently improves standard accuracy under the same weighting scheme. In particular, KWT (non-IDK) achieves higher average accuracy than both SFT and KWT. This improvement suggests that familiarity-based weighting suppresses low-knowledge or high-conflict instances, allowing the model to focus on well-aligned examples and thereby improving standard accuracy compared to uniform SFT. These results indicate that the primary source of accuracy degradation is not the instance-level weighting strategy itself, but the supervision involving the <IDK> token, which introduces a trade-off between calibrated uncertainty expression and standard accuracy. We suggest first training the model with SFT and then applying KWT to better balance accuracy and uncertainty, and leave this for future work.

6.1 Predictive Distribution Shift

We analyze how SFT, KWT, and SEAL shift the predictive distribution away from the base model by measuring the token-level KL divergence for each question–answer pair. Specifically, for each question–answer pair, we compute the KL divergence as follows:

$$\mathrm{KL}(M_{\text{base}} \,\|\, M_{\text{trained}}) = \frac{1}{T}\sum_{t=1}^{T}\sum_{x} P_{\text{base}}(x \mid x_{<t}) \log \frac{P_{\text{base}}(x \mid x_{<t})}{P_{\text{trained}}(x \mid x_{<t})}. \quad (11)$$

Table 12 reports the results averaged across all samples. On average, KWT achieves lower KL divergence than SFT, indicating that it preserves the base model's distribution more faithfully during fine-tuning. This suggests that KWT avoids distorting the base model's predictive behavior, even while improving uncertainty expression. SEAL, in contrast, exhibits higher KL divergence than SFT, reflecting its token-level probability reallocation mechanism, which introduces larger deviations from the base distribution.

SRFT (Fu et al., 2025) shows that large deviations from the base policy can destabilize training, highlighting the importance of maintaining proximity to the base distribution. Accordingly, KWT preserves the base model's predictive distribution more effectively than both SFT and SEAL, helping maintain the model's underlying knowledge while enabling more stable fine-tuning.

7 Conclusion

We show that accurately estimating how well a model knows each training instance helps mitigate knowledge misalignment between pre-training and supervised fine-tuning. Using fine-grained knowledge scores obtained via multi-sampled inference, Knowledge-Weighted Fine-Tuning (KWT) weights the training signal to reinforce known examples while encouraging <IDK> responses on unknown ones. This improves uncertainty expression without hurting accuracy on answerable questions, reducing hallucinations in both in-domain and out-of-domain settings. We also introduce nAUPC, an uncertainty-aware metric that jointly evaluates accuracy and abstention behavior. Overall, aligning fine-tuning with a model's fine-grained knowledge offers a simple and generalizable approach to reducing hallucinations.

Limitations

Our approach relies on multi-sampled inference with multiple few-shot configurations to estimate instance-level knowledge scores, which requires performing a large number of inference runs per training instance. This inevitably increases computational cost compared to standard fine-tuning. However, this overhead is not unique to our method and is shared by many uncertainty- or consistency-based approaches that depend on repeated inference or sampling. Reducing the number of samples or developing more efficient approximations for knowledge estimation remains an important direction for future work.

References

Xinyun Chen, Renat Aksitov, Uri Alon, Jie Ren, Kefan Xiao, Pengcheng Yin, Sushant Prakash, Charles Sutton, Xuezhi Wang, and Denny Zhou. 2023. Universal self-consistency for large language model generation.

Yung-Sung Chuang, Yujia Xie, Hongyin Luo, Yoon Kim, James R. Glass, and Pengcheng He. 2024. Dola: Decoding by contrasting layers improves factuality in large language models. In *The Twelfth International Conference on Learning Representations*.

Roi Cohen, Konstantin Dobler, Eden Biran, and Gerard de Melo. 2024. I don't know: Explicit modeling of uncertainty with an [idk] token. In *Advances in Neural Information Processing Systems*, volume 37, pages 10935–10958. Curran Associates, Inc.

Yuqian Fu, Tinghong Chen, Jiajun Chai, Xihuai Wang, Songjun Tu, Guojun Yin, Wei Lin, Qichao Zhang, Yuanheng Zhu, and Dongbin Zhao. 2025. Srft: A single-stage method with supervised and reinforcement fine-tuning for reasoning.

Zorik Gekhman, Gal Yona, Roee Aharoni, Matan Eyal, Amir Feder, Roi Reichart, and Jonathan Herzig. 2024. Does fine-tuning LLMs on new knowledge encourage hallucinations? In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 7765–7784, Miami, Florida, USA. Association for Computational Linguistics.

Gemma Team and Google DeepMind. 2025. Gemma 3 technical report.

Gaurav Ghosal, Tatsunori Hashimoto, and Aditi Raghunathan. 2024. Understanding finetuning for factual knowledge extraction. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org.

Aaron Grattafiori et al. 2024. The llama 3 herd of models.

Lei Huang, Xiaocheng Feng, Weitao Ma, Yuchun Fan, Xiachong Feng, Yuxuan Gu, Yangfan Ye, Liang Zhao, Weihong Zhong, Baoxin Wang, Dayong Wu, Guoping Hu, Lingpeng Kong, Tong Xiao, Ting Liu, and Bing Qin. 2025. Alleviating hallucinations from knowledge misalignment in large language models via selective abstention learning. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 24564–24579, Vienna, Austria. Association for Computational Linguistics.

Di Jin, Eileen Pan, Nassim Oufattole, Wei-Hung Weng, Hanyi Fang, and Peter Szolovits. 2021. What disease does this patient have? a large-scale open domain question answering dataset from medical exams. *Applied Sciences*, 11(14):6421.

Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, Scott Johnston, Sheer El-Showk, Andy Jones, Nelson Elhage, Tristan Hume, Anna Chen, Yuntao Bai, Sam Bowman, Stanislav Fort, Deep Ganguli, Danny Hernandez, Josh Jacobson, Jackson Kernion, Shauna Kravec, Liane Lovitt, Kamal Ndousse, Catherine Olsson, Sam Ringer, Dario Amodei, Tom Brown, Jack Clark, Nicholas Joseph, Ben Mann, Sam McCandlish, Chris Olah, and Jared Kaplan. 2022. Language models (mostly) know what they know.

Adam Tauman Kalai, Ofir Nachum, Santosh S. Vempala, and Edwin Zhang. 2025. Why language models hallucinate.

Katie Kang, Eric Wallace, Claire Tomlin, Aviral Kumar, and Sergey Levine. 2025. Unfamiliar finetuning examples control how language models hallucinate. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 3600–3612, Albuquerque, New Mexico. Association for Computational Linguistics.

Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. 2023. Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation. In *The Eleventh International Conference on Learning Representations*.

Joosung Lee, Cheonbok Park, Hwiyeol Jo, Jeonghoon Kim, Joonsuk Park, and Kang Min Yoo. 2025. Enhancing hallucination detection via future context.

Jiaqi Li, Yixuan Tang, and Yi Yang. 2025. Know the unknown: An uncertainty-sensitive method for LLM instruction tuning. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 2972–2989, Vienna, Austria. Association for Computational Linguistics.

Junyi Li, Xiaoxue Cheng, Xin Zhao, Jian-Yun Nie, and Ji-Rong Wen. 2023a. HaluEval: A large-scale hallucination evaluation benchmark for large language

models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 6449–6464, Singapore. Association for Computational Linguistics.

Xiang Lisa Li, Ari Holtzman, Daniel Fried, Percy Liang, Jason Eisner, Tatsunori Hashimoto, Luke Zettlemoyer, and Mike Lewis. 2023b. Contrastive decoding: Open-ended text generation as optimization. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12286–12312, Toronto, Canada. Association for Computational Linguistics.

Genglin Liu, Xingyao Wang, Lifan Yuan, Yangyi Chen, and Hao Peng. 2024. Examining llms' uncertainty expression towards questions outside parametric knowledge.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. 2023. Self-refine: Iterative refinement with self-feedback. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Niels Mündler, Jingxuan He, Slobodan Jenko, and Martin Vechev. 2024. Self-contradictory hallucinations of large language models: Evaluation, detection and mitigation. In *The Twelfth International Conference on Learning Representations*.

Qwen Team. 2025. Qwen2.5 technical report.

Mengjie Ren, Boxi Cao, Hongyu Lin, Cao Liu, Xianpei Han, Ke Zeng, Wan Guanglu, Xunliang Cai, and Le Sun. 2024. Learning or self-aligning? rethinking instruction fine-tuning. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6090–6105, Bangkok, Thailand. Association for Computational Linguistics.

Weijia Shi, Xiaochuang Han, Mike Lewis, Yulia Tsvetkov, Luke Zettlemoyer, and Wen-tau Yih. 2024. Trusting your evidence: Hallucinate less with context-aware decoding. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers)*, pages 783–791, Mexico City, Mexico. Association for Computational Linguistics.

Katherine Tian, Eric Mitchell, Huaxiu Yao, Christopher D Manning, and Chelsea Finn. 2024. Fine-tuning language models for factuality. In *The Twelfth International Conference on Learning Representations*.

Ante Wang, Linfeng Song, Baolin Peng, Lifeng Jin, Ye Tian, Haitao Mi, Jinsong Su, and Dong Yu. 2024a. Improving LLM generations via fine-grained self-endorsement. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 8424–8436, Bangkok, Thailand. Association for Computational Linguistics.

Xinglin Wang, Yiwei Li, Shaoxiong Feng, Peiwen Yuan, Boyuan Pan, Heda Wang, Yao Hu, and Kan Li. 2024b. Integrate the essence and eliminate the dross: Fine-grained self-consistency for free-form language generation. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11782–11794, Bangkok, Thailand. Association for Computational Linguistics.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations*.

Johannes Welbl, Nelson F. Liu, and Matt Gardner. 2017. Crowdsourcing multiple choice science questions. In *Proceedings of the 3rd Workshop on Noisy User-generated Text*, pages 94–106, Copenhagen, Denmark. Association for Computational Linguistics.

Yuqing Yang, Ethan Chern, Xipeng Qiu, Graham Neubig, and Pengfei Liu. 2024. Alignment for honesty. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Zhangyue Yin, Qiushi Sun, Qipeng Guo, Jiawen Wu, Xipeng Qiu, and Xuanjing Huang. 2023. Do large language models know what they don't know? In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 8653–8665, Toronto, Canada. Association for Computational Linguistics.

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 7113–7139, Mexico City, Mexico. Association for Computational Linguistics.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in neural information processing systems*, 36:46595–46623.

Qihuang Zhong, Liang Ding, Xiantao Cai, Juhua Liu, Bo Du, and Dacheng Tao. 2025. KaFT: Knowledge-aware fine-tuning for boosting LLMs' domain-specific question-answering performance. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 24085–24100, Vienna, Austria. Association for Computational Linguistics.

A Prompt Templates

A.1 Input Format

During training, the input format is as follows:

Training format

Question: `{question}`
Answer: `{response}`

Here, the response is set to the gold answer if the knowledge score is greater than zero, and to the gold answer followed by <IDK> if the knowledge score is zero. As described in Section 5.1, when external knowledge is provided at inference time, the prompt is modified by adding a Knowledge: knowledge field before the question.

A.2 LLM-as-a-Judge Evaluation Prompt

To compute the knowledge score, we must determine whether each model-generated response matches the gold answer. One approach is to use an LLM-as-a-judge framework, which evaluates correctness using the following prompt:

Matching Prompt

Question: `{question}`
Knowledge: `{knowledge}`
Answer1: `{gold answer}`
Answer2: `{model answer}`
Are Answer1 and Answer2 semantically equivalent?
Answer only 'yes' or 'no':

Here, the `Knowledge` field is included only when the dataset provides an associated supporting knowledge paragraph; otherwise, it is omitted from the prompt. This method is also used for evaluating the model performance.

B IDK Ratio

IDK Score. The OOD evaluation datasets provide explicit labels indicating whether each question is answerable or unanswerable. Since unanswerable questions should ideally trigger <IDK> response, while answerable questions should not, this metric combines the IDK Rate (IR) on both subsets. It assigns higher scores to models that produce fewer <IDK> on answerable queries and more <IDK> on unanswerable ones.

$$\text{IR} = \frac{100 \times (A + B)}{A + B + C + D} \tag{12}$$

$$\text{IDK Score} = \frac{(100 - \text{IR}_{\text{ans}}) + \text{IR}_{\text{unans}}}{2} \tag{13}$$

Table 13 shows the proportion of <IDK> responses. KWT achieves the highest IDK score on unanswerable questions in both NEC and RefuNQ, indicating more appropriate uncertainty-aware behavior. In contrast, R-Tuning consistently produces very high <IDK> ratios, even on answerable questions, showing a tendency to over-refuse rather than distinguish between answerable and unanswerable inputs. This pattern remains the same in out-of-domain evaluations, indicating that precise knowledge scoring is crucial for improving both accuracy and uncertainty-aware behavior across settings.

C Experiments with Qwen 2.5

Table 14 reports the results using Qwen 2.5-3B as the base model (M_{base}), and the overall trends are consistent with our main experiments. Across all datasets, KWT (LLM) achieves the highest nAUPC while substantially reducing A-FPR and improving IDK Precision compared to all baselines. Although KWT (LLM) attains slightly lower standard accuracy than R-Tuning, it remains comparable to SFT and other KWT variants, indicating that improved uncertainty expression is achieved without severe accuracy degradation. In contrast, SEAL exhibits perfect A-FPR and IDK Precision due to deterministic token-level abstention, but suffers from substantial accuracy degradation across datasets.

D Variation of Response

We adopt the append strategy as our default approach, where an <IDK> token is added at the end of the response when the knowledge score is zero. Table 16 reports the results for alternative strategies, where <IDK> is either prepended to the response or used to fully replace the original answer.

KWT (append-IDK) outperforms KWT (prepend-IDK) across all metrics. This suggests that allowing the model to first produce an answer and then express uncertainty yields more stable behavior. KWT (only-IDK), which replaces the entire response with <IDK>, behaves similarly to SEAL in that any output containing <IDK> is counted as incorrect. As a result, it achieves perfect A-FPR and IDK Precision but suffers from substantial accuracy degradation.

E Timing of IDK Expression

We adopt Append-IDK as the default strategy, where the <IDK> token is appended to the end

Model	NEC			RefuNQ			SelfAware		
	IR_{ans} ↓	IR_{unans} ↑	IDK Score ↑	IR_{ans} ↓	IR_{unans} ↑	IDK Score ↑	IR_{ans} ↓	IR_{unans} ↑	IDK Score ↑
R-Tuning	94.45	100	52.78	87.82	95.44	53.81	86.95	77.71	**45.38**
KWT (EM)	79.92	81.67	50.88	68.62	81.96	56.67	68.46	43.02	37.28
KWT (Rouge)	38.18	49.76	55.79	50.57	69.86	59.65	39.54	21.22	40.84
KWT (LLM)	50.05	67.13	**58.54**	47.13	66.77	**59.82**	41.98	16.18	37.10

Table 13: Comparison of IDK ratios on the out-of-domain datasets: NEC, RefuNQ, and SelfAware.

Dataset	Model	Acc ↑	nAUPC ↑	A-FPR ↓	IDK Precision ↑
HaluEval	R-Tuning	30.4	5.7	38.3	83.5
	SEAL	27.8	10.6	0.0†	100.0†
	KWT (EM)	**30.9**	9.1	25.6	87.0
	KWT (Rouge)	**30.9**	10.9	14.9	89.5
	KWT-RF (LLM)	30.1	11.0	17.0	90.7
	KWT-U (LLM)	30.1	10.9	15.6	90.5
	KWT (LLM)	29.8	**11.3**	**6.5**	**93.6**
MedQA	R-Tuning	**20.0**	-2.4	78.0	82.5
	SEAL	17.7	**5.7**	0.0†	100.0†
	KWT (EM)	18.7	-0.6	66.0	83.9
	KWT (Rouge)	16.9	1.6	46.5	88.3
	KWT-RF (LLM)	20.6	1.9	47.7	86.4
	KWT-U (LLM)	19.2	2.4	41.8	87.8
	KWT (LLM)	17.8	3.9	**24.7**	**91.2**
SciQ	R-Tuning	**73.5**	22.2	32.4	45.8
	SEAL	70.7	51.2	0.0†	100.0†
	KWT (EM)	70.6	32.5	19.4	55.1
	KWT (Rouge)	70.8	41.4	10.7	63.6
	KWT-RF (LLM)	69.4	42.3	8.7	69.1
	KWT-U (LLM)	70.8	46.3	6.4	72.6
	KWT (LLM)	72.4	**51.3**	**2.8**	**78.3**

Table 14: Comparison of uncertainty-aware behavior on the in-domain datasets using the Qwen 2.5 model. † indicates a deterministic outcome induced by token-level abstention, where any output containing <IDK> is counted as incorrect.

of the response. As alternative designs, we consider Prepend-IDK, which places the token at the beginning of the response, and Only-IDK, which replaces the entire answer with the <IDK> token. Among these variants, Append-IDK consistently achieves better overall performance. A detailed comparison of these response variants is provided in Appendix D.

Figure 2 shows that the generation of <IDK> reflects its placement during training. With Append-IDK, the probability of generating <IDK> remains near zero until the final stage, indicating that uncertainty is expressed only after the model has processed most of the response. In contrast, Prepend-IDK concentrates <IDK> probability at the beginning of generation, resulting in an early and irreversible commitment to uncertainty expression. This delayed decision is consistent with prior findings showing that later parts of a generation can reveal hallucinations, as erroneous content often persists into future context (Lee et al., 2025).

F Additional Analysis on Hallucination

To directly analyze hallucination behavior, we measure how often the model generates <IDK> among incorrect responses using the following recall metric:

$$\text{IDK-Recall} = \frac{B}{B + D} \tag{14}$$

where B denotes incorrect answers that include <IDK>, and D denotes incorrect answers without <IDK>. A higher value indicates that the model more frequently expresses uncertainty when it produces an incorrect response.

R-Tuning achieves the highest recall; however, when considered with A-FPR and IDK-Precision,

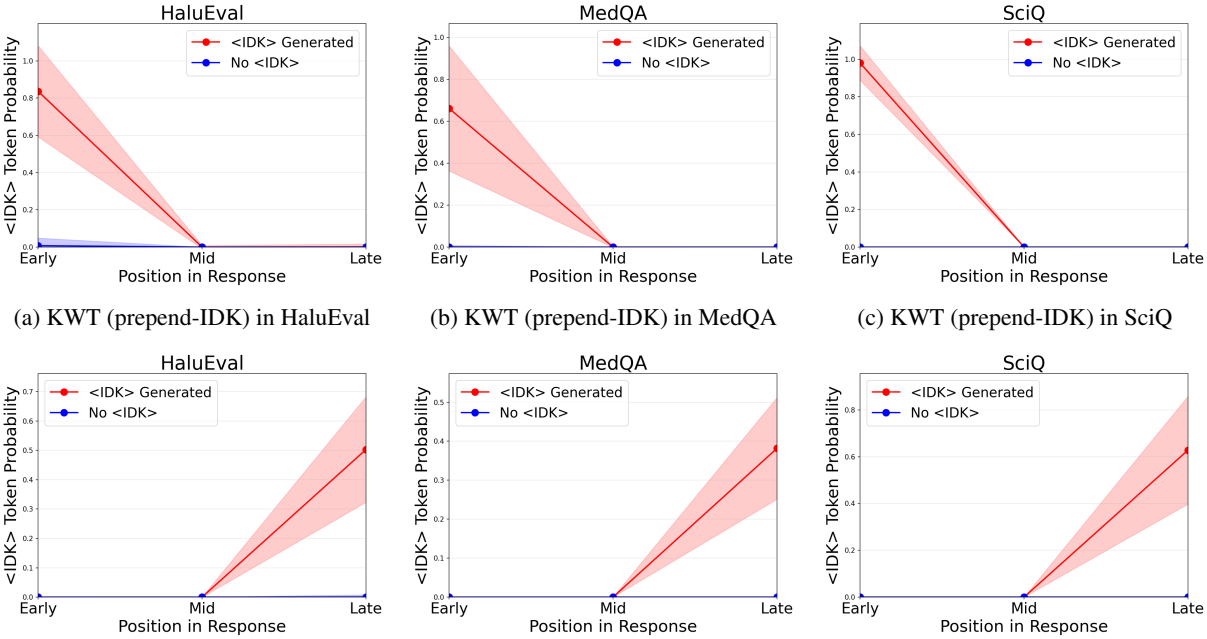
Model	HaluEval	MedQA	SciQ
R-Tuning	**89.5**	**89.6**	**73.7**
SEAL	35.0	33.8	12.4
KWT (LLM)	56.1	51.1	22.2

Table 15: Hallucination recall ($B/(B + D)$) across datasets.

this largely reflects excessive <IDK> generation rather than well-calibrated uncertainty. This shows that recall alone does not necessarily indicate effective hallucination mitigation. Since metrics such as FPR or recall in isolation are insufficient for selecting the optimal model, we argue that nAUPC—capturing the trade-off between uncertainty-aware abstention and certainty preservation—provides a more comprehensive evaluation criterion.

Dataset	Model	Acc ↑	nAUPC ↑	A-FPR ↓	IDK Precision ↑
HaluEval	KWT (prepend-IDK)	29.7	8.3	26.8	87.2
	KWT (only-IDK)	21.2	11.2	0.0†	100.0†
	KWT (append-IDK)	**29.8**	**11.3**	**11.4**	**92.1**
MedQA	KWT (prepend-IDK)	16.3	1.7	45.2	89.6
	KWT (only-IDK)	7.5	3.6	0.0†	100.0†
	KWT (append-IDK)	**18.5**	**4.3**	**21.6**	**91.2**
SciQ	KWT (prepend-IDK)	**62.8**	29.8	15.6	65.6
	KWT (only-IDK)	52.7	35.3	0.0†	100.0†
	KWT (append-IDK)	62.6	**38.1**	**3.7**	**78.3**

Table 16: Comparison of KWT variants across datasets with both standard accuracy and nAUPC. KWT (prepend-IDK) prepends the <IDK> token to the response, KWT (only-IDK) replaces the entire response with <IDK>, and KWT (append-IDK) appends <IDK> to the end of the response, which is the default setting in our experiments. † indicates a deterministic outcome induced by token-level abstention, where any output containing <IDK> is counted as incorrect.



(a) KWT (prepend-IDK) in HaluEval (b) KWT (prepend-IDK) in MedQA (c) KWT (prepend-IDK) in SciQ

(d) KWT (append-IDK) in HaluEval (e) KWT (append-IDK) in MedQA (f) KWT (append-IDK) in SciQ

Figure 2: Token-level probability of generating the <IDK> token at each relative position in the response under KWT. The top row corresponds to the prepend-IDK setting, while the bottom row shows the append-IDK setting.