RETHINKING PERPLEXITY WITH TRANSFERABLE SEMANTIC REWEIGHTING

Anonymous authors
Paper under double-blind review

ABSTRACT

Perplexity provides an efficient likelihood-based signal for assessing how strongly a language model supports a target generation, but it uniformly aggregates token likelihoods despite large differences in their predictability and semantic importance. Consequently, variation driven by lexical frequency, surface realization, or model-specific regularities can obscure likelihood changes associated with meaningful contextual evidence. We introduce semantic reweighting of contextual token likelihoods, which learns non-negative weights over token-level negative log-likelihoods from contextual representations, allowing the resulting score to emphasize more informative parts of generations. Across nine model–dataset settings, our method reduces average disagreement with document relevance annotations from 15.69% for standard perplexity to 3.55%. Moreover, weighting functions transferred across models and datasets retain 91.72% of the improvement achieved by weighting, suggesting that the informativeness of weighting exhibits transferable structure. These results show that a pre-trained contextual reweighting model can make perplexity more sensitive to semantic evidence while retaining its computational efficiency, with strong cross-domain generalization and simple plug-and-play deployment.

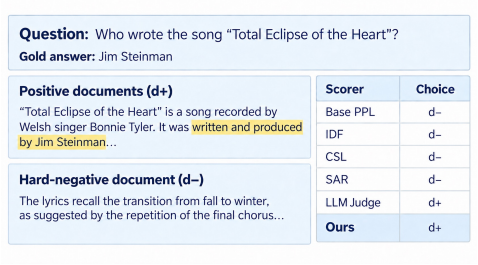
1 INTRODUCTION

Perplexity (PPL) provides a simple and efficient likelihood based signal for assessing how strongly an autoregressive language model supports a target generation. It can be computed directly from token negative log likelihoods without additional model calls or sampled generations, making it attractive for uncertainty estimation, generation evaluation, and retrieval augmented generation (Huang et al., 2024; Shorinwa et al., 2025; Dai et al., 2025). Standard PPL, however, uniformly aggregates token likelihoods, even though likelihood variations are known to be affected by lexical frequency, surface realization, local linguistic structure, and model specific regularities that need not reflect semantic support. Prior work has shown that perplexity is sensitive to token frequency (Dudy & Bedrick, 2020), that language models favor high marginal probability tokens under uncertainty (van der Poel et al., 2022), and that tokens contribute unequally to sequence level meaning and confidence (Duan et al., 2024; Bakman et al., 2024). Uniform aggregation can therefore obscure likelihood changes associated with meaningful contextual evidence.

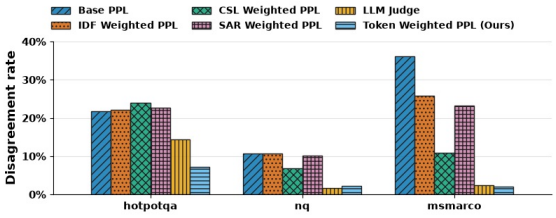
We observe this limitation directly when PPL measures contextual support. Given a question, a fixed target answer, and two contexts (one containing useful evidence and the other irrelevant information), human evaluators and LLM based judges consistently prefer the useful context, while standard PPL can favor the irrelevant one. Because we keep the target answer fixed, the disagreement reflects how PPL scores the answer rather than differences in generated content. Such failures undermine the reliability of PPL as a practical signal for uncertainty estimation and context quality assessment.

The above observation also aligns with recent work that improves likelihood based evaluation by reweighting token contributions. TokenSAR and MARS weight tokens according to their semantic importance (Duan et al., 2024; Bakman et al., 2024), while Contextualized Sequence Likelihood (CSL) derives context dependent weights from model attention (Lin et al., 2024). However, existing methods derive token weights from semantic importance or attention without directly learning which likelihoods distinguish supporting from irrelevant context. Consequently, they fail to resolve the disagreement between likelihood based evaluation and human judgment of context quality.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.



(a) Example from NQ



(b) Baseline and our scorers disagreement rates

Figure 1: (a) shows a question, its gold answer, supporting and irrelevant contexts, and the context selected by each scorer. (b) Disagreement with document relevance annotations on 10,000 document pairs per dataset (lower is better). Our context-sensitive token weighting scorer consistently outperforms base PPL and the IDF, CSL, and SAR weighted PPL baselines across HotpotQA, NQ, and MS MARCO. It achieves performance comparable to the LLM-as-judge baseline, with lower disagreement on HotpotQA and similar disagreement rate on NQ and MS MARCO.

In this paper, we ask which token likelihoods are informative for distinguishing useful contextual evidence. We introduce semantic reweighting of contextual token likelihoods, which learns a non-negative token weight from the contextual representation available when token is predicted. We train these weights directly from pairs of supporting and irrelevant contexts while keeping the underlying token likelihoods fixed. The resulting score learns which likelihood contributions are most useful for assessing contextual support rather than aggregating them uniformly. Across nine model-dataset settings, our method reduces disagreement with human annotations from 15.69% to 3.55%.

Remarkably, the learned weighting function is highly transferable. When transferred across models and datasets, it preserves 91.72% of the disagreement reduction achieved by weights trained directly in the target setting. Semantic sensitivity experiments further show that this behavior cannot be explained solely by lexical frequency or token regularities. Together, these results suggest that token likelihoods contain substantially more information about semantic evidence than standard PPL exposes, and that their informativeness exhibits transferable structure across models and tasks.

2 Background & Related Work

2.1 Perplexity and Answer Likelihood

Let P_θ be an autoregressive language model and let $a^* = (a_1^*, \ldots, a_n^*)$ be a gold answer to question q. When the model is conditioned on a document d, its length-normalized NLL and PPL are

$$\mathrm{PPL}(a^* \mid q, d) = \exp(\mathrm{NLL}(a^* \mid q, d)) = \exp\left(-\frac{1}{n}\sum_{i=1}^{n} \log P_\theta(a_i^* \mid q, d, a_{<i}^*)\right). \tag{1}$$

Thus, PPL is the exponential of average token surprisal and is a monotone transformation of length-normalized NLL. Lower PPL indicates that the model assigns a larger geometric-mean probability, and hence lower likelihood-derived uncertainty, to the reference answer (Malinin & Gales, 2021; Huang et al., 2024). PPL is attractive because it can be computed by teacher forcing in a single scoring pass, without sampling alternative generations or invoking an external evaluator.

Conditional answer likelihood also provides a task-oriented signal for retrieval. Retrieval-augmented models can train or rank retrievers according to how strongly a candidate document increases the likelihood of the gold answer (Izacard et al., 2023; Nian et al., 2025). Related work uses final-answer likelihood to select useful prefixes of multi-step retrieval chains (Wang et al., 2025). These approaches motivate comparing $\mathrm{PPL}(a \mid q, d)$ across candidate documents. However, absolute PPL is affected by token frequency, tokenization, the model's predictive distribution, and instance difficulty, so values obtained for substantially different questions or domains need not be directly comparable (Dudy & Bedrick, 2020; Wu et al., 2026). Our experiments therefore use PPL only as a within-instance ranking score: q, a, and P_θ are fixed, and only d changes. This controls cross-instance difficulty but does not remove unequal token-level contributions within the answer.

2.2 Semantic Aware Likelihood Scoring

One line of work moves from surface-form likelihood to uncertainty over meaning. Semantic entropy groups sampled generations that express the same meaning before aggregating their probability mass (Kuhn et al., 2023). SelfCheckGPT instead treats disagreement among stochastically sampled responses as evidence of hallucination (Manakul et al., 2023). In retrieval evaluation, SePer estimates the model's probability mass on generations that semantically entail the gold answer, both with and without retrieved evidence, and defines retrieval utility by the resulting change in semantic belief (Dai et al., 2025). RePPL combines uncertainty arising during semantic propagation with uncertainty across stochastic generations and produces token-level explanatory scores (Huang et al., 2026). These methods reduce sensitivity to paraphrasing and expose uncertainty that a single reference likelihood can miss, but generally require multiple generations and additional semantic, clustering, or entailment computations.

A complementary line retains token likelihoods while changing how token-level evidence is aggregated. Fixed corpus statistics provide simple, interpretable weights. Inverse document frequency (IDF) upweights rare, discriminative terms (Sparck Jones, 1972) and is used, for example, as an optional weighting scheme in BERTScore (Zhang et al., 2020). Although IDF and BERTScore were not introduced as uncertainty estimators, they illustrate how predefined frequency statistics can replace uniform token importance. Related analyses show that linguistic items are not equally informative under likelihood-based evaluation (Dudy & Bedrick, 2020); frequency aware token objectives have also been used to counter the underemphasis of rare, content-bearing tokens in neural machine translation (Gu et al., 2020). Likewise, mutual-information scoring contrasts conditional and marginal token probabilities to reduce a model's preference for high-probability generic tokens (van der Poel et al., 2022). Such functions correct a specified source of regularity, but their form is fixed before observing the downstream relevance task.

Closer to our token reweighting approach, CaMeLS (Hu et al., 2023) learns context-dependent weights over token-level NLL, but targets knowledge acquisition in out-of-date language models through online fine-tuning, addressing a different downstream task from our work.

2.3 LLM-as-Judge Evaluation

LLM-as-judge methods prompt a capable language model to score one response or compare several responses under natural-language evaluation criteria. G-Eval combines rubric-guided reasoning with form-based scoring and reports stronger correlation with human judgments than conventional reference-based NLG metrics (Liu et al., 2023). MT-Bench and Chatbot Arena further show that strong LLM judges can approximate human preferences in pairwise and multi-turn evaluation (Zheng et al., 2023). For document comparison, a judge can directly examine the question, gold answer, and candidate evidence, allowing it to assess semantic support without relying on the surface probability of each answer token. We therefore treat LLM-as-judge as a strong external baseline for document-relevance prediction.

This flexibility comes with different costs and failure modes from likelihood-based scoring. Judging requires an additional autoregressive inference, and its outcome can depend on the evaluator model, prompt, decoding procedure, and response order. Prior work documents position and self-preference effects as well as sensitivity to superficial or misleading features (Zheng et al., 2023; Chen et al., 2024). Large proprietary judges additionally introduce latency, monetary cost, and reproducibility concerns, motivating smaller learned evaluators (Li et al., 2026). Our objective is not to reproduce every capability of an LLM judge; rather, it is to narrow the gap between uniform PPL and relevance judgments while retaining the efficiency, determinism, and token-level decomposability of likelihood scoring.

3 Contextual Aware Weight Model Methodology

3.1 Task Definition

Let $\mathcal{X}$ be a collection of tuples $x = (q, a^*, d^+, d^-)$, where q is a question, $a^* = (a_1^*, \ldots, a_n^*)$ is its human-provided gold answer, d^+ is a document annotated as relevant to answering q, and d^- is a hard-negative document selected from the remaining candidates. We evaluate whether a likelihood

based score preserves this relevance ordering by assigning a lower conditional weighted perplexity to d^+ than to d^-, such that

$$\mathrm{PPL}_w(a^* \mid q, d^+) < \mathrm{PPL}_w(a^* \mid q, d^-),$$

where q, a^*, and P_θ remain fixed and only the conditioning document changes following (Dudy & Bedrick, 2020; Wu et al., 2026).

3.2 Limits of Predefined Likelihood Reweighting

Standard perplexity uniformly averages token-level negative log-likelihoods, although the likelihood assigned to a token may reflect factors other than its contribution to contextual utility (Dudy & Bedrick, 2020; Gu et al., 2020; Duan et al., 2024; Bakman et al., 2024). A natural way to reduce such effects is to reweight token likelihoods according to their estimated propensity, giving less influence to tokens whose probability is high for reasons unrelated to the conditioning evidence.

We use the weighted aggregation

$$\mathrm{PPL}_w(a^* \mid q, d) = \exp\left(-\frac{\sum_{i=1}^{n} w_i \log P_\theta\left(a_i^* \mid q, d, a_{<i}^*\right)}{\sum_{i=1}^{n} w_i}\right), \tag{2}$$

where $w_i \geq 0$ controls the contribution of token a_i^*. Standard perplexity is recovered when $w_i = 1$.

Bias correction. The source of bias in token likelihood is not known a priori. A token may receive high probability because it is frequent in the training distribution, because it is strongly preferred by the trained model independent of the current evidence, or because the model has already parameterized the corresponding knowledge. We therefore consider several weighting schemes, each corresponding to a different hypothesis about the source of this propensity. These schemes can be viewed as inverse-propensity-style corrections: tokens that are likely under a hypothesized background source receive less relative influence in the final likelihood score.

Table 1 shows that correcting these hypothesized sources of bias produces only limited or inconsistent improvement over standard PPL. This suggests that the disagreement cannot be attributed to a single observable propensity such as corpus frequency (InvLogFreq) or model prior. The token likelihood reflects several interacting effects, making the relevant source of bias difficult to specify in advance.

Semantic awareness. We further evaluate semantic likelihood-based methods that assign token contribution using contextual semantic information (Duan et al., 2024; Bakman et al., 2024; Dai et al., 2025). Table 1 shows that semantic weighting reduces disagreement more substantially than propensity-based correction, confirming that token contribution should depend on semantic context. A considerable gap nevertheless remains between likelihood-based evaluation and human annotations. These results suggest that the main limitation is not token reweighting itself, but the need to specify in advance what should be corrected. Corpus frequency, model prior, and predefined semantic criteria each capture only part of the factors underlying token likelihood. These results suggest that neither correcting hypothesized token-level bias nor incorporating predefined semantic importance is sufficient to explain the disagreement between PPL and human judgments.

3.3 Learning Context-Sensitive Token Weights

Rather than attributing the disagreement between PPL and contextual utility to token-generation bias or insufficient semantic awareness, we hypothesize that it also arises from how token likelihoods are aggregated under the given context, where the appropriate aggregation function is difficult to specify analytically. We therefore propose to learn a context-sensitive token weight

$$w_i = g_\phi(h_i), \tag{3}$$

where h_i denotes the hidden representation immediately preceding answer token a_i. The model computes h_i from the question q, document d, and answer prefix $a_{<i}^*$, so it captures the information available when predicting a_i. We use the learned weight to measure how informative each token-level likelihood is for distinguishing useful from non-useful conditioning context.

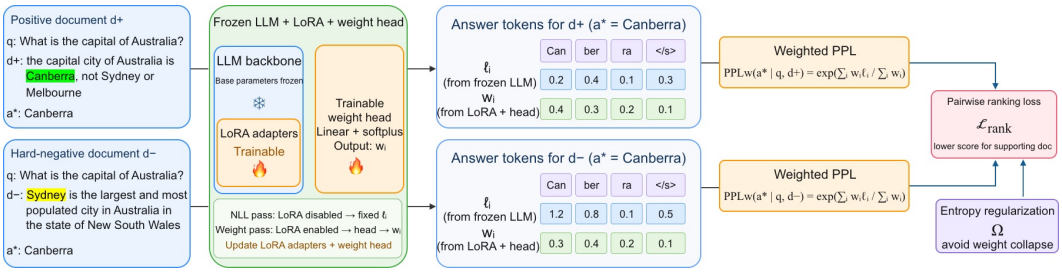


Figure 2: Overview of the context-sensitive token weighting framework. The weighting model uses contextual representations from the language model to estimate token contributions, which we combine with the fixed token-level NLLs and optimize using document-pair supervision.

We obtain h_i from the same LLM that produces the token likelihoods. This keeps the weighting function in the model's own representation space and lets it capture interactions among the question, document, answer prefix, and parameterized knowledge. Since the pretrained representation does not directly optimize contextual-utility weighting, we, therefore, fine-tune the model for the weighting task. This adaptation encourages the hidden representation to capture features that distinguish contextual support from background model preference.

We use LoRA (Hu et al., 2022) to adapt the contextual representations while keeping the pretrained backbone parameters frozen, and attach a linear token-level weight head. A linear projection followed by softplus produces a non-negative weight:

$$w_i = g_\phi(h_i) = \mathrm{softplus}\left(\mathbf{v}^\top h_i + b\right). \tag{4}$$

During training, we teacher-force the gold answer and take h_i from the position immediately before a_i in the LoRA-adapted model. We jointly optimize the LoRA parameters and the weight-head parameters $\mathbf{v}$ and b. Throughout training and evaluation, including weighted PPL computation, token NLLs are obtained from the original frozen base model without LoRA adaptation and are excluded from gradient computation.

Each training instance contains a supporting document d^+ and a hard-negative document d^- for the same question and gold answer. Following Eq. 2, we compute the weighted perplexity under document d as

$$\mathrm{PPL}_w(a^* \mid q, d) = \exp\left(\frac{\sum_{i=1}^{n} w_i^d \, \ell_i^d}{\sum_{i=1}^{n} w_i^d}\right), \qquad \ell_i^d = -\log P_\theta(a_i^* \mid q, d, a_{<i}^*). \tag{5}$$

Our supervision is pairwise by construction. In particular, for the same (q, a^*), the supporting document should receive lower weighted perplexity than the hard negative. Since the exponential transformation is monotonic, we optimize the corresponding weighted NLL directly. Let

$$\Delta = \frac{\sum_i w_i^{d^-} \ell_i^{d^-}}{\sum_i w_i^{d^-}} - \frac{\sum_i w_i^{d^+} \ell_i^{d^+}}{\sum_i w_i^{d^+}}. \tag{6}$$

We optimize

$$\mathcal{L}_{\mathrm{rank}} = \log\left(1 + \exp(-\beta\Delta)\right), \tag{7}$$

where $\beta > 0$ controls the sharpness of the loss. Minimizing Eq. 7 encourages

$$\mathrm{PPL}_w(a^* \mid q, d^+) < \mathrm{PPL}_w(a^* \mid q, d^-),$$

which directly matches the document-pair comparison used in our evaluation.

To avoid concentrating most of the weight on a small number of tokens, we use entropy regularization (Ahmed et al., 2019) that encourages a higher-entropy weight distribution among tokens in both documents.

$$\Omega = \frac{1}{2}\left[\sum_i \pi_i^{d^+} \log \pi_i^{d^+} + \sum_i \pi_i^{d^-} \log \pi_i^{d^-}\right], \qquad \text{where} \quad \pi_i^d = \frac{w_i^d}{\sum_j w_j^d}. \tag{8}$$

We combine this regularizer with the ranking objective,

$$\mathcal{L} = \mathcal{L}_{\mathrm{rank}} + \lambda\Omega, \tag{9}$$

where λ controls the trade-off between preserving the document-pair ranking signal and preventing the learned weighting from collapsing onto a few tokens.

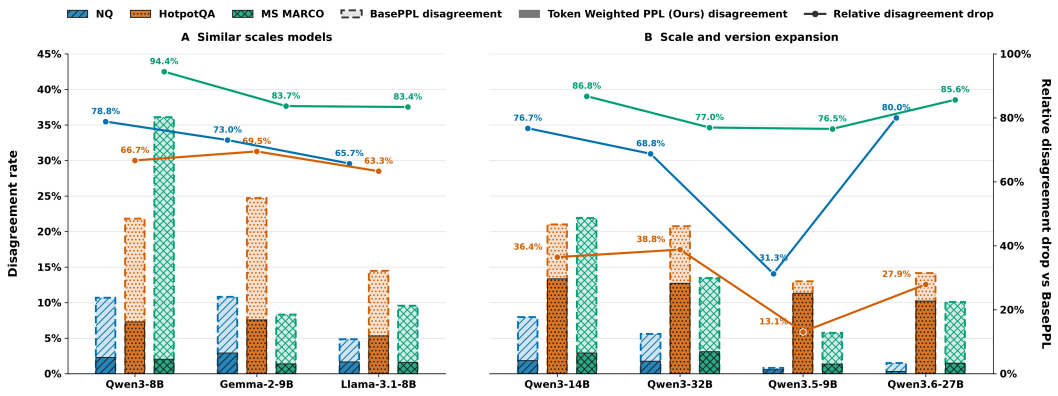


Figure 3: Disagreement with document relevance annotations for base PPL and learned weighted PPL across three datasets. Panel A compares models at same scale, while Panel B extends the evaluation to larger models and newer Qwen versions. Light bars with dashed outlines show base PPL disagreement, and darker bars show our methods disagreement (left axis; lower is better). Lines indicate the relative drop in disagreement compared with base PPL (right axis; higher is better). Learned weighting consistently reduces disagreement across all evaluated model–dataset settings.

4 Experiments

4.1 Experimental Setup

We evaluate both the effectiveness and transferability of the learned token-weighting function across different language models and QA datasets. We first construct a 3×3 training matrix using three base models—Qwen3-8B, Gemma-2-9B, and Llama-3.1-8B, and three datasets—HotpotQA, MS MARCO, and Natural Questions (NQ) (Bajaj et al., 2018; Yang et al., 2018; Kwiatkowski et al., 2019). For each mode–dataset combination, we independently train an embedding-to-weight model following the procedure described in Sec. 3.3, resulting in nine learned weighting models.

We further train four models from the Qwen family—Qwen3-14B, Qwen3-32B, Qwen3.5-9B, and Qwen3.6-27B—to evaluate if the benefits of learned weighting persist across model versions and parameter scales. These additional models are used only for native performance evaluation.

For both training and evaluation, we randomly sample 10,000 hold out question answer pairs (q, a) and document pairs (d^+, d^-) separately. Appendix B provides further details on the training settings. For evaluation, we check if a scoring method agrees with the relevance annotation, report the *disagreement rate* described in Sec. 3.1. Lower disagreement indicates better retrieval performance.

To evaluate transferability, let

$$S = (m_s, \mathcal{D}_s), \qquad T = (m_t, \mathcal{D}_t)$$

denote a source weighting model and a target evaluation setting, respectively. We consider three transfer conditions: *dataset transfer* ($m_s = m_t$, $\mathcal{D}_s \neq \mathcal{D}_t$), *model transfer* ($m_s \neq m_t$, $\mathcal{D}_s = \mathcal{D}_t$), and *joint transfer* ($m_s \neq m_t$, $\mathcal{D}_s \neq \mathcal{D}_t$). The native setting $S = T$ serves as the upper-reference performance for learned weighting. Across the nine target settings, this produces 18 dataset transfers, 18 model transfers, and 36 joint transfers, for 72 directed transfer evaluations in total.

Importantly, transferring a weighting model does not replace the likelihood distribution of the target LLM. For a source weighting model S evaluated on target T, we compute

$$L_{S \to T}(d) = \frac{\sum_i \widehat{w}_i^{S,d} \, \ell_i^{T,d}}{\sum_i \widehat{w}_i^{S,d}}, \tag{10}$$

where $\widehat{w}_i^{S,d}$ is the source model's predicted weight aligned to the target answer tokens, and $\ell_i^{T,d}$ is the original token NLL from target model m_t. Unlike a direct relevance head tied to the source backbone's hidden representations, the predicted token weights can be aligned across tokenizations and combined with target-model likelihoods, enabling transfer between models with different hidden

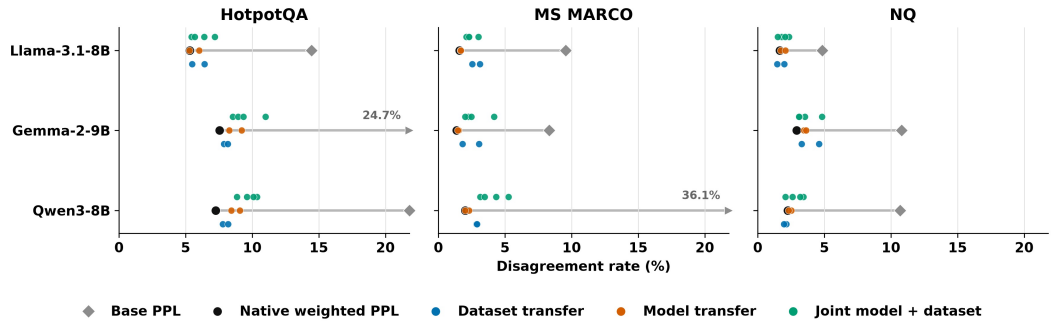


Figure 4: Base PPL, native weighting, and transferred weighting disagreement rates across nine target model-dataset settings. Lower/left is better; transferred weights retain most of the native weighting improvement under dataset, model, and joint model-dataset shifts.

dimensions. When source and target models use different tokenizations, source weight mass is projected onto target tokens according to overlapping answer-character spans. Thus, the transfer experiment specifically measures whether the learned token-importance pattern generalizes while preserving the target model's original likelihood signal.

We further quantify transferability by the retention fraction of the native weighting PPL (which weighting model trained on the target setting) disagreement drop rate retained under transfer:

$$R_{S\to T} = \frac{D_{B\to T} - D_{S\to T}}{D_{B\to T} - D_{T\to T}}, \tag{11}$$

where $D_{B\to T}$ is the base PPL disagreement, $D_{T\to T}$ is the native weighting PPL disagreement, and $D_{S\to T}$ is the disagreement after transferring weights from S, all at target dataset T. $R_{S\to T} = 1$ indicates that the transferred weighting retains the full native improvement, while $R_{S\to T} = 0$ indicates no improvement over base PPL.

4.2 Native Performance

Figure 3 compares native weighting PPL with base PPL across model families, versions, and parameter scales. Panel A compares three contemporary model families at a similar parameter scale: Qwen3-8B, Gemma-2-9B, and Llama-3.1-8B. Panel B extends this comparison within the Qwen family. Together with Qwen3-8B, these settings examine different model versions and parameter scales from 8B to 32B. Learned weighting improves over base PPL on all three datasets for every evaluated model, suggest that the benefit of learned weighting is not specific to a single backbone.

4.3 Dataset and Base Model Transferability

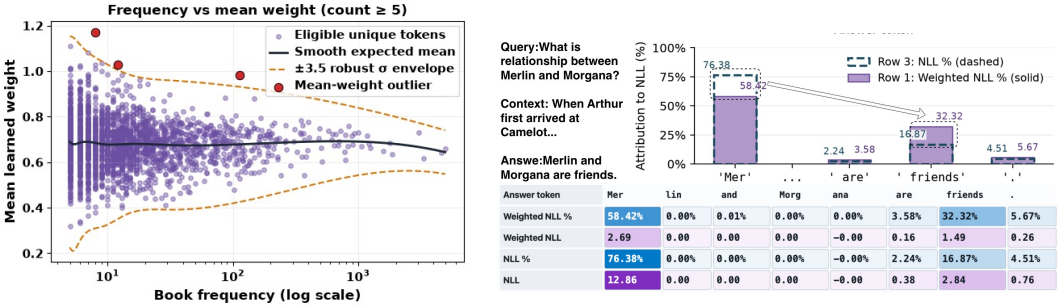
We next evaluate transferability under the protocol defined in Section 4.1. Dataset transfer changes the dataset used to train weight model while keeping the model fixed; model transfer does the reverse; and joint transfer changes both. Figure 4 presents all transfers among the nine model-dataset combinations, retention measures by the Eq. 11. The results show that most of the native improvement is preserved under all three transfer types. More detail analysis can be found in Appendix. A

4.4 Baseline Comparison

Table 1 compares learned weighted PPL with eight scoring baselines and an LLM judge using Qwen3-8B across three datasets. These include base PPL; three bias correction schemes and four semantic aware methods—-semantic entropy (Kuhn et al., 2023), SePer (Dai et al., 2025), SAR (Duan et al., 2024), and CSL (Lin et al., 2024). Semantic entropy measures uncertainty over response, while SePer estimates the increase in semantic belief in the gold answer after retrieval. TokenSAR and CSL reweight token likelihoods using semantic relevance and model attention, respectively. As gold answer usually only contain one sentence, leaving limited scope for sentence-level weighting,

Table 1: Disagreement rates between scores and human annotations. Random scorer should archive 0.5 disagreement. Predefined weighting under different hypothesized bias provides limited or inconsistent improvement over base PPL. Semantic aware methods yield larger improvements in several settings, while learned token weighting achieves the lowest disagreement among likelihood-based methods. Lower is better; best results are shown in bold.

Dataset	Base PPL	IDF Weighted	InvLogFreq Weighted	Model Prior Weighted	Semantic Entropy	SEPER	SAR Weighted	CSL Weighted	LLM Judge	Learned (Ours)
HotpotQA	0.218	0.222	0.238	0.280	0.327	0.225	0.226	0.240	0.143	**0.073**
NQ	0.107	0.108	0.114	0.130	0.210	0.120	0.101	0.068	**0.017**	**0.023**
MS MARCO	0.361	0.259	0.416	0.146	0.263	0.169	0.232	0.109	**0.024**	**0.020**



(a) Learned weight distribution by frequency

(b) Shift in token-level NLL contributions to perplexity

Figure 5: Analysis of learned token weights. (a) Mean learned weight as a function of token frequency in *Harry Potter and the Philosopher's Stone*, showing systematic frequency-dependent structure and only few outliers. (b) Example comparing each answer token's contribution to the perplexity under uniform and learned weighting. Weighting reduces the dominant contribution of the opening token and shifts attribution toward the later, semantically decisive token *friends*.

we use the token-only variant of SAR to evaluate token contributions. All methods requiring sampling use 10 responses per conditioning input, following SePer (Dai et al., 2025), for which we use the soft matching variant. For CSL, we select 10 attention heads using a separate calibration set of 1,000 question–answer pairs from each dataset, excluded from training process.

Predefined bias correction yields limited or inconsistent gains over base PPL: all three schemes perform similarly or worse on HotpotQA and NQ, although IDF and Model Prior improve substantially on MS MARCO. Semantic-aware baselines achieve stronger results in selected settings, with CSL reducing disagreement on NQ and on MS MARCO. Yet all remain behind the LLM judge across the three datasets. Our learned semantic-aware weighting consistently outperforms all eight scoring baselines. It also achieves lower disagreement than the LLM judge on HotpotQA (7.3% vs. 14.3%) and MS MARCO (2.0% vs. 2.4%), while yielding slightly higher disagreement on NQ (2.3% vs. 1.7%). These results support learning contextual token importance to improve likelihood-based document evaluation and bring its agreement with human annotations closer to that of LLM judgment.

4.5 Analysis of Learned Token Weights

For the frequency analysis in Figure 5(a), we use the weighting model trained with Qwen3-8B on HotpotQA. We tokenize *Harry Potter and the Philosopher's Stone* and divide it into windows of 10,000 tokens with an overlap of 1,000 tokens. We then apply the trained weighting model to each token and aggregate the resulting weights by token frequency. For each eligible token type, the figure reports its frequency in the book and its mean learned weight across occurrences. Figure 5(a) reveals two distinct patterns. First, high-frequency tokens, which often correspond to stop words or common subword units, receive relatively stable weights close to the mean. Second, low-frequency tokens show substantially greater variation in their learned weights, suggesting that their weights are more sensitive to context.

To quantify how token weighting responds to changes in contextual evidence, we randomly sample 1,000 NQ questions using the same evaluation setup as Table 1. We compare our learned weighting

Context: Every evening, Maya writes the next day's appointments on a large whiteboard beside the clinic entrance. The nurses use it to see which patients are expected and when each room will be occupied... If an appointment changes, Maya updates the board immediately, which helps prevent staff from preparing for the wrong patient.

Answer token	The	white	board	keeps	clinic	staff	updated	on	the	day	's	appointments
What is the whiteboard used for?												
Weighted NLL %	20.79%	0.00%	0.00%	32.28%	28.21%	0.00%	6.84%	0.71%	3.33%	3.19%	2.00%	0.04%
Weighted NLL	4.22	0.00	0.00	6.55	5.72	0.00	1.39	0.14	0.68	0.65	0.41	0.01
NLL %	36.23%	0.00%	0.00%	27.41%	19.38%	0.00%	6.06%	0.73%	2.22%	2.44%	1.80%	0.03%
NLL	19.66	0.00	0.00	14.88	10.52	0.00	3.29	0.40	1.20	1.32	0.97	0.02
Who does the whiteboard help keep informed?												
Weighted NLL %	18.27%	0.02%	0.00%	14.56%	29.76%	0.00%	23.81%	2.95%	2.21%	4.15%	2.59%	0.05%
Weighted NLL	4.99	0.01	0.00	3.98	8.13	0.00	6.50	0.81	0.60	1.13	0.71	0.01
NLL %	30.27%	0.02%	0.00%	13.72%	22.00%	0.00%	21.43%	3.28%	1.53%	3.09%	2.34%	0.04%
NLL	20.68	0.01	0.00	9.38	15.03	0.00	14.64	2.24	1.05	2.11	1.60	0.03

Figure 6: Token-level NLL attribution under different queries with a fixed context and answer. For both queries, learned weighting reduces the opening token's relative contribution and increases the contribution of later informative tokens, while the attribution pattern varies with the query.

with context-conditioned variants of SAR and CSL, the two strongest token-weighting baselines. For each method, we keep the question and gold answer fixed, normalize the answer-token weights $\tilde{w}_i$ to sum to one under each document, and compute their paired L1 distance $D = \sum_i |\tilde{w}_i^+ - \tilde{w}_i^-|$. This metric measures how much relative token-weight mass is redistributed when the conditioning document changes. Our method yields the largest mean distance, 0.393 ± 0.198, followed by CSL, 0.359 ± 0.207 and SAR, 0.245 ± 0.174. This ordering is consistent with their document-ranking performance on NQ in Table 1. These results show that learned weighting adjusts relative token importance more strongly across positive and hard-negative contexts, complementing the preceding qualitative example with quantitative evidence of context sensitivity. The alignment between larger L1 distance and the observed performance ordering supports the role of context-sensitive token weighting in effective document evaluation.

4.6 How Weighting Redistributes Token-Level NLL Contributions

Figure 5(b) use same model to evaluate fixed sample text. we further illustrates how learned weighting redistributes token level contributions to the uncertainty score. Under uniform weighting, the opening subword `Mer` dominate the total NLL, whereas *friends*, which expresses the relationship queried by the question, contributes only 16.87%. This imbalance illustrates that a token's surprisal does not necessarily reflect its semantic importance, consistent with the token-level generative inequality discussed by Duan et al. (2024). The concentration of NLL on the opening token also resembles the sharp decline in predictive uncertainty after the first generated token observed by Arteaga et al. (2025). Learned weighting reduces the opening token's contribution and increases that of *friends* to 32.32%. This example demonstrates that the weighting function can partially mitigate opening token dominance and assign greater relative contribution to a later, semantically decisive token, without modifying the underlying token probabilities. Figure 6 extends this analysis by varying the query while keeping the context and answer fixed. The contributions also change with the query: when the question shifts from "What is the whiteboard used for?" to "Who does the whiteboard help keep informed?", the share attributed to "keeps" falls, while that attributed to "clinic" and "updated" rises.

5 Conclusion

We introduced semantic reweighting of contextual token likelihoods, which learns token importance while preserving the original model probabilities. Across nine model–dataset settings, it reduces average disagreement with relevance annotations from 15.69% to 3.55%, retaining 91.72% of the native improvement under transfer. Our method also achieves lower disagreement than all evaluated likelihood-based baselines across the three datasets. These findings suggest that learned aggregation can better capture contextual evidence than uniform perplexity and existing token-weighted likelihood methods.

AI USE STATEMENT

In this work, we used generative AI tools for "aid or polish writing. Details are described in the paper" and "retrieval and discovery (e.g., finding related work)". Be more specific, we used ChatGPT to polish the grammar, phrasing, and readability of author-written text and to assist with literature discovery under author-specified search directions. For literature discovery, we provided research topics and experimental observations and requested potentially relevant publications, for example like studies that might explain opening-token dominance and query-dependent changes in token-level negative log-likelihood contributions. We also consulted ChatGPT for possible literature-based interpretations of these observations. We have not used generative AI tools for all other tasks with required disclosure.

We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

Section 3.3 specifies the token-weighting model and training objective, and Section 4.1 defines the evaluation and transfer protocols. Appendix B.1 documents dataset preparation, data splits, prompts, training hyperparameters, checkpoint selection, evaluation cohorts, and cross-model token alignment. Together, these details specify the procedures used to train the weighting models and reproduce the reported comparisons.

REFERENCES

Zafarali Ahmed, Nicolas Le Roux, Mohammad Norouzi, and Dale Schuurmans. Understanding the impact of entropy on policy optimization. In *International conference on machine learning*, pp. 151–160, 2019.

Gabriel Y. Arteaga, Thomas B. Schön, and Nicolas Pielawski. Hallucination detection in LLMs: Fast and memory-efficient finetuned models. In *Northern Lights Deep Learning Conference*, 2025.

Payal Bajaj, Daniel Campos, Nick Craswell, Li Deng, Jianfeng Gao, Xiaodong Liu, Rangan Majumder, Andrew McNamara, Bhaskar Mitra, Tri Nguyen, Mir Rosenberg, Xia Song, Alina Stoica, Saurabh Tiwary, and Tong Wang. Ms marco: A human generated machine reading comprehension dataset, 2018.

Yavuz Faruk Bakman, Duygu Nur Yaldiz, Baturalp Buyukates, Chenyang Tao, Dimitrios Dimitriadis, and Salman Avestimehr. MARS: Meaning-aware response scoring for uncertainty estimation in generative LLMs. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 7752–7767, 2024.

Guiming Hardy Chen, Shunian Chen, Ziche Liu, Feng Jiang, and Benyou Wang. Humans or LLMs as the judge? a study on judgement bias. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 8301–8327, 2024.

Lu Dai, Yijie Xu, Jinhui Ye, Hao Liu, and Hui Xiong. Seper: Measure retrieval utility through the lens of semantic perplexity reduction. In *The Thirteenth International Conference on Learning Representations*, 2025.

Jinhao Duan, Hao Cheng, Shiqi Wang, Alex Zavalny, Chenan Wang, Renjing Xu, Bhavya Kailkhura, and Kaidi Xu. Shifting attention to relevance: Towards the predictive uncertainty quantification of free-form large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 5050–5063, 2024.

Shiran Dudy and Steven Bedrick. Are some words worth more than others? In *Proceedings of the First Workshop on Evaluation and Comparison of NLP Systems*, pp. 131–142, 2020.

Shuhao Gu, Jinchao Zhang, Fandong Meng, Yang Feng, Wanying Xie, Jie Zhou, and Dong Yu. Token-level adaptive training for neural machine translation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1035–1046, 2020.

Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022.

Nathan Hu, Eric Mitchell, Christopher Manning, and Chelsea Finn. Meta-learning online adaptation of language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 4418–4432, 2023.

Xinmeng Huang, Shuo Li, Mengxin Yu, Matteo Sesia, Hamed Hassani, Insup Lee, Osbert Bastani, and Edgar Dobriban. Uncertainty in language models: Assessment through rank-calibration. In *EMNLP*, pp. 284–312, 2024.

Yiming Huang, Junyan Zhang, Zihao Wang, Biquan Bie, Yunzhong Qiu, Xuming Hu, Yi R. Fung, and Xinlei He. Reppl: Recalibrating perplexity by uncertainty in semantic propagation and language generation for explainable qa hallucination detection. *arXiv preprint arXiv:2505.15386*, 2026.

Gautier Izacard, Patrick Lewis, Maria Lomeli, Lucas Hosseini, Fabio Petroni, Timo Schick, Jane Dwivedi-Yu, Armand Joulin, Sebastian Riedel, and Edouard Grave. Atlas: Few-shot learning with retrieval augmented language models. *Journal of Machine Learning Research*, 24(251): 1–43, 2023.

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

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

Zhuochun Li, Yong Zhang, Ming Li, Yuelyu Ji, Yiming Zeng, Ning Cheng, Yun Zhu, Yanmeng Wang, Shaojun Wang, Jing Xiao, and Daqing He. Rethinking llm-as-a-judge: Representation-as-a-judge with small language models via semantic capacity asymmetry. *arXiv preprint arXiv:2601.22588*, 2026.

Zhen Lin, Shubhendu Trivedi, and Jimeng Sun. Contextualized sequence likelihood: Enhanced confidence scores for natural language generation. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 10351–10368, 2024.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. G-eval: NLG evaluation using gpt-4 with better human alignment. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 2511–2522, 2023.

Andrey Malinin and Mark Gales. Uncertainty estimation in autoregressive structured prediction. In *International Conference on Learning Representations*, 2021.

Potsawee Manakul, Adian Liusie, and Mark Gales. SelfCheckGPT: Zero-resource black-box hallucination detection for generative large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 9004–9017, 2023.

Jinming Nian, Zhiyuan Peng, Qifan Wang, and Yi Fang. W-rag: Weakly supervised dense retrieval in rag for open-domain question answering. In *Proceedings of the 2025 International ACM SIGIR Conference on Innovative Concepts and Theories in Information Retrieval (ICTIR)*, pp. 136–146, 2025.

Ola Shorinwa, Zhiting Mei, Justin Lidard, Allen Z. Ren, and Anirudha Majumdar. A survey on uncertainty quantification of large language models: Taxonomy, open research challenges, and future directions. *arXiv preprint arXiv:2412.05563*, 2025.

Karen Sparck Jones. A statistical interpretation of term specificity and its application in retrieval. *Journal of documentation*, 28(1):11–21, 1972.

ZhongXiang Sun, Xiaoxue Zang, Kai Zheng, Jun Xu, Xiao Zhang, Weijie Yu, Yang Song, and Han Li. RedeEP: Detecting hallucination in retrieval-augmented generation via mechanistic interpretability. In *The Thirteenth International Conference on Learning Representations*, 2025.

Liam van der Poel, Ryan Cotterell, and Clara Meister. Mutual information alleviates hallucinations in abstractive summarization. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 5956–5965, 2022.

Liang Wang, Haonan Chen, Nan Yang, Xiaolong Huang, Zhicheng Dou, and Furu Wei. Chain-of-retrieval augmented generation. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

Siyang Wu, Honglin Bao, Sida Li, Ari Holtzman, and James A. Evans. Mapping overlaps in benchmarks through perplexity in the wild. *arXiv preprint arXiv:2509.23488*, 2026.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 2369–2380, 2018.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*, 2020.

Zihao Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. Calibrate before use: Improving few-shot performance of language models. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139, pp. 12697–12706, 2021.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging LLM-as-a-judge with MT-bench and chatbot arena. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2023.

Table 2: Transfer performance of the learned weighting function. Retention measures the fraction of native improvement over base PPL preserved after transfer, while performance loss is the increase in disagreement relative to native weighting PPL.

Transfer type	Retained improvement (%)	Performance loss (pp)
Dataset transfer	93.22	0.66
Model transfer	95.38	0.49
Joint model + dataset	89.14	1.23
Overall	91.72	0.90

A Model Transferability & Sensitivity Analysis: Additional Evidence

A.1 Model Transferability Analysis

Figure 4 first shows that learned weighting consistently improves over base PPL in all nine native mode–dataset settings. Averaged across the nine settings, disagreement decreases from 15.69% with base PPL to 3.55% with native weighting PPL, corresponding to a 12.15 percentage-point reduction. The improvement is observed across all three model families: Qwen3-8B decreases from 22.85% to 3.85% on average across datasets, Gemma-2-9B from 14.61% to 3.94%, and Llama-3.1-8B from 9.62% to 2.85%. The largest individual reduction occurs for Qwen3-8B on MS MARCO, where disagreement drops from 36.07% to 2.01%. These results are also competitive with the LLM-as-judge baseline reported in Table 1

More importantly, most of this improvement is preserved when the learned weighting function is transferred. As summarized in Table 2, transferred weights retain over 89% of the native improvement across all transfer types, with an average performance loss of at most 1.23 percentage points relative to native weighting. Every transferred configuration still outperforms the corresponding target model's base PPL. Across all 72 directed transfers, the median retained improvement is 93.09%. Model transfer retains the largest fraction of the native improvement, at 95.38%.

These results indicate that the benefit of learned token weighting is not confined to the model–dataset combination on which the weighting function is trained. Instead, the results suggest that the learned weighting function captures token level regularities that are substantially shared across datasets and model families, consistent with our interpretation of weighting as a transferable correction for token likelihood regularity.

A.2 Model Sensitivity Analysis

We investigate whether the learned embedding-to-weight model captures semantic factors underlying token likelihood regularity. For each gold answer token a_i, we measure the association between its raw learned weight w_i^d and its similarity absolute value to the conditioning document d. We compare two document representations: mean pooling over all document tokens, and over the top 10% of document tokens selected by attention for the corresponding answer token, inspired by ReDeEP (Sun et al., 2025). Let v_i denote the contextual representation of a_i used by the semantic probe and u_i^d the pooled document representation; we compute $s_i^d = |\cos(v_i, u_i^d)|$. These representations probe the learned weights independently of the causal predictive states used to produce them.

To distinguish within-answer weight allocation from differences across examples, we additionally center both measurements within each answer document pair:

$$\widetilde{w}_i^d = w_i^d - \frac{1}{n}\sum_{j=1}^{n} w_j^d, \qquad \widetilde{s}_i^d = s_i^d - \frac{1}{n}\sum_{j=1}^{n} s_j^d. \tag{12}$$

This removes differences between example in average weight and similarity while retaining relative variation among tokens sharing the same question and document. The motivation is consistent with evidence that language model predictions depend on prompt specific biases (Zhao et al., 2021).

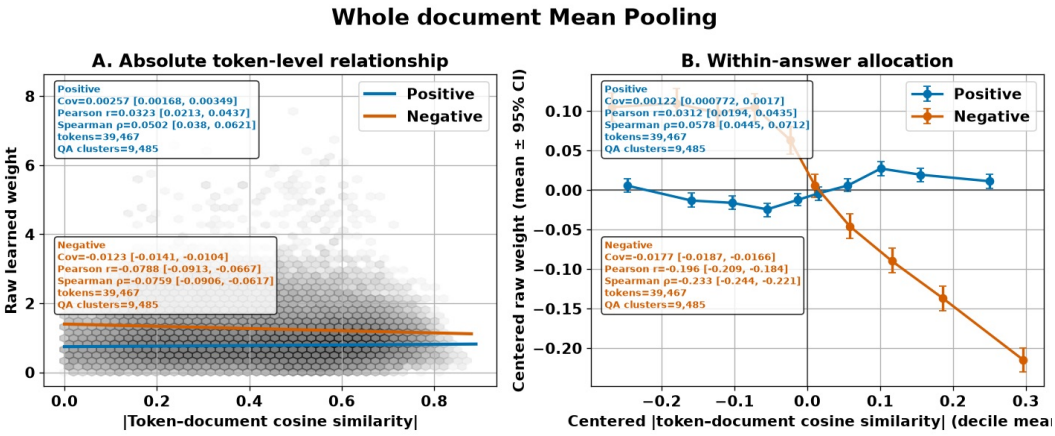


Figure 7: Association between absolute token–document cosine similarity and learned weights using whole-document mean pooling. Panels show overall relationships (A) and within-answer centered relationships (B), separately for supporting and hard-negative documents. Error bars indicate 95% confidence intervals.

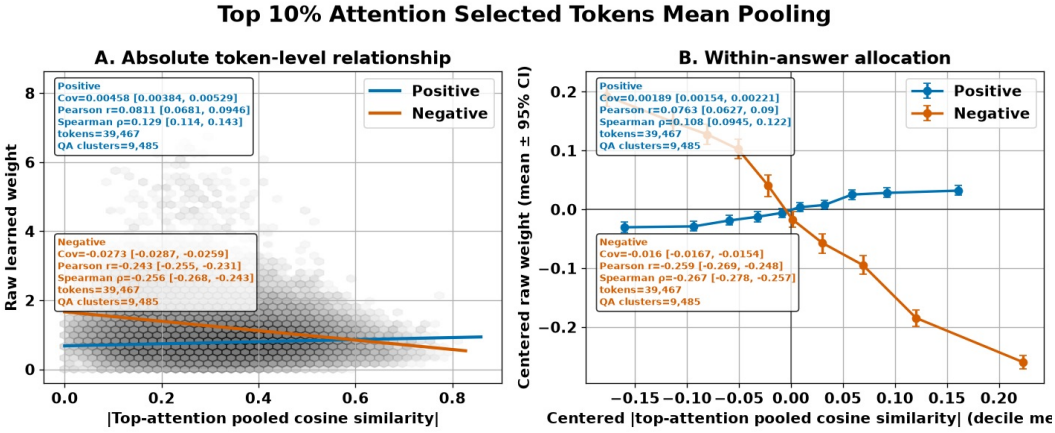


Figure 8: Association between absolute token–document cosine similarity and learned weights using mean pooling over the top 10% of document tokens selected by attention. Panels show overall relationships (A) and within-answer centered relationships (B), separately for supporting and hard-negative documents. Error bars indicate 95% confidence intervals.

Figure 7 shows positive associations for supporting documents and negative associations for negative documents under both pooling strategies in hotpotqa dataset. Attention selected pooling strengthens the pooled Pearson correlations from 0.032 to 0.081 for supporting documents and from -0.079 to -0.243 for hard negatives. Figure 8 shows within answer centering over the top 10% of document tokens selected by attention for the corresponding, answer token makes the negative-document relationship more pronounced, yielding correlations of -0.196 and -0.259 for whole document and attention selected pooling, respectively. The corresponding positive document correlations remain small at 0.031 and 0.076, although attention selected pooling reveals a clearer increasing trend across similarity bins. Thus, tokens with greater relative similarity tend to receive higher weights under supporting documents and lower weights under hard negatives.

The stronger associations with attention selected representations suggest that learned weights are more closely related to localized document content than to whole document similarity. The opposite trends across document types are consistent with a weighting function that responds to semantic relationships relevant to document ranking, rather than uniformly rewarding similarity. Together, these findings support sensitivity to semantic factors underlying token likelihood regularity, while the modest correlations indicate that this probe explains only part of the learned weighting behavior.

B Experiments Setting Details

B.1 Native and Transfer Experiment Settings

Models and implementation. The core training matrix contains `Qwen/Qwen3-8B`, `meta-llama/Llama-3.1-8B`, and `google/gemma-2-9b`, each trained separately on HotpotQA, Natural Questions (NQ), and MS MARCO. The native-only expansion uses `Qwen/Qwen3-14B`, `Qwen/Qwen3-32B`, `Qwen/Qwen3.5-9B`, and `Qwen/Qwen3.6-27B` on the same datasets. We use the corresponding Hugging Face tokenizers and text-generation backbones. The checked-in training implementation uses bfloat16 backbone parameters on CUDA and a float32 weight head; it does not enable 4-bit quantization. The original backbone parameters remain frozen, while the LoRA parameters and the linear weight head are optimized jointly.

Dataset conversion and reference answers. The converters use the training splits of `hotpotqa/hotpot_qa` (`distractor`), `google-research-datasets/natural_questions` (`default`), and `microsoft/ms_marco` (`v2.1`).

For HotpotQA, sentences belonging to a passage are concatenated with spaces, and passages are deduplicated by title. Each annotated supporting passage is a separate positive document; the implementation does not concatenate all supporting passages into one multi-hop document. The dataset answer is retained unchanged.

For NQ, we use the first valid annotation with a long-answer span and nonempty short-answer text, excluding yes/no annotations. Multiple short-answer strings in that annotation are joined with a comma and a space. The positive document is the long-answer span with HTML tokens removed; documents are deduplicated by URL and span boundaries. We score only the selected reference answer, without combining likelihoods across alternative valid answer forms.

For MS MARCO, the converter reservoir-samples 100,000 eligible rows using seed 42. An eligible row has a nonblank question, a nonblank answer, and at least one nonblank passage marked `is_selected=1`. The answer is the first nonblank `wellFormedAnswers` entry, falling back to `answers`. The corpus contains the exact-text-deduplicated union of selected passages.

Hard-negative construction. The shared mining pipeline uses Qwen3-8B's final-layer hidden state at the last input token to embed each question and passage. Embeddings are ℓ_2-normalized and indexed using exact FAISS inner-product search, which is equivalent to cosine search. The mining prompts are:

```
Retrieve questions answered by this passage.
Passage:
{document}
Matching question:

Retrieve the passage that answers this question.
Question:
{question}
Matching passage:
```

The globally highest-scoring passage is retained as a hard negative only if its identifier is outside the question's annotated positive set. Questions whose top-ranked passage is annotated positive are omitted from this mined pair pool. Each retained negative can be paired with each annotated positive; thus pair counts need not equal unique-question counts. Corpus embedding prompts are left-truncated to 4,096 tokens, with a maximum batch size of eight and an 8,192-padded-token batch budget. No answer-string exclusion is applied to mined negatives, so "negative" denotes absence from the annotated positive set rather than a separate factuality judgment.

Preparation, splits, and checkpoint selection. For each model–dataset setting, the preparation script reservoir-samples 10,000 valid document pairs with seed 42, shuffles them, and divides them into 8,000 preparation pairs and 2,000 internal test pairs. The former is split again, using seed 42, into 6,400 optimization pairs and 1,600 validation pairs. These are pair-level splits. The training loader shuffles with seed $42 + e$ at epoch e. Completed runs select the checkpoint with the lowest

Table 3: Training settings in the native-model training notebooks. Batch sizes count document pairs, each containing positive and negative contexts.

Setting	Value
Epochs	5
Microbatch size	1 pair
Gradient accumulation	16 microbatches
Effective batch size	16 pairs
Optimizer	AdamW
Learning rate	10^{-5}, constant
Adam momentum coefficients; epsilon	$(0.9, 0.999)$; 10^{-8}
Weight decay	0.03
Gradient-norm clipping	0.5
Ranking coefficient β	1.0
Entropy coefficient λ	0.1
LoRA rank; scaling parameter	$r = 8$; $\alpha = 16$
LoRA dropout	0.10
LoRA targets	Query and value projections
Weight-head initialization	Zero weights and bias
Gradient checkpointing	Enabled, non-reentrant
Training context cap	3,000 tokens per document

validation objective, including regularization. Each run records its split manifest and checkpoint identifiers. Notice that these results are shared among all different models.

There is no learning-rate warmup or scheduler in the training loop. The usual LoRA targets are `q_proj` and `v_proj`; for Qwen3.5-9B and Qwen3.6-27B the expansion notebook additionally targets `in_proj_qkv`. The zero-initialized head initially produces the same weight, $\mathrm{softplus}(0) = \log 2$, for every token.

Likelihood prompt and context budgets. The standard likelihood prompt is below. Braces denote substituted fields; the final colon is followed by one space.

```
Given question: {question}
Relevant context: {document}
The correct answer to the question should already be in the context.
Output answer:
```

This prompt uses only the document being scored. It is supplied as plain text without a chat template. Training preserves the first 3,000 document tokens. The native/transfer evaluation runner defaults to an 18,000-token context cap and further shortens documents to fit each model's sequence limit while reserving the answer. The Table 1 baseline notebook separately sets a 2,048-token cap for SAR and CSL; sampled-generation baselines instead retain at most 4,096 whitespace-delimited context words. These limits are method-specific in the implementation.

Evaluation cohorts and metric. The native/transfer runner samples 10,000 evaluation pairs per dataset with seed 42 after excluding question IDs appearing in any preparation-training partition for that dataset across the compared checkpoints. This exclusion also covers the validation subset carved from those partitions. All methods within this runner reuse the persisted ordered pair sample. A pair agrees with the annotation only when the positive score is strictly lower than the negative score; exact ties count as disagreements. Ranking weighted mean NLL is equivalent to ranking its exponential.

Cross-model weight projection. For each target dataset, the source weighting model processes that dataset's questions, documents, and reference answers. The source model remains necessary to compute weights; its head is not attached directly to the target model. Let I_i^S and I_j^T be answer-character spans from the source and target fast tokenizers, and define $o_{ij} = |I_i^S \cap I_j^T|$. Projection

uses

$$\widehat{w}_j^S = \sum_i w_i^S \frac{o_{ij}}{\sum_k o_{ik}}, \qquad L_{S\to T} = \frac{\sum_j \widehat{w}_j^S \ell_j^T}{\sum_j \widehat{w}_j^S}. \tag{13}$$

The implementation conserves total source weight mass. Identical-model transfers use the original weight vector after checking token-grid identity. Zero-width spans, source tokens without target overlap, and uncovered target tokens raise errors rather than receiving an arbitrary fallback weight. There are 18 dataset-only, 18 model-only, and 36 joint transfers. No target retraining is performed.

Predefined weighting baselines. IDF uses $w_i = -\log[(\mathrm{df}(a_i)+1)/(N+1)]$, where N is the number of documents in the active dataset corpus and document frequencies are computed using the scoring model's tokenizer. InvLogFreq uses $w_i = -\log P_{\mathrm{unigram}}(a_i)$ from the Google Books English 1-gram frequency cache, with smoothing 10^{-9}. Thus the implemented weight is negative log frequency, not the reciprocal of log frequency. Model Prior weighting uses $w_i = \exp(-\ell_i)$ from the base model's conditional token probability. All three use normalized weighted mean NLL.

LLM judge. The baseline notebook instantiates the judge with the same Qwen3-8B wrapper. It performs no free-form generation: the score is $\log P(\texttt{0} \mid p) - \log P(\texttt{1} \mid p)$ at the end of the following prompt, with lower scores indicating stronger support.

```
You are a factual judge.  Given a question, two contexts, and a
candidate answer, determine whether the answer is supported by Context
1.  Use Context 2 only as a comparison.

Context 1 (being evaluated):  {document}
Context 2 (comparison only):  {other document}
Question:  {question}
Candidate Answer:  {answer}

Is the candidate answer supported by Context 1?
Reply with 1 for Yes (Supported) or 0 for No (Not Supported).
Response:
```

The positive and negative scores swap the two document fields, so the currently evaluated document always occupies Context 1. The judge therefore sees both documents, whereas standard likelihood scoring sees only one. The prompt-budget policy allocates the available context budget symmetrically between the judge's two documents.

B.2 Semantic Sensitivity Experiment Setting

Frequency and attribution analysis. The long-text analysis in Figure 5(a) uses the Qwen3-8B weighting checkpoint and the Harry Potter text described in Section 4.5. The notebook sets the window length to 10,000 tokens and overlap to 1,000 tokens (stride 9,000). Weights are predicted from raw preceding text without any prompt. The first token of a window has no preceding state and is not predicted within that window. Overlap predictions make sure no token are skipped because of first token. Overlap parts are retained for diagnostics but are not double-counted in primary token statistics. Token-type statistics aggregate occurrence weights; the notebook's minimum count for eligible statistics is five.

For the qualitative QA visualization in Figure 5(b), token contribution is normalized as $\ell_i / \sum_j \ell_j$ for uniform weighting and $w_i \ell_i / \sum_j w_j \ell_j$ for learned weighting. These contribution fractions differ from normalized weights $w_i / \sum_j w_j$.

Paired contextual weight distance. The controlled NQ analysis selects 1,000 unique questions from the persisted 10,000-pair NQ baseline sample, using seed 42. A further 1,000 calibration questions are sampled from the remaining questions, with explicit question-ID disjointness. The implementation uses `Qwen/Qwen3-8B` in bfloat16, the NQ-trained weighting model, and the checkpoint `epoch_005`; it checks the resolved checkpoint against this expected identifier. Documents are capped at 2,048 tokens. All methods use the same answer-token IDs, retain all answer tokens, and score both positive and hard-negative documents.

This analysis uses *context-conditioned variants* of the Table 1 baselines. For SAR, similarity compares the full likelihood prompt plus answer against the same text with one answer token deleted. For CSL, a `Context: {document}` line is inserted immediately before the final question in the assessment prompt above, and the ten heads are recalibrated on the separate contextual calibration sample. Consequently, these variants can respond to changing documents even when the question and answer stay fixed.

For each method and document, weights are normalized to sum to one, and we calculate $D = \sum_i |\widetilde{w}_i^+ - \widetilde{w}_i^-|$. Zero-total nonnegative vectors fall back to uniform weights in the shared normalization helper. Reported deviations are sample standard deviations across questions, using denominator $n - 1$, rather than standard errors or variation across training seeds. The analysis also computes percentile 95% confidence intervals from 2,000 question-level bootstrap resamples with seed 42; method differences use paired resampling.

Representation probe in Appendix A.2. The semantic-probe notebook samples 10,000 HotpotQA pairs with seed 42 and uses the final hidden layer of the adapted weighting backbone. Answer-token semantic representations are computed at the token positions themselves under `Question: {question}\nAnswer: `, independently of the predictive states used by the weight head. Whole-document vectors average token representations under `Question: {question}\nDocument: `. For attention-selected pooling, the notebook reconstructs final-layer answer-query/document-key attention, applies rotary positions and grouped-query head expansion, normalizes over document tokens, averages attention across query heads, and selects $\max(1, \lceil 0.1|d| \rceil)$ document tokens for each answer token. The reported mean-pooling variant averages their hidden representations. This separately constructed probe should not be interpreted as extracting unmodified attention from the likelihood-scoring prompt.

The analysis uses absolute cosine similarity and raw learned weights, separately for positive and negative documents. For the within-answer analysis, both quantities are centered by their answer–document means. The notebook computes Pearson and Spearman correlations, decile summaries, and question-level bootstrap summaries with 1,000 resamples. Probe features are detached and are not used to update the weighting model.

C More Contribution Shift Case Studies

We provide more token-level NLL contribution redistribution cases here:

Answer token	Mer	lin	and	Morg	ana	are	friends	.
Who is Merlin's friend?								
Weight %	7.02%	15.69%	14.55%	14.02%	14.84%	10.31%	14.61%	8.97%
Weight	0.07	0.16	0.15	0.14	0.15	0.10	0.15	0.09
NLL %	63.58%	0.00%	32.52%	0.00%	-0.00%	1.01%	0.22%	2.67%
NLL	15.89	0.00	8.13	0.00	-0.00	0.25	0.05	0.67
Who is Morgana's friend?								
Weight %	7.91%	15.87%	8.43%	15.03%	15.41%	12.42%	15.08%	9.84%
Weight	0.08	0.16	0.08	0.15	0.15	0.12	0.15	0.10
NLL %	44.97%	0.00%	32.57%	17.73%	0.00%	1.78%	0.30%	2.64%
NLL	11.46	0.00	8.30	4.52	0.00	0.45	0.08	0.67

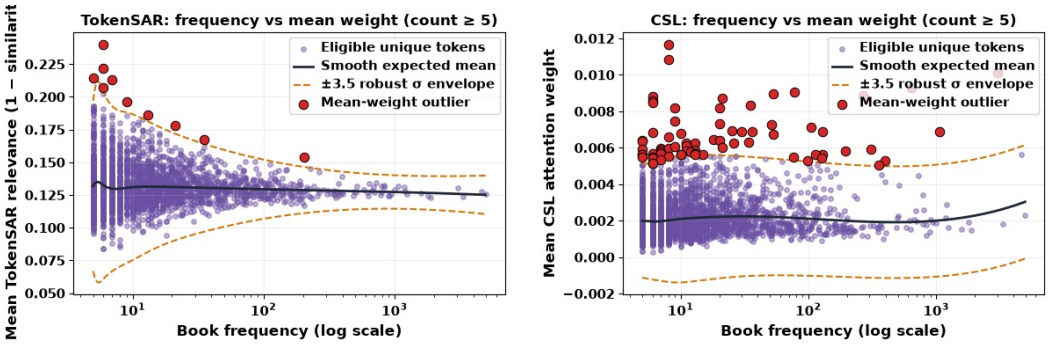
Figure 9: Token-level NLL attribution under different queries with a fixed context and answer. Same context and answer as in Figure 5(b) with different queries. Fixed context: When Arthur first arrived at Camelot, he quickly learned that Merlin was the person to find whenever something went wrong. What surprised him was that Merlin, in turn, always seemed to find Morgana. When Merlin disappeared for hours, Gwen knew to check the old garden where Morgana liked to read; when Morgana was furious with the court, Merlin was usually the only one who could make her laugh before supper. They argued constantly—about magic, horses, terrible poetry, and whether Merlin had stolen the last honey cake—but somehow each always knew what the other was thinking. Years later, when people told stories of Camelot's great alliances and legendary loyalties, Arthur would sometimes smile and think that the strongest bond in the castle had never needed a title.

Answer token	The	forest	blocks	strong	winds	and	gives	birds	a	place	to
How does the forest help the nearby village?											
Weighted NLL %	14.90%	0.02%	33.76%	0.18%	10.83%	0.60%	18.94%	18.77%	0.31%	0.00%	0.00%
Weighted NLL	3.74	0.00	8.47	0.05	2.72	0.15	4.75	4.71	0.08	0.00	0.00
NLL %	33.32%	0.01%	25.06%	0.15%	9.85%	0.71%	14.56%	13.27%	0.28%	0.00%	0.00%
NLL	19.46	0.01	14.64	0.09	5.75	0.41	8.51	7.75	0.16	0.00	0.00
Why is the forest important to seabirds?											
Weighted NLL %	14.05%	0.00%	51.02%	0.13%	8.60%	0.46%	10.54%	12.51%	1.30%	0.02%	0.00%
Weighted NLL	3.68	0.00	13.36	0.03	2.25	0.12	2.76	3.28	0.34	0.00	0.00
NLL %	32.72%	0.00%	37.67%	0.11%	7.24%	0.53%	8.06%	10.13%	1.15%	0.01%	0.00%
NLL	20.96	0.00	24.13	0.07	4.64	0.34	5.16	6.49	0.74	0.01	0.00

Figure 10: Token-level NLL attribution under different queries with a fixed context and answer. Second uery contains **seabird** increase the token related to **birds** contribution. Fixed context: A coastal forest protects the nearby village from strong ocean winds. Its trees also provide nesting sites for many seabirds that return every spring. Because of these two roles, local officials have restricted logging in the area.

Answer token	A	pipeline	carries	des	al	inated	seaw	ater	to	the	reservoir
How does Pelora maintain its drinking-water supply during the dry season?											
Weighted NLL %	75.04%	2.70%	0.58%	0.01%	0.00%	0.00%	0.00%	0.00%	21.19%	0.48%	0.00%
Weighted NLL	5.41	0.19	0.04	0.00	0.00	0.00	0.00	0.00	1.53	0.03	0.00
NLL %	86.76%	0.91%	0.22%	0.00%	0.00%	0.00%	0.00%	0.00%	11.91%	0.19%	0.00%
NLL	32.00	0.34	0.08	0.00	0.00	0.00	0.00	0.00	4.39	0.07	0.00
What role does desalinated seawater play in Pelora's water system?											
Weighted NLL %	61.89%	25.54%	0.73%	0.07%	0.00%	0.00%	0.00%	0.00%	11.26%	0.51%	0.01%
Weighted NLL	6.24	2.58	0.07	0.01	0.00	0.00	0.00	0.00	1.14	0.05	0.00
NLL %	76.75%	14.31%	0.38%	0.04%	0.00%	0.00%	0.00%	0.00%	8.23%	0.28%	0.00%
NLL	28.10	5.24	0.14	0.01	0.00	0.00	0.00	0.00	3.01	0.10	0.00

Figure 11: Token-level NLL attribution under different queries with a fixed context and answer. Second query contains **role** increase the token related to **pipeline** contribution. Fixed context: A small island called Pelora receives most of its fresh water from a reservoir in the northern hills. During the dry season, rainfall becomes too scarce to refill the reservoir naturally. To compensate, an underground pipeline carries desalinated seawater from a treatment plant on the southern coast into the reservoir. The same pipeline also supplies several farming communities along its route. Without it, both the island's drinking-water reserve and much of its agriculture would face serious shortages during the summer.



(a) Token-only SAR (b) CSL

Figure 12: Mean token weights as a function of token frequency in Harry Potter and the Philosopher's Stone for (a) token-only SAR and (b) CSL. SAR weights concentrate around a nearly flat mean trend as frequency increases, while CSL exhibits more upper outliers across a wider frequency range. The panels use different vertical scales.

D Frequency Distribution in CSL and SAR Weighting

We compare the frequency–weight patterns of token-only SAR and CSL with our learned weighting in Figure 5(a). Each point represents a token type occurring at least five times, with its weight averaged across occurrences, same setting as in Sec. 4.5. SAR exhibits a nearly flat mean trend like our learned weights, with weights concentrating tightly around this trend as frequency increases. CSL shows a more asymmetric distribution, with numerous upper outliers spanning a wider frequency range. Our learned weights exhibit a relatively flat mean trend while retaining greater dispersion than SAR at intermediate frequencies and fewer upper outliers than CSL under the plotted robust-envelope criterion. These plots describe variation across token-level averages; they do not directly measure sensitivity to changes in context.