DON'T BE CHOOSY: SCORING OVER CHOOSING FOR VERBALIZED LLM CONFIDENCE

Anonymous authors
Paper under double-blind review

ABSTRACT

Accurate uncertainty estimation is critical for the reliable deployment of large language models (LLMs). A simple and widely used approach is to ask a model to verbalize its confidence on each item, referred to as AbsScore henceforth. However, AbsScores can be poorly discriminative, exhibiting overconfidence or collapsing to a small set of values. Recent work proposed pairwise elicitation as an alternative, asking the model which of two items it is more confident about. While effective, this formulation forces a preference even when confidences are similar and discards information about the magnitude of the confidence difference. To overcome these limitations, we introduce **PairScore**, a simple alternative: present two items together, but ask the model to score both rather than choose between them. This preserves graded confidence while retaining the benefits of pairwise elicitation, and the resulting scores can be aggregated directly without fitting a separate ranking model. PairScore improves confidence ranking over AbsScore and PairChoice in most claim-verification settings, while gains in best-of-N response selection depend on the task. PairScore also improves calibration and reduces sensitivity to presentation order and comparison partners. We extend both formats beyond pairs: k**-score** jointly scores k items and k**-rank** returns a full ordering of them. Scoring improves as more items share a prompt, whereas ranking peaks at small k and then declines; k-score matches or exceeds k-rank at every k tested.

1 INTRODUCTION

Large language models (LLMs) are increasingly deployed in high-stakes settings such as health, law, and decision support, where errors can carry substantial costs. A practical safeguard is to elicit a *confidence* or *uncertainty* estimate alongside each model output, allowing low-confidence predictions to be flagged, filtered, or deferred to human review (Chen et al., 2023; Gupta et al., 2024). A simple, low-cost, and widely used approach is direct verbalization (AbsScore): prompting the model to express its confidence as a numerical score, for example on a 0–100 scale (Kadavath et al., 2022; Tian et al., 2023; Yang et al., 2024).

However, verbalized numerical confidence has well-documented limitations: scores can cluster near scale endpoints, shift with prompt phrasing, and vary across repeated queries (Shrivastava et al., 2023; Tian et al., 2023; Xiong et al., 2024). Such pathologies can make the resulting confidence scores less informative for distinguishing and ranking model outputs and less reliable in general.

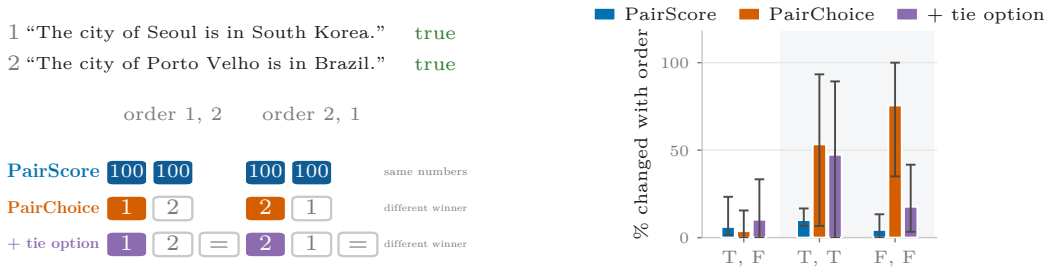
One way to address these limitations is to replace absolute judgments with relative ones, comparing items rather than assessing each in isolation. This idea has roots in the classical law of comparative judgment (Thurstone, 1927). More recently, Shrivastava et al. (2025) applied pairwise comparisons to LLM uncertainty estimation: given two items, the model indicates which one it is more confident about, and the resulting comparisons are aggregated using a Bradley–Terry model. We refer to this formulation as **PairChoice**.

Method	Context	Output
AbsScore	Single item	A numerical score.
PairChoice	Two items	Binary preference.
PairScore	Two items	Two numerical scores.

Table 1: Comparison of absolute and relative confidence elicitation methods.

PairChoice trades the problems of absolute scoring for a different set of limitations. First, it forces the model to choose one item even when it might have similar confidence in both. This can arise,

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) One pair, shown in both orders.

(b) All pairs, by truth values of the pair.

Figure 1: **Forced choice favors position when a tie is correct. (a)** Two true claims shown to Qwen3-8B in both orders. PairScore assigns equal scores in either order; PairChoice selects the first claim, changing its winner even with a tie option (third row). **(b)** Fraction of pairs whose answer changes under order reversal, grouped by truth values (T: true, F: false), averaged over 6 open-weight models; whiskers show the range. Shading marks pairs where a tie is correct. Such pairs cannot be excluded in practice because ground truth is unknown.

for example, when both items are true or both are false, yet PairChoice still requires a preference. Second, a binary choice loses information about *how much* more confident the model is in one item than the other. For example, confidence scores of $(90\%, 10\%)$ and $(51\%, 49\%)$ would induce the same PairChoice outcome despite representing different levels of confidence. Section 2 shows the first limitation on controlled pairs.

Motivated by these limitations, we ask a simple question: *rather than choosing between two items, why not score both?* We call this **PairScore**. It retains the pairwise setup of PairChoice, but asks the model to assign a numerical confidence score to each item instead of selecting one over the other (Table 1). PairScore therefore avoids a forced preference while preserving graded confidence information, and its scores can be aggregated directly without fitting a separate ranking model.

The same choice arises with more than two items. Given k items in one prompt, a model can be asked for a full ordering (k**-rank**), which generalises PairChoice, or for k scores (k**-score**), which generalises PairScore. We test whether presenting more items together improves confidence ranking by sweeping k from one, where k-score is AbsScore, to twenty. Performance improves beyond pairs in most settings, up to a setting-dependent k, and then plateaus or deteriorates. k-score performs better than or on par with k-rank throughout. Our main contributions are:

- We introduce **PairScore**, a simple alternative to PairChoice that retains the pairwise elicitation setup while replacing a forced binary choice with a numerical confidence score for each item.
- We compare absolute scoring, pairwise choice, and pairwise scoring across claim verification, best-of-N response selection, and additional ranking applications. PairScore improves confidence ranking over both alternatives in most claim-verification settings; response-selection gains depend on the task.
- We study when and how PairScore helps. On controlled pairs, a forced choice answers by position when a tie is correct and loses discrimination against extreme-confidence partners, while joint scoring largely avoids these effects; joint scoring is also better calibrated than either alternative.
- We extend both formats beyond pairs. Scoring k items together improves with k and holds up to twenty items, whereas asking for a full ordering peaks early and degrades. Scoring therefore scales better than ranking as the number of compared items grows.

2 Problems with PairChoice

To study fundamental problems with PairChoice, we run two controlled experiments that hold the content fixed and vary one factor at a time: the order in which a pair is presented, and the partner an item is presented with. Protocols and prompts are given in Appendix D.

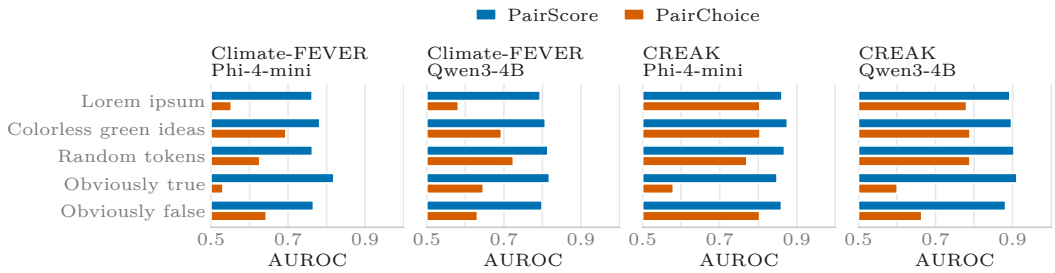


Figure 2: **PairScore is more robust to the choice of partner than PairChoice.** AUC-ROC when every claim is paired with the same fixed partner, for five partners. Bars start at chance.

Does the answer change when only the order changes? We construct pairs from the city statements of Marks & Tegmark (2024), whose labels are known by construction, in three categories: one true and one false claim, where a preference is the correct answer, and two true or two false claims. Each pair is presented to 6 open-weight models in both orders. We record the winner by claim rather than by position and measure the fraction of pairs whose answer changes between the two orders.

When one claim is true and the other false, both formats select the true claim in either order. When a tie is the correct answer, the formats differ (Figure 1). PairChoice changes its answer on about half of the both-true pairs and on most of the both-false pairs. PairScore changes on a small fraction of either, and in most cases assigns both claims the same score. Adding a tie option to the PairChoice prompt only partly closes the gap. On both-false pairs most models use the option and order dependence drops. On both-true pairs most models rarely use it, and the answer continues to depend on order. Joint scoring remains robust in these three settings of pairs.

Does the answer change when only the partner changes? A pairwise format evaluates an item together with another item. If the output reflects the item, the choice of partner should have little effect. We pair every claim in CREAK and Climate-FEVER with a single fixed partner and vary only the partner: three meaningless texts, one obviously true statement, and one obviously false statement.

PairScore is largely unaffected. Its AUC-ROC varies only a little across the five partners (Figure 2). PairChoice is affected substantially. The meaningless partners leave it close to PairScore, but the obviously true partner reduces it to near chance, since a claim compared against a statement the model is certain of usually loses regardless of its own content. PairScore is therefore more robust to the choice of partner. Appendix D adds a partner test with real claims and shows that joint scoring separates items whose absolute scores coincide.

3 Related Work

Verbalized confidence and uncertainty estimation. LLM uncertainty estimation has been studied through token probabilities, consistency across sampled responses (Wang et al., 2023; Kuhn et al., 2023), internal representations (Marks & Tegmark, 2024; Kossen et al., 2024), and verbalized confidence (Kadavath et al., 2022). Our work focuses on verbalized confidence, which requires only text-based access to a model and is therefore applicable to black-box models. Several works have found drawbacks of verbalized confidence scores, including score concentration to a few values, sensitivity to elicitation prompts and scales (Xiong et al., 2024; Yang et al., 2024). Moreover, pursuing better calibration metrics does not eliminate these problems (Xia et al., 2026). From a mechanistic perspective Kirchhof et al. (2026) show uncertainty is linearly encoded in hidden states but does not surface faithfully in numerical outputs. We investigate whether changing the elicitation context, by presenting two items together, can improve the informativeness of verbalized confidence scores.

Relative confidence elicitation. The closest work is Shrivastava et al. (2025), who elicit pairwise confidence preferences and aggregate them using methods such as Bradley–Terry to improve selective classification on multiple-choice question answering. Their formulation, which we call **PairChoice**, combines a pairwise context with a binary preference. We build directly on this work by separating these two design choices: **PairScore** retains the pairwise context but asks for a numerical confidence

score for each item. This permits the model to express similar confidence in both items and preserves graded confidence information that a binary choice cannot convey. Our evaluation extends beyond multiple-choice question answering to claim verification, best-of-N response selection, and additional selection and ranking applications.

Comparative judgment and LLM evaluation. Relative elicitation connects to the classical study of comparative judgment (Thurstone, 1927) and the use of preferences in reward learning and LLM alignment (Christiano et al., 2017; Ouyang et al., 2022). More directly, pairwise LLM judgments have been used to evaluate generated text: Liu et al. (2024) find that pairwise preference aggregation better aligns with human judgments than pointwise scoring. The AI co-scientist system of Gottweis et al. (2026) also uses an Elo-based pairwise tournament to rank research hypotheses, motivating our research-proposal ranking evaluation.

4 METHODS

We first define the three confidence elicitation methods considered in this work: AbsScore, PairChoice, and PairScore. We then describe how pairwise outputs are aggregated into per-item confidence scores, how pairs are sampled under a fixed query budget, and how the resulting scores are used for ranking and best-of-N selection.

4.1 SCORE ELICITATION

We study three ways of eliciting confidence from an LLM: absolute scoring (**AbsScore**), pairwise choice (**PairChoice**), and our proposed pairwise scoring method (**PairScore**). Full prompts are provided in Appendix I.

AbsScore. The model sees a single item and reports its confidence on a numerical scale from 0 to 100. For example: "Rate your confidence that 'The Eiffel Tower is in Berlin' is factually correct. Output a number from 0–100." This is a standard approach to verbalized confidence (Kadavath et al., 2022; Xiong et al., 2024). The reply is parsed as the first number in range, and a reply with no such number is retried once with a larger token budget and otherwise discarded. Scores are used as given, with no rescaling, so any tendency to cluster at a few values is preserved rather than corrected.

PairChoice. The model sees two items and chooses the one it is more confident about. For example: "Which statement are you more confident is correct? A: 'The Eiffel Tower is in Berlin.' B: 'Water boils at 100°C at sea level.' Output A or B." We refer to this approach as PairChoice. Shrivastava et al. (2025) explored this phrasing to find which questions an LLM is more confident in answering correctly. The two items are assigned to slots A and B at random on every call, so a preference for one slot cannot systematically favour either item. The reply must name exactly one item; a reply that names both or neither is retried once and otherwise discarded rather than resolved by a coin flip.

PairScore. PairScore retains the pairwise setup of PairChoice but replaces the forced choice with numerical scoring. Given two items, the model assigns each a confidence score on the same 0–100 scale used by AbsScore. PairScore therefore allows the model to express graded confidence in both items rather than reducing the comparison to a binary preference. It can also express a tie, by giving both items the same score, and the size of the gap between the two scores. The prompt asks for the reply in the form "A: [score], B: [score]"; slots are randomised as in PairChoice, both scores are kept, and a reply missing either score is retried once and otherwise discarded.

4.2 PAIRWISE AGGREGATION

PairChoice and PairScore both query items in pairs, but require different procedures to obtain a confidence score for each item.

PairChoice. Each comparison produces a binary preference, so we use the Bradley–Terry (BT) model (Bradley & Terry, 1952), following Shrivastava et al. (2025), to aggregate comparisons into per-item scores. Each item i is assigned a latent strength θ_i, with

$$P(i \succ j) = \sigma(\theta_i - \theta_j), \tag{1}$$

Algorithm 1 PairChoice with Bradley-Terry

Require: Items $x_1, \ldots, x_n$; budget B; regularization λ
Ensure: Confidence scores $S = [s_1, \ldots, s_n]$
$\theta_i \leftarrow 0$ for all i ▷ *uniform initialization*
$\mathcal{O} \leftarrow \emptyset$; $\mathcal{C} \leftarrow \emptyset$ ▷ *outcomes, compared pairs*
for $t = 1$ to B **do**
  $(i, j) \leftarrow$ random pair $\notin \mathcal{C}$
  $\mathcal{C} \leftarrow \mathcal{C} \cup \{(i, j)\}$
  $w \leftarrow \textsc{PairChoiceLLM}(x_i, x_j)$ ▷ *returns winner*
  $\mathcal{O} \leftarrow \mathcal{O} \cup \{(w, l)\}$ ▷ *w: winner index, l: loser index.*
  $\theta \leftarrow \text{FitBT}(\mathcal{O}, \lambda)$ ▷ *L2-regularized Bradley-Terry model learning*
  $S = \sigma(\theta)$ ▷ *σ is element-wise sigmoid operation here.*
end for
return S. ▷ *σ = sigmoid*

Algorithm 2 PairScore

Require: Items $x_1, \ldots, x_n$; budget B
Ensure: Confidence scores $S = [s_1, \ldots, s_n]$
$\text{scores}_i \leftarrow []$ for all i ▷ *collected scores per item*
$\mathcal{C} \leftarrow \emptyset$ ▷ *compared pairs*
for $t = 1$ to B **do**
  $(i, j) \leftarrow$ random pair $\notin \mathcal{C}$
  $\mathcal{C} \leftarrow \mathcal{C} \cup \{(i, j)\}$
  $(p_i, p_j) \leftarrow \textsc{PairScoreLLM}(x_i, x_j)$ ▷ *joint scores on 0–100*
  Append p_i to scores_i; append p_j to scores_j
end for
$s_i = \text{mean}(\text{scores}_i) \quad \forall i \in [n]$
return S

where $\sigma(\cdot)$ is the sigmoid function. Given T pairwise comparisons on N items, we estimate the strengths by minimizing the ℓ_2-regularized negative log-likelihood

$$\min_{\theta \in \mathbb{R}^N} \; -\sum_{t=1}^{T} \log \sigma(\theta_{w_t} - \theta_{l_t}) + \frac{\lambda}{2} \|\theta\|_2^2, \tag{2}$$

where w_t and l_t denote the preferred and non-preferred items in comparison t. We optimize this objective using L-BFGS-B and use the fitted strengths θ_i as per-item ranking scores.

PairScore. Each comparison directly returns two numerical confidence scores, $(p_i, p_j) \in [0, 100]^2$. Since an item may appear in multiple comparisons, it can receive multiple scores under different pairings. We use the mean of these scores as its final confidence score. Unlike PairChoice, PairScore therefore does not require fitting an additional ranking model.

Algorithms 1 and 2 summarise the two procedures.

Pair Sampling and Budget. With N items, exhaustively querying all $\binom{N}{2}$ pairs is generally impractical. We therefore sample pairs uniformly at random under a fixed query budget. For claim verification, all methods receive the same budget of $5N$ LLM calls. AbsScore scores each item five times, once at temperature 0 and four times at temperature 0.7, and averages the resulting scores. PairChoice and PairScore use the $5N$ calls for pairwise queries.

4.3 k-WISE EXTENSION

Pairs are the smallest joint context. Both pairwise formats naturally extend to more than two items in one prompt, and a longer context gives the model more items to weigh each one against, at the cost of a longer list to attend to and answer in one pass. **k-score** shows k items and asks for k scores on the 0–100 scale; an item's confidence is the mean of its scores across the groups it appears in, exactly as

Dataset	Method	GPT-5 mini	GPT-4o mini	GPT-4.1 nano	Gemini 2.5-FL	Qwen3 4B	Qwen2.5 3B	Phi-4 mini	Qwen3 1.7B
CREAK	AbsScore(1x)	94.7	95.7	94.6	95.0	88.9	87.7	83.9	71.8
	AbsScore(5x)	**97.9**	96.5	95.1	96.7	89.7	88.5	89.2	78.4
	PairChoice	97.6	96.1	94.9	93.8	83.2	76.3	**89.9**	59.0
	PairScore	96.4	**97.5***	**95.8**	**97.1**	**91.9***	**90.1**	88.1	**83.4***
FEVER	AbsScore(1x)	94.9	89.8	85.6	90.1	81.3	79.0	79.9	62.3
	AbsScore(5x)	**95.6**	91.0	87.9	**91.4**	81.7	**80.4**	82.1	63.1
	PairChoice	94.3	90.1	89.8*	85.9	77.7	66.6	**85.9***	49.4
	PairScore	93.7	**92.3**	**90.2***	91.1	**86.1***	73.7	82.3	**71.7***
TruthfulQA	AbsScore(1x)	86.1	87.0	**76.3**	86.5	73.4	65.3	63.8	56.2
	AbsScore(5x)	91.3	**87.9**	76.1	87.1	73.1	65.2	65.1	58.6
	PairChoice	**94.5***	85.7	74.3	88.5	71.7	72.4*	67.1	57.2
	PairScore	92.0	87.1	75.6	**91.7***	**79.1***	**82.4***	**74.7***	**69.1***
CliniFact	AbsScore(1x)	64.0	60.1	52.7	**60.7**	56.4	55.1	52.3	48.8
	AbsScore(5x)	65.0	62.1	53.8	60.4	56.4	57.8	54.6	47.0
	PairChoice	69.6*	58.2	**59.5***	59.4	50.4	**61.1**	52.0	48.4
	PairScore	**71.1***	**63.4**	56.5	54.4	**56.5**	50.5	**55.7**	**55.8***

Table 2: AUC-ROC (%) for all eight models across four claim-verification datasets. **Bold** = best method per model–dataset setting; * = significantly better than AbsScore(5x) at 95% (bootstrap).

in PairScore. k**-rank** shows k items and asks for a full ordering from most to least likely correct; the ordering is expanded into its $\binom{k}{2}$ pairwise outcomes and fitted with the same Bradley–Terry model as PairChoice. At k=1 k-score is AbsScore, and at k=2 the two formats are PairScore and PairChoice, so a sweep over k is continuous with the rest of the paper. Groups are formed round-robin: each round shuffles all items and cuts them into blocks of k, so every item is scored the same number of times at every k.

4.4 Using the Elicited Scores

Claim verification. Given N items with binary ground-truth, we rank them according to the aggregated confidence scores. We evaluate this ranking using AUC-ROC; higher AUC-ROC implies better separation of positive and negative items.

Best-of-N selection. Given K candidate responses, AbsScore selects the highest-scoring one. PairChoice and PairScore use sequential elimination (**PC-Elim** and **PS-Elim**): $K-1$ pairwise comparisons, each advancing the preferred or higher-scored response. Where final answers can be matched directly, we also compare against **Majority Vote**.

Additional ranking tasks. We also rank abstracts for systematic-review study selection by the model's confidence that they meet the inclusion criteria, scored by AUC-ROC against the final decisions, and compare pairwise judgments of research proposals against expert quality scores (Appendix H).

5 Experiments

5.1 Claim Verification

Setup. We evaluate on four binary claim-verification datasets: CREAK (Onoe et al., 2021) (entity facts), FEVER (Thorne et al., 2018) (Wikipedia), TruthfulQA (Lin et al., 2022) (adversarial QA), and CliniFact (Zhang et al., 2025) (clinical claims). We use N=500 items from each dataset with a

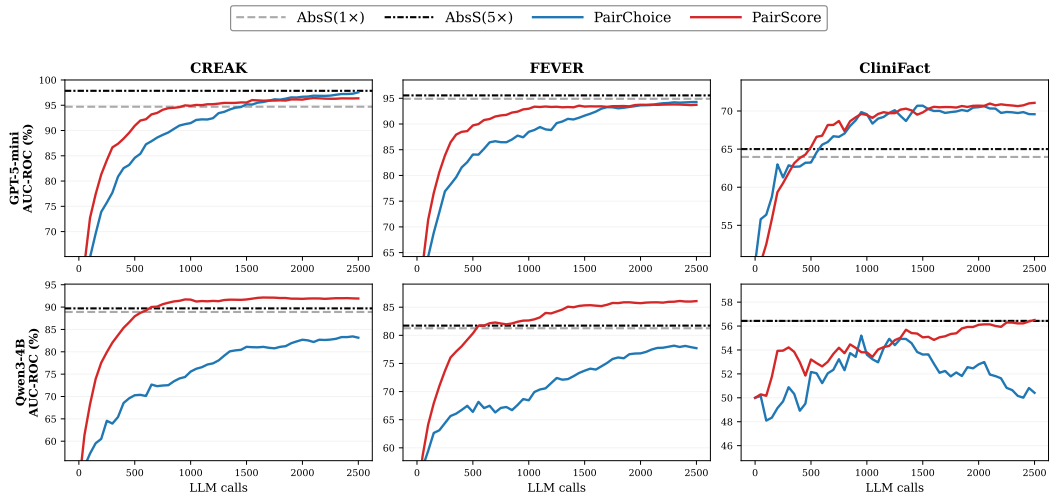


Figure 3: Convergence of AUC-ROC (%) vs. LLM calls for GPT-5-mini (top) and Qwen3-4B (bottom) across CREAK, FEVER, and CliniFact. Gray dashed = AbsScore(1×); black dash-dot = AbsScore(5×). PairScore reaches or exceeds the 5× AbsScore baseline with far fewer calls; PairChoice often does not reach it. Full results for all 8 models in Appendix C.1.

fixed seed and report AUC-ROC against the ground-truth binary labels. We evaluate eight models spanning different families and sizes, giving 32 model–dataset settings. Dataset and subsampling details are provided in Appendix A.1, and exact model and inference details in Appendix A.2.

Does relative elicitation improve over absolute scoring? Relative elicitation improves confidence ranking in a substantial fraction of the model–dataset settings, although the magnitude of the gain varies (Table 2). Relative methods help most for the smaller models, such as Qwen3-1.7B on CREAK and FEVER. The gains are smaller when AbsScore already provides a strong ranking, as for GPT-5-mini and Gemini-2.5-FL on FEVER. PairScore falls below AbsScore(5×) in a minority of settings, by up to seven points (Qwen2.5-3B on FEVER and CliniFact, Gemini-2.5-FL on CliniFact).

Does PairScore improve over PairChoice? PairScore outperforms PairChoice in most model–dataset settings (Table 2), and on TruthfulQA for nearly every model. Since the two methods use the same paired inputs and query budget, this comparison isolates the form of the elicited judgment: PairChoice reduces each comparison to a binary preference, whereas PairScore retains a graded confidence estimate for both items. The results indicate that this additional information is often useful for confidence ranking, although PairChoice remains competitive in some settings.

How does performance vary with the query budget? The preceding comparisons use the full $5N$-call budget. We next examine how performance changes as pairwise comparisons are added. Figure 3 shows the resulting AUC-ROC curves for GPT-5-mini and Qwen3-4B, with AbsScore(1×) and AbsScore(5×) as reference points. PairScore generally improves steadily with additional comparisons and often reaches or exceeds the AbsScore(5×) baseline before using the full budget. PairChoice converges less consistently and frequently remains below the same baseline. The corresponding curves for all eight models are reported in Appendix C.1; a comparison by token usage is given in Appendix C.2: PairScore reaches the AbsScore(5×) AUC-ROC in most settings while spending about two thirds of its token budget.

Are the scores calibrated? AUC-ROC measures ranking, not whether a confidence of 80 means an 80% chance of being correct, so we also evaluate the scores as probabilities. AbsScore and PairScore are divided by 100, and PairChoice uses $\sigma(\theta)$, the same quantity used for ranking; no recalibration is applied to any method. With p_i the resulting probability and $y_i \in \{0, 1\}$ the label, the Brier score (Brier, 1950) is $\frac{1}{n}\sum_i (p_i - y_i)^2$, and the expected calibration error (ECE; Naeini et al., 2015; Guo et al., 2017) is $\sum_b \frac{|B_b|}{n} |\mathrm{acc}(B_b) - \mathrm{conf}(B_b)|$ over ten equal-mass bins B_b. We report both for the 6

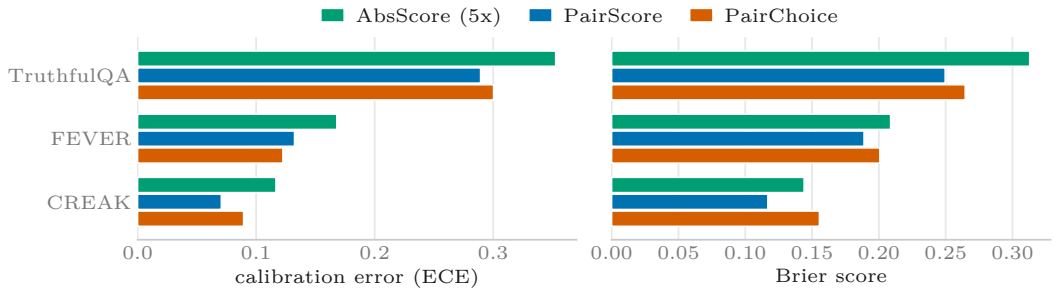


Figure 4: **PairScore is the best calibrated of the three formats.** Expected calibration error under equal-mass binning (left) and Brier score (right), by dataset, averaged over 6 open-weight models. Lower is better. PairChoice is read through $\sigma(\theta)$; nothing is recalibrated.

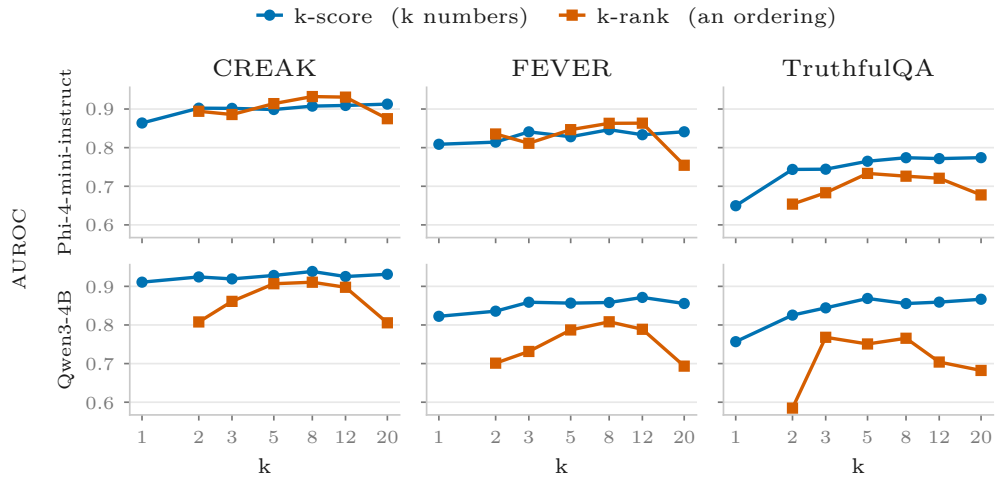


Figure 5: **k-score keeps improving with k; k-rank peaks early and then declines.** AUC-ROC against the number of claims shown in one prompt, one panel per dataset and model. k=1 is AbsScore and k=2 the pairwise formats.

open-weight models of Section 2 on CREAK, FEVER and TruthfulQA (Figure 4). The protocol and results under both ECE binnings are given in Appendix E.

PairScore has the lowest Brier score on every dataset and the lowest ECE on all datasets except FEVER, where PairChoice is slightly better. PairChoice is better than AbsScore on average, but not consistently: across individual model–dataset settings the two are close to even. As in the ranking results, the improvement over AbsScore is largest where AbsScore is weakest, on TruthfulQA and for the smaller models, and small where AbsScore is already strong. Absolute calibration nonetheless remains poor for every format on TruthfulQA (ECE near 0.3), so scores there should not be read as probabilities without recalibration.

Does the gain grow beyond pairs? We vary k from 1 to 20 for the k-score and k-rank formats of Section 4.3, on CREAK, FEVER and TruthfulQA with Qwen3-4B and Phi-4-mini-instruct, using N=500 items and five rounds per k (Appendix F). As expected from the intuition in Section 4.3, k-score improves as more claims are placed in the same prompt and then plateaus, with no degradation up to twenty claims (Figure 5). As in the pairwise results, the improvement is largest where the single-claim

	HumanEval+ (acc. %)			**AlpacaEval** (judge 0–7)		
Generator	AbsS	PC-Elim	PS-Elim	AbsS	PC-Elim	PS-Elim
GPT-4o-mini	68.3	67.7	**69.5**	3.88	4.19	**4.43**
GPT-5-mini	73.8	**81.1**	79.9	3.96	**4.96**	4.52
Gemini-2.5-FL	57.3	**72.0**	69.5	3.50	3.98	**4.56**

Table 3: Best-of-8 selection on the two tasks where pairwise selection helps most. Full results on seven benchmarks in Appendix G.

Table 4: Study-selection results, AUC-ROC (%); bold marks the best method per row.

Review — domain	Model	AbsScore	PairChoice	PairScore
Hall et al. (2012) — computer science	Gemini-2.5-FL	98.2	96.9	**98.4**
	GPT-4o-mini	97.8	98.0	**99.1**
Bladon et al. (2022) — conservation	Gemini-2.5-FL	**93.8**	91.8	92.9
	GPT-4o-mini	**96.7**	95.6	96.0
Walker et al. (2018) — biology & medicine	Gemini-2.5-FL	83.5	79.8	**85.1**
	GPT-4o-mini	79.1	**83.1**	**83.1**
Leenaars et al. (2020) — medicine	Gemini-2.5-FL	76.1	**81.9**	78.8
	GPT-4o-mini	**78.6**	76.0	**78.6**
Chausson et al. (2020) — ecology	Gemini-2.5-FL	85.1	79.2	**85.7**
	GPT-4o-mini	82.7	78.8	**83.3**

baseline is weakest. k-rank improves initially, peaks at a small k, and then falls below its starting point. Part of this decline is due to parsing: at large k the models often fail to return a usable ordering, whereas they almost always return the requested numbers. k-score is better than k-rank at every k we tested, and larger k also reduces the cost per item, since the instruction is shared across the block.

5.2 Best-of-N Selection

We also test whether the elicited scores can select among candidate responses. For each problem we generate $N{=}8$ responses at temperature 0.7 and select one with AbsScore, PairChoice-Elim or PairScore-Elim, using the same model that generated the responses, on seven benchmarks covering mathematical reasoning, code generation, knowledge QA and open-ended instruction following. The benefit is task dependent and smaller than in claim verification. It is clearest on HumanEval+ and AlpacaEval, where a pairwise method beats AbsScore for every generator (Table 3), and when the two select different responses the pairwise choice is usually the better one. The methods are nearly indistinguishable on MBPP+, results on mathematical reasoning and knowledge QA are mixed, and where final answers can be matched directly, majority voting is stronger than any confidence-based selection. The full setup and results are in Appendix G.

5.3 Additional Applications

Does PairScore generalize to other ranking problems? The first of two further applications is study selection for five systematic reviews spanning computer science, conservation, medicine, biology, and ecology (De Bruin et al., 2023; Hall et al., 2012; Walker et al., 2018; Leenaars et al., 2020; Bladon et al., 2022; Chausson et al., 2020). PairScore matches or exceeds AbsScore in most settings, where PairChoice usually falls short (Table 4). Second, on research-proposal ranking, we evaluate 656 within-topic comparisons among 93 proposals with expert quality scores (Si et al., 2025). With terse single-order prompts to Claude Sonnet 4.5, PairScore improves pairwise ranking accuracy from 50.6% with PairChoice to 60.7%; with reasoning space and both orders queried, the two are within two points (Appendix H).

6 Conclusion

We introduced **PairScore**, which elicits confidence by scoring two items jointly rather than independently (AbsScore) or through a forced choice (PairChoice). On controlled pairs, a forced choice answers by position when a tie is correct and loses discrimination against extreme-confidence partners; joint scoring largely avoids both effects. Across our experiments, PairScore often improves confidence ranking and selection, with particularly clear benefits where absolute scores provide limited discrimination. It is also better calibrated than either alternative, and its gain holds as more items are scored together, whereas asking for a full ordering peaks early and degrades. This study is limited to verbalized confidence from LLMs, suitable for black-box models. Future work can explore mechanistic understanding of these methods.

AI USE STATEMENT

In this work, we used generative AI tools for code, for editing text, for creating figures, and for checking citations. We have not used generative AI tools to originate the research questions, to design the experiments, to generate or fabricate experimental results, or to select which results to report; the remaining tasks in the ICLR disclosure list are not applicable to this work. We have reviewed all AI-assisted work. We take responsibility for the final content of this work, including all text, claims, and artifacts.

REFERENCES

Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, and Charles Sutton. Program synthesis with large language models. *arXiv preprint arXiv:2108.07732*, 2021.

Andrew J. Bladon, Rebecca K. Smith, and William J. Sutherland. *Butterfly and Moth Conservation: Global Evidence for the Effects of Interventions for Butterflies and Moths*. Conservation Evidence, University of Cambridge, 2022.

Ralph Allan Bradley and Milton E Terry. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39(3/4):324–345, 1952.

Glenn W. Brier. Verification of forecasts expressed in terms of probability. *Monthly Weather Review*, 78(1):1–3, 1950.

Alexandre Chausson, Beth Turner, Dan Seddon, Nicole Chabaneix, Cécile A. J. Girardin, Valerie Kapos, Isabel Key, Dilys Roe, Alison Smith, Stephen Woroniecki, and Nathalie Seddon. Mapping the effectiveness of nature-based solutions for climate change adaptation. *Global Change Biology*, 26(11):6134–6155, 2020. doi: 10.1111/gcb.15310.

Jiefeng Chen, Jinsung Yoon, Sayna Ebrahimi, Sercan Arik, Tomas Pfister, and Somesh Jha. Adaptation with self-evaluation to improve selective prediction in LLMs. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 5190–5213, 2023.

Paul F. Christiano, Jan Leike, Tom B. Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. In *Advances in Neural Information Processing Systems*, 2017.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

Jonathan De Bruin, Yongchao Ma, Gerbrich Ferdinands, Jelle Teijema, and Rens Van de Schoot. SYNERGY – open machine learning dataset on study selection in systematic reviews, 2023.

Yann Dubois, Balázs Galambosi, Percy Liang, and Tatsunori B. Hashimoto. Length-controlled AlpacaEval: A simple way to debias automatic evaluators. In *First Conference on Language Modeling (COLM)*, 2024.

Juraj Gottweis, Wei-Hung Weng, Alexander Daryin, Tao Tu, Petar Sirkovic, Artiom Myaskovsky, Grzegorz Glowaty, Felix Weissenberger, Alessio Orlandi, Dan Popovici, et al. Accelerating scientific discovery with co-scientist. *Nature*, 655(8122):487–496, 2026.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *International Conference on Machine Learning (ICML)*, pp. 1321–1330, 2017.

Neha Gupta, Harikrishna Narasimhan, Wittawat Jitkrittum, Ankit Singh Rawat, Aditya Krishna Menon, and Sanjiv Kumar. Language model cascades: Token-level uncertainty and beyond. In *International Conference on Learning Representations (ICLR)*, 2024.

Tracy Hall, Sarah Beecham, David Bowes, David Gray, and Steve Counsell. A systematic literature review on fault prediction performance in software engineering. *IEEE Transactions on Software Engineering*, 38(6):1276–1304, 2012. doi: 10.1109/TSE.2011.103.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the MATH dataset. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

Mandar Joshi, Eunsol Choi, Daniel S. Weld, and Luke Zettlemoyer. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1601–1611. Association for Computational Linguistics, 2017. doi: 10.18653/v1/P17-1147.

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. Language models (mostly) know what they don't know. *arXiv preprint arXiv:2207.05221*, 2022.

Michael Kirchhof, Luca Füger, Adam Goliński, Eeshan Gunesh Dhekane, Arno Blaas, Seong Joon Oh, and Sinead Williamson. SelfReflect: Can LLMs communicate their internal answer distribution? In *International Conference on Learning Representations (ICLR)*, 2026.

Jannik Kossen, Jiatong Han, Muhammed Razzak, Lisa Schut, Shreshth Malik, and Yarin Gal. Semantic entropy probes: Robust and cheap hallucination detection in LLMs. *arXiv preprint arXiv:2406.15927*, 2024.

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

Cathalijn H. C. Leenaars, Frans R. Stafleu, David de Jong, Maikel van Berlo, Tijmen Geurts, Tineke Coenen-De Roo, Jan-Bas Prins, Rosalie Kempkes, Janneke Elzinga, André Bleich, Rob B. M. de Vries, Franck Meijboom, and Merel Ritskes-Hoitinga. A systematic review comparing experimental design of animal and human methotrexate efficacy studies for rheumatoid arthritis: Lessons for the translational value of animal studies. *Animals*, 10(6):1047, 2020. doi: 10.3390/ani10061047.

Stephanie Lin, Jacob Hilton, and Owain Evans. TruthfulQA: Measuring how models mimic human falsehoods. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3214–3252. Association for Computational Linguistics, 2022. doi: 10.18653/v1/2022.acl-long.229. URL https://aclanthology.org/2022.acl-long.229/.

Jiawei Liu, Chunqiu Steven Xia, Yuyao Wang, and Lingming Zhang. Is your code generated by ChatGPT really correct? rigorous evaluation of large language models for code generation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Yinhong Liu, Han Zhou, Zhijiang Guo, Ehsan Shareghi, Ivan Vulić, Anna Korhonen, and Nigel Collier. Aligning with human judgement: The role of pairwise preference in large language model evaluators. In *Conference on Language Modeling (COLM)*, 2024.

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

Mahdi Pakdaman Naeini, Gregory F. Cooper, and Milos Hauskrecht. Obtaining well calibrated probabilities using Bayesian binning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 29, 2015.

Yasumasa Onoe, Michael J.Q. Zhang, Eunsol Choi, and Greg Durrett. CREAK: A dataset for commonsense reasoning over entity knowledge. In *NeurIPS Datasets and Benchmarks Track*, 2021.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35, 2022.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. GPQA: A graduate-level Google-proof Q&A benchmark. In *First Conference on Language Modeling (COLM)*, 2024.

Vaishnavi Shrivastava, Percy Liang, and Ananya Kumar. Llamas know what gpts don't show: Surrogate models for confidence estimation. *arXiv preprint arXiv:2311.08877*, 2023.

Vaishnavi Shrivastava, Ananya Kumar, and Percy Liang. Language models prefer what they know: Relative confidence estimation via confidence preferences. *arXiv preprint arXiv:2502.01126*, 2025.

Chenglei Si, Diyi Yang, and Tatsunori Hashimoto. Can llms generate novel research ideas? a large-scale human study with 100+ nlp researchers. In *International Conference on Learning Representations (ICLR)*, 2025.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. FEVER: a large-scale dataset for fact extraction and VERification. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, 2018.

Louis Leon Thurstone. A law of comparative judgment. *Psychological Review*, 34(4):273–286, 1927.

Katherine Tian, Eric Mitchell, Allan Zhou, Archit Sharma, Rafael Rafailov, Huaxiu Yao, Chelsea Finn, and Christopher D. Manning. Just ask for calibration: Strategies for eliciting calibrated confidence scores from language models fine-tuned with human feedback. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2023.

Vickie R. Walker, Abee L. Boyles, Katherine E. Pelch, et al. Human and animal evidence of potential transgenerational inheritance of health effects: An evidence map and state- of-the-science evaluation. *Environment International*, 115:48–69, 2018. doi: 10.1016/j.envint.2017.12.032.

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

Yuxi Xia, Dennis Ulmer, Terra Blevins, Yihong Liu, Hinrich Schütze, and Benjamin Roth. Calibration is not enough: Evaluating confidence estimation under language variations. *arXiv preprint arXiv:2601.08064*, 2026.

Miao Xiong, Zhiyuan Hu, Xinyang Lu, YIFEI LI, Jie Fu, Junxian He, and Bryan Hooi. Can LLMs express their uncertainty? an empirical evaluation of confidence elicitation in LLMs. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=gjeQKFxFpZ.

Daniel Yang, Yao-Hung Hubert Tsai, and Makoto Yamada. On verbalized confidence scores for LLMs. *arXiv preprint arXiv:2412.14737*, 2024.

Boya Zhang, Alban Bornet, Anthony Yazdani, Philipp Khlebnikov, Marija Milutinovic, Hossein Rouhizadeh, Poorya Amini, and Douglas Teodoro. A dataset for evaluating clinical research claims in large language models. *Scientific Data*, 12(1):86, 2025.

A Datasets and Models

A.1 Datasets

Claim verification. Table 5 lists the four claim-verification datasets. Each is subsampled to n=500 items using a fixed random seed (seed 42), stratified to preserve the original class ratio where possible.

Best-of-N. Table 6 lists the eight best-of-N benchmarks. For benchmarks larger than our target n, we take the first n problems in the canonical ordering (no random subsampling, to allow reproducibility).

Dataset	Domain	T	F
CREAK (Onoe et al., 2021)	Entity facts	242	258
FEVER (Thorne et al., 2018)	Wikipedia	252	248
TruthfulQA (Lin et al., 2022)	Adversarial QA	96	404
CliniFact (Zhang et al., 2025)	Clinical claims	140	360

Table 5: Claim-verification datasets. T/F = True/False label counts in our 500-item subsample.

Dataset	Category	n
GSM8K (Cobbe et al., 2021)	Math reasoning	200
MATH-500 (Hendrycks et al., 2021)	Math competition	200
AIME 2024	Math olympiad	90
HumanEval+ (Liu et al., 2023)	Code generation	164
MBPP+ (Austin et al., 2021)	Code generation	200
GPQA Diamond (Rein et al., 2024)	Science QA	198
TriviaQA (Joshi et al., 2017)	Factual QA	200
AlpacaEval (Dubois et al., 2024)	Open-ended	200

Table 6: Best-of-N datasets. All use N=8 responses per problem.

A.2 Model Details

API models. Table 7 lists the exact API identifiers used via OpenRouter.

Label	API identifier	Used for
GPT-5-mini	`openai/gpt-5-mini`	Claim verif + BoN
GPT-4o-mini	`openai/gpt-4o-mini`	Claim verif + BoN
GPT-4.1-nano	`openai/gpt-4.1-nano`	Claim verif + BoN
Gemini-2.5-FL	`google/gemini-2.5-flash-lite`	Claim verif + BoN

Table 7: API models and identifiers. FL = Flash-Lite.

All API calls were made through OpenRouter (`https://openrouter.ai`). For AbsScore, we use temperature 0.0 for the deterministic call and temperature 0.7 for the four stochastic repetitions. For pairwise methods, all comparisons use temperature 0.0 and each comparison is a single call. The order of the two items in the prompt is randomized independently for each comparison to debias position effects. For PairChoice, a reply that does not name exactly one item is retried once and otherwise discarded; for PairScore, scores are mapped back to the original item order.

Local models. Table 8 lists the local models and the experiments each appears in. Models up to 4B parameters run in `bfloat16`; the two larger models run in 4-bit `nf4`, on every dataset for a given model, so no model's results mix precisions. All runs use a single NVIDIA RTX 2080 Ti (11 GB) with the HuggingFace `transformers` library, single-GPU autoregressive decoding at batch size 1. Decoding is greedy (temperature 0) except where a protocol samples repetitions, namely the repeated AbsScore calls and the fixed-partner experiment, which use temperature 0.7 for the sampled repetitions exactly as for the API models.

HuggingFace ID	Params	Prec.	Used for
`Qwen/Qwen3-1.7B`	1.7B	bf16	CV, D, C
`Qwen/Qwen2.5-3B-Instruct`	3B	bf16	CV, D, C
`microsoft/Phi-4-mini-instruct`	3.8B	bf16	CV, D, C, P, K
`Qwen/Qwen3-4B`	4B	bf16	CV, D, C, P, K
`Qwen/Qwen2.5-7B-Instruct`	7B	nf4	D, C
`Qwen/Qwen3-8B`	8B	nf4	D, C

Table 8: Local models. CV = claim verification (Table 2); D = diagnostics and tie option (Section 2); C = calibration (Figure 4); P = fixed partners (Figure 2); K = k-wise (Figure 5).

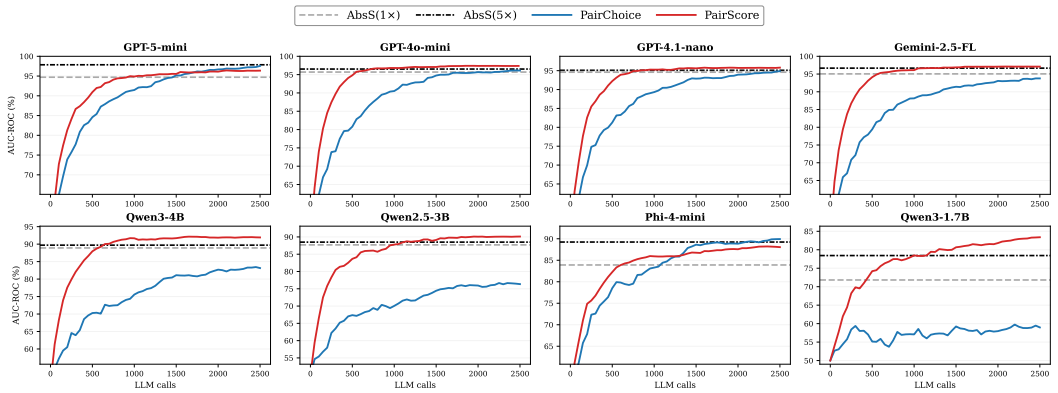


Figure 6: Convergence on CREAK. All 8 models; gray dashed = AbsScore(1×); black dash-dot = AbsScore(5×).

B Algorithm Pseudocode

Algorithm 3 AbsScore (Repeated Numerical Scoring)

Require: Items $x_1, \ldots, x_n$; repetitions R; temperature τ
Ensure: Confidence scores $s_1, \ldots, s_n$

for $i = 1$ to n **do**
  $s_i^{(1)} \leftarrow \textsc{AbsScoreLLM}(x_i, \text{temp}{=}0)$ ▷ *deterministic*
end for
for $r = 2$ to R **do**
  for $i = 1$ to n **do**
    $s_i^{(r)} \leftarrow \textsc{AbsScoreLLM}(x_i, \text{temp}{=}\tau)$
  end for
end for
return $s_i = \frac{1}{R}\sum_{r=1}^{R} s_i^{(r)}$ [nR calls]

C Claim Verification: Additional Results

C.1 Convergence: All Models

Figures 6–9 show AUC-ROC (%) vs. LLM calls for all 8 models on each dataset (4 models per row, 2 rows per figure). The pattern is consistent across model families: PairScore typically reaches or exceeds the AbsScore(5×) baseline within 500–1000 calls, with the largest gains for the weakest models (Qwen3-1.7B); PairChoice does so less often.

Note: The main experiments in Section 5 use $N{=}500$ items per dataset, 8 models, and the $[0, 100]$ scale with AbsScore, PairChoice, and PairScore. The full AUC-ROC numbers for all eight models are in Table 2.

C.2 Token Efficiency

Table 9 reports, for each dataset and pairwise method, the fraction of (dataset, model) settings where the method eventually exceeds AbsScore(5×) AUC, and the mean token count at first crossover as a fraction of the AbsScore(5×) token budget. Token counts for numerical calls are exact (from `prompt_tokens` + `completion_tokens` in the call log); pairwise token counts are estimated via `tiktoken` (cl100k_base) from the item texts and prompt templates, with fixed completion token budgets (2 for PairChoice, 12 for PairScore).

The key takeaway: despite consuming ∼1.0× (PairChoice) or ∼1.4× (PairScore) the tokens of AbsScore(5×) in total, pairwise methods reach the AbsScore(5×) threshold at a fraction of those

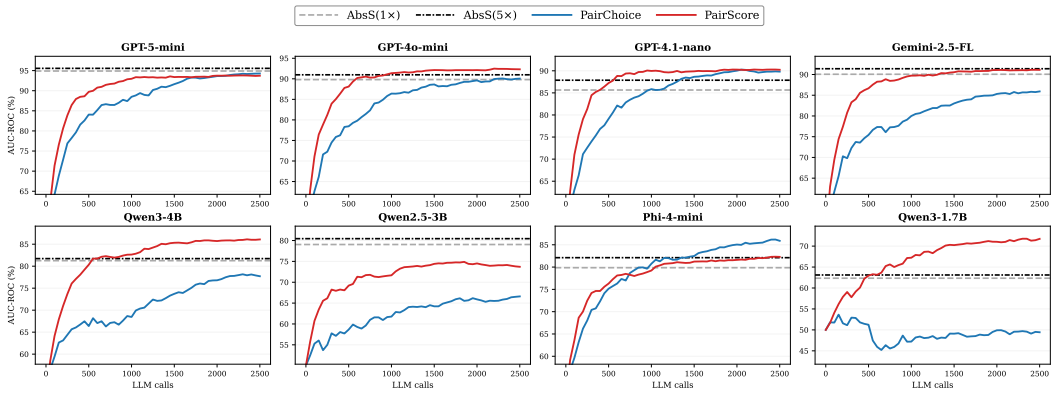


Figure 7: Convergence on FEVER.

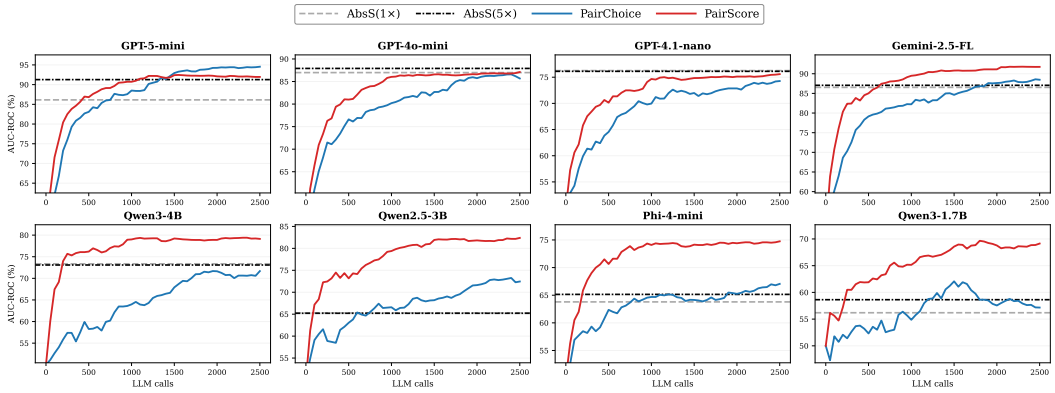


Figure 8: Convergence on TruthfulQA.

tokens when they do exceed it. PairScore is most consistent, exceeding the AbsScore(5×) AUC in 74% of settings at 0.65× of its token budget.

C.3 Anecdotal Examples

Table 10 shows concrete cases where PairScore separates claims that AbsScore ties.

Claim verification. Table 10 shows pairs where AbsScore assigns identical scores to a true and false claim, but PairScore correctly separates them. A recurring pattern is *floor collapse*: AbsScore outputs 0 for both items (the model declines to commit to either), but when forced to compare, it correctly identifies the true claim. A second pattern is *midpoint collapse*: AbsScore outputs 50 for both items (maximum uncertainty), but the comparison resolves the ambiguity.

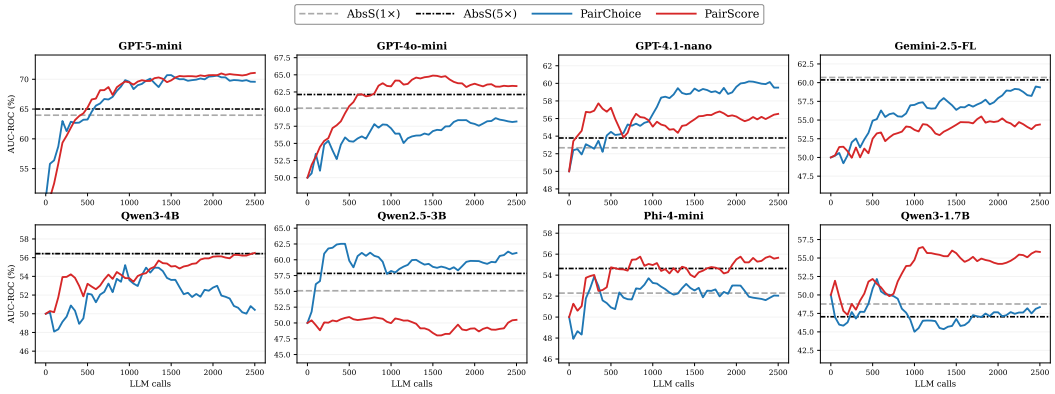


Figure 9: Convergence on CliniFact.

		Exceeds AbsS(5×)		Token ratio at crossover	
Dataset	**Method**	**Count**	**%**	**Mean**	**(vs. AbsS)**
CREAK	PairChoice	1/6	17%	134K	0.82×
	PairScore	5/6	83%	101K	0.64×
FEVER	PairChoice	2/7	29%	116K	0.72×
	PairScore	5/7	71%	116K	0.73×
TruthfulQA	PairChoice	3/7	43%	151K	0.81×
	PairScore	5/7	71%	94K	0.53×
CliniFact	PairChoice	3/7	43%	185K	0.79×
	PairScore	5/7	71%	178K	0.71×
Overall	PairChoice	9/27	33%	147K	0.78×
	PairScore	20/27	74%	123K	0.65×

Table 9: Token efficiency of pairwise methods relative to AbsScore(5×). "Exceeds" = fraction of (dataset, model) settings where the method eventually surpasses AbsScore(5×) AUC. "Token ratio" = mean tokens at first crossover ÷ AbsScore(5×) token budget (lower is more efficient; entries where the method never exceeds are counted at total token usage). Settings without complete token logs are excluded, leaving 27 of 32 settings.

True claim	**False claim**	**AbsS (T/F)**	**PS (T/F)**
Nancy Pelosi followed Paul Ryan as Speaker of The House.	Richard Linklater directed *Gone with the Wind*.	0 / 0	84 / 0
Annie is a movie.	Benjamin Franklin was born in 1790.	0 / 0	96 / 0
ZX008 0.5 mg/kg/day is superior to placebo in reducing convulsive seizures [correct clinical claim].	Umooze is superior to placebo on QoL index [incorrect clinical claim].	50 / 50	74 / 8

Table 10: Claim verification examples where AbsScore ties but PairScore correctly separates (GPT-4o-mini). AbsS = AbsScore(5×) score; PS = PairScore.

D TWO DIAGNOSTICS: PROTOCOLS AND FULL RESULTS

This appendix gives the protocols and full results behind Section 2: the diagnostic pairs of Figure 1, the fixed-partner experiment of Figure 2, and the effect of joint scoring on items whose absolute scores coincide.

D.1 DIAGNOSTIC PAIRS

Why hand-built pairs. The claim under test is about what a format can represent, not how accurately a model ranks real claims. An AUC-ROC on a benchmark cannot separate the two. The diagnostic therefore uses claims whose truth values are settled by construction and varies only the presentation.

Claims. Statements of the form "The city of X is in Y." from the `cities` corpus of Marks & Tegmark (2024), each with a ground-truth label. A second bank of arithmetic statements, $x + y = z$ with z correct or off by one to three, gives the same picture; the main text reports the cities bank.

Strata. Pairs are assigned to six strata by their truth labels *after* they are built. The labels are never shown to the model and never used to choose a pairing.

- **One true, one false** (60 pairs). Does the format track content when content differs?
- **Both true** (60). Can the format say that both look right?
- **Both false** (60). Can it say that both look wrong?
- **The same claim twice** (120). Collected but not reported: with one distinct claim there is nothing for a different winner to be.
- **A claim beside an unrelated true partner** (120). Control: does an irrelevant partner move the claim's score?
- **A claim alone under the pair rubric** (120). Control: does the wording of the pair prompt move it, without any partner?

Elicitation. Every pair is run in both orders, and the winner is recorded by claim identity rather than by the letter it was printed under. The prompts are those of the main experiments with one difference for PairScore, "Provide a score for each statement on a scale of 0 to 100", so the same rubric can be issued for one statement in the control condition ("a claim alone under the pair rubric", with output format "A: [score]"). The tie condition adds one line to the PairChoice prompt and changes nothing else, under two wordings; the main text reports the wording with the lower order dependence, the comparison most favourable to the forced choice. All prompts are given verbatim in Appendix I.1.

Decoding and parsing. Greedy decoding throughout, so the measured effect is a property of the mode of the output distribution and not of sampling noise. Token budgets are 48 for AbsScore, 128 for PairScore and 32 for PairChoice, with one retry at eight times the budget, because small instruction-tuned models often spend the first budget on a preamble. A response that cannot be parsed is recorded as a failure and excluded; nothing is imputed.

Metric. Each pair falls into one of five states, defined the same way for every format: *consistent* (the same claim wins in both orders), *tie* (no preference either way), *mixed* (a preference one way and a tie the other), *flipped* (the preference reverses) and *unparsed*. Order dependence is the share of flipped plus mixed among pairs answered in both orders. Mixed is counted because, once a tie option exists, answering "tie" in one order and naming a winner in the other is still the order deciding the answer, and counting reversals alone would flatter the tie condition.

Models. Six open-weight instruction-tuned models: Qwen3-1.7B, Qwen2.5-3B-Instruct, Phi-4-mini-instruct, Qwen3-4B, Qwen2.5-7B-Instruct and Qwen3-8B. Models of 7B parameters and above run in 4-bit `nf4` on a single 11 GB card, on every bank, so no model's row mixes precisions.

Dataset	AbsScore bin	n	AbsS	PS(low)	PS(med)	PS(high)
CREAK	0–20	49	2.7	9.8 ± 15.1	12.1 ± 13.0	11.2 ± 13.9
	60–80	4	71.2	74.2 ± 4.0	73.4 ± 5.4	71.0 ± 6.5
	80–100	40	93.1	91.3 ± 5.9	91.1 ± 5.3	92.2 ± 6.1
TruthfulQA	0–20	48	3.5	17.7 ± 15.7	17.2 ± 13.2	19.6 ± 15.5
	20–40	20	26.0	43.5 ± 17.6	42.4 ± 16.6	43.5 ± 14.8
	80–100	20	89.5	85.3 ± 7.0	84.0 ± 13.6	85.7 ± 11.2
CliniFact	40–60	60	50.0	56.1 ± 8.4	62.5 ± 6.8	64.5 ± 6.1
	60–80	38	70.4	68.2 ± 6.2	72.2 ± 4.5	72.9 ± 3.6

Table 11: Mean PairScore (± std. dev.) when items are paired with low-, medium-, and high-confidence anchors (GPT-4o-mini). Shifts across anchor levels are small relative to the differences between items.

Controls. PairScore differs from AbsScore in two ways at once: a second claim, and a differently worded instruction. Two controls separate them. Scoring a claim alone under the pair rubric, and scoring it beside an unrelated true statement, both leave AUC-ROC within 0.02 of plain single-claim scoring on every model. The gain therefore comes from a partner that carries information, not from the presence of a partner or the wording of the prompt.

D.2 Fixed partners

Partners at three confidence levels. A first test, on GPT-4o-mini, pairs each item with anchors drawn from low-, medium- and high-confidence groups (100 items per dataset, 10 anchors per level). Table 11 reports the mean PairScore under each. The score moves somewhat with the anchor's confidence level, but the shifts are modest against the differences between items. The fixed-partner experiment below is the sharper version of the same question: it holds the partner completely fixed, gives it no truth value or an obvious one, and compares PairScore with PairChoice under the same conditions.

Protocol. We take 250 claims each from CREAK and Climate-FEVER and pair every claim with the same partner text, so the only thing that varies across conditions is the partner. The five partners, verbatim:

Fixed partners

```
Lorem ipsum dolor sit amet, consectetur adipiscing elit.
Colorless green ideas sleep furiously beside the square circle.
947 382 blx qtm fgh 2846 rnp wvz 153.
The Earth orbits the Sun.
Two plus two equals five.
```

The first three carry no truth value: a Latin placeholder, a grammatical sentence with no meaning, and a string of random tokens. The last two are plain statements that any model marks true and false respectively. Each call uses the PairScore or PairChoice prompt of Appendix I with the claim in one slot and the partner in the other; which slot is chosen at random per call, so neither format can answer by position. Each claim is scored three times, once greedily and twice at temperature 0.7, and the scores are averaged. PairScore reads the claim's own score. PairChoice records whether the model prefers the claim to the partner, and the claim's score is the fraction of calls it wins. AUC-ROC is computed against the ground-truth labels over the claims scored in every condition. Models: Qwen3-4B and Phi-4-mini-instruct, both in bf16.

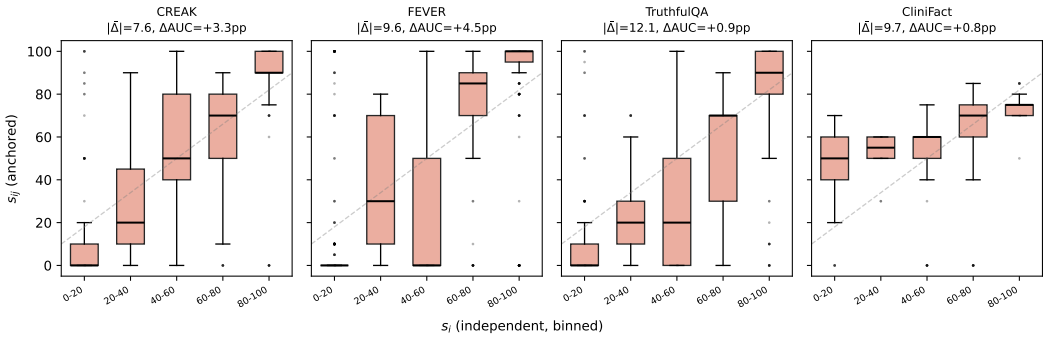


Figure 10: Joint scores vs. independent AbsScores, by dataset (GPT-4o-mini, 500 pairs per dataset). Items are grouped by their independent AbsScore (x-axis); boxes show the distribution of scores the same items receive when scored jointly with a partner. Wide boxes indicate that joint scoring spreads out items that AbsScore assigns nearly identical scores; the annotated mean absolute shift and AUC change are given per dataset (AUC improves on all four).

Results. If a format measures the claim, its AUC-ROC should not depend on the partner. Across the five partners PairScore moves by 0.02–0.06 AUC-ROC and PairChoice by 0.14–0.23. The meaningless partners leave PairChoice close to PairScore; the obviously true partner pulls it toward chance, and the obviously false partner pulls it part of the way. A forced choice against a partner the model is sure of is pulled toward the partner's truth value, and the claim's own merit counts for less.

D.3 Joint scoring and collapsed absolute scores

A recurring limitation of AbsScore is *score collapse*: the model assigns identical or near-identical confidence scores to items that differ in correctness. True and false claims can therefore receive the same score. Concrete examples are given in Appendix C.3. We test whether joint scoring separates such items by comparing the score an item receives in isolation with the score the same item receives when scored jointly with another item. Using GPT-4o-mini, we score 500 pairs per dataset both independently and jointly. Figure 10 groups items by their independent AbsScore and shows the distribution of their joint scores. Joint scoring spreads out items that AbsScore places at the extremes or at the midpoint, and the spread is useful: the joint scores achieve higher AUC-ROC than the independent scores on all four datasets.

E CALIBRATION: PROTOCOL AND PER-DATASET RESULTS

Metrics. For measuring calibration properties we ensure the scores are normalized to the range $[0, 1]$. The Brier score (Brier, 1950) is $\frac{1}{n}\sum_i (s_i - y_i)^2$, here y_i is the groundtruth binary label of claim i. Expected calibration error (ECE; Naeini et al., 2015; Guo et al., 2017) is computed with ten bins under two binnings: equal-width bins over $[0, 1]$, and equal-mass bins at the deciles of the predicted distribution. We report both because ECE is known to move with the binning, and because verbalized scores concentrate at the endpoints of the scale, which is precisely the case in which equal-width bins can be misleading. Table 12 reports both, with the Brier score; the two binnings give the same ordering of methods on every dataset, and the main text uses equal-mass.

Coverage. The same 6 open-weight models as the diagnostic pairs of Appendix D, on CREAK, FEVER and TruthfulQA, every cell complete for all three formats. Cells in which fewer than 50 items produced a score would be dropped; none were. The per-dataset results are in Figure 4 in the main text.

Dataset	Format	ECE (equal-mass)	ECE (equal-width)	Brier
CREAK	AbsScore (5x)	0.117	0.115	0.144
	PairScore	**0.071**	**0.072**	**0.117**
	PairChoice	0.089	0.091	0.155
FEVER	AbsScore (5x)	0.168	0.175	0.209
	PairScore	0.132	0.139	**0.189**
	PairChoice	**0.123**	**0.130**	0.201
TruthfulQA	AbsScore (5x)	0.353	0.354	0.313
	PairScore	**0.289**	**0.288**	**0.250**
	PairChoice	0.300	0.303	0.265

Table 12: Calibration by dataset, averaged over the 6 open-weight models. Expected calibration error under equal-mass and equal-width binning (ten bins each) and the Brier score; lower is better, best per column in bold. PairChoice is read through $\sigma(\theta)$ with no recalibration.

F k-WISE ELICITATION: FULL PROTOCOL AND RESULTS

Groups. Each of R=5 rounds shuffles all N=500 items and cuts them into consecutive blocks of k. Every item therefore appears in exactly five groups at every k, so the number of times an item is scored does not change with k and coverage cannot be confounded with the quantity under study. The number of calls is $5\lceil N/k \rceil$, from 2500 at k=1 to 125 at k=20. The order of items within a group is reshuffled on every call.

Prompts. k-score asks for a score for each of the k claims, listed under labels `A`–`T`; at k=1 it reduces to the AbsScore prompt. k-rank asks for the claims ranked from most to least likely to be correct as comma-separated letters; at k=2 it reduces to the binary PairChoice prompt. The endpoints of the sweep are therefore exactly the formats compared elsewhere in the paper. Both prompts are given verbatim in Appendix I.2.

Aggregation. A k-score item takes the mean of its five scores. A k-rank ordering is expanded into all $\binom{k}{2}$ pairwise outcomes and aggregated with Bradley–Terry.

Decoding and parsing. Greedy decoding at temperature 0. Token budgets are $\max(160, 24k)$ for k-score and $\max(64, 8k)$ for k-rank, each with one retry at three times the budget. A response that cannot be parsed is recorded as a failure and excluded; nothing is imputed. Per-run parse success and item coverage are stored alongside every result. Seed 42, single seed.

Cost and parse rate. Prompt tokens per item fall with k for both formats, since the instruction is shared across the block; for k-score they drop from 411 at k=1 to 311 at k=5 and 214 at k=20.

Answer tokens are where the formats differ: k-rank returns a few letters, k-score returns k numbers. Parse rate also differs. At k=20, 99% of k-score calls return a usable answer against 71% for k-rank, which accounts for part of the k-rank decline in Figure 5.

G BEST-OF-N SELECTION: SETUP AND FULL RESULTS

This appendix gives the full best-of-N setup and results summarised in Section 5.2.

We next ask whether the elicited confidence scores are useful for selecting among multiple candidate responses. For each problem, we generate $N=8$ responses at temperature 0.7 and use each method to select one response.

Setup. We compare Random selection, AbsScore, PairChoice-Elim (PC-Elim), PairScore-Elim (PS-Elim), and an Oracle upper bound. AbsScore scores each candidate independently and selects the highest-scoring response, while PC-Elim and PS-Elim use $N-1$ pairwise comparisons to select among the candidates. All confidence-based selection is performed by the same model that generated the responses; no external model is used for selection.

Our main evaluation spans seven benchmarks (GSM8K, the eighth in Table 6, appears only in the majority-vote comparison) covering mathematical reasoning, code generation, knowledge QA, and open-ended instruction following. We evaluate GPT-4o-mini, GPT-5-mini, and Gemini-2.5-Flash-Lite; results for GPT-4.1-nano are reported in Appendix G.1. Full benchmark details are given in Table 6. For mathematical reasoning, where final answers can also be matched directly, we separately compare against majority-vote baselines below.

When does pairwise selection help? The benefit of pairwise selection is strongly task-dependent (Table 14). The clearest gains appear on HumanEval+ and AlpacaEval. On HumanEval+, a pairwise method outperforms AbsScore for each of the three generator models, with particularly large gains for Gemini-2.5-Flash-Lite. On AlpacaEval, pairwise selection also improves the average judge score for all three models; moreover, when AbsScore and a pairwise method select different responses, the pairwise-selected response is preferred in 70–88% of the non-tied comparisons (Table 13). In contrast, the methods are nearly indistinguishable on MBPP+, while results on mathematical reasoning and knowledge QA are more mixed.

Generator	Method	AbsS wins	Method wins	Tie
GPT-4o-mini	PC-Elim	6.5%	15.5%	78.0%
	PS-Elim	5.0%	22.5%	72.5%
GPT-5-mini	PC-Elim	4.0%	23.0%	73.0%
	PS-Elim	6.0%	23.0%	71.0%
Gemini-2.5-FL	PC-Elim	5.0%	18.5%	76.5%
	PS-Elim	2.5%	19.0%	78.5%

Table 13: Head-to-head: AbsScore vs. pairwise (AlpacaEval, n=200, GPT-4o-mini judge).

What happens when answers can be directly matched? For mathematical reasoning, the candidate responses provide an additional signal: their final answers can be matched directly. We therefore compare confidence-based selection against Majority Vote (MV) and confidence-weighted variants (Table 15). In MV, responses vote for their final answer and the most frequent answer is selected; in weighted MV, these votes are weighted using either AbsScore (wMV-AbsS) or PairChoice-Elim confidence (wMV-PCe). Across the math benchmarks, majority voting is generally the strongest non-oracle baseline, while confidence weighting provides little additional benefit. Pairwise selection can still improve over AbsScore on the harder problems, but when answer agreement is available, it provides a weaker signal than directly aggregating the candidate answers.

		Math		*Code*		*Knowledge QA*		*Open*
Model	**Method**	**Math 500**	**AIME**	**HuEval+**	**MBPP+**	**GPQA**	**Trivia QA**	**Alpaca**
GPT-4o-mini	Random	66.8	8.9	64.6	45.4	29.3	64.0	3.52
	AbsScore	71.2	11.1	68.3	**46.6**	33.3	66.0	3.88
	PC-Elim	**72.0**	8.9	67.7	45.4	**34.3**	**71.0**	4.19
	PS-Elim	71.8	**12.2**	**69.5**	46.2	33.3	67.0	**4.43**
	Oracle	82.2	14.4	76.2	51.4	71.2	89.0	6.18
GPT-5-mini	Random	74.0	40.0	75.0	**51.5**	**64.1**	85.5	3.68
	AbsScore	75.5	46.7	73.8	51.0	63.1	85.5	3.96
	PC-Elim	**76.5**	**50.0**	**81.1**	51.0	**64.1**	81.0	**4.96**
	PS-Elim	75.5	46.7	79.9	**51.5**	58.1	**86.0**	4.52
	Oracle	82.5	56.7	84.8	54.5	86.4	92.5	6.36
Gemini-2.5-FL	Random	54.0	7.8	61.6	**52.0**	28.8	63.5	3.81
	AbsScore	55.0	11.1	57.3	50.0	32.8	64.5	3.50
	PC-Elim	**58.0**	**15.6**	**72.0**	51.5	**33.8**	**65.5**	3.98
	PS-Elim	55.5	13.3	69.5	50.0	30.8	62.0	**4.56**
	Oracle	65.5	18.9	80.5	57.5	66.7	87.0	6.01

Table 14: Best-of-8 selection. Accuracy (%) for math, code, and knowledge QA; average GPT-4o-mini judge score (0–7 scale) for AlpacaEval (open-ended instruction following). AbsScore rates each response once on 0–100 and picks the highest score (single call per response; no averaging over repetitions, unlike claim verification). **Bold** = best non-oracle per (model, dataset). HuEval+ = HumanEval+. GPT-4.1-nano results in Appendix G.1.

Dataset	Model	Rand	MV	AbsS	PCe	PSe	wMV AbsS	wMV PCe	Orac
GSM8K	GPT-4.1-nano	92.0	**95.5**	92.0	92.0	91.5	**95.5**	94.0	97.5
	GPT-4o-mini	**91.5**	90.0	89.0	88.5	89.0	89.5	89.5	96.0
	Gemini-2.5-FL	87.5	**91.5**	89.0	87.5	87.0	91.0	91.0	93.0
MATH-500	GPT-4.1-nano	59.0	71.5	64.0	64.0	66.0	**72.0**	70.0	75.0
	GPT-4o-mini	67.5	71.5	69.5	**72.0**	68.5	71.5	70.0	80.5
	GPT-5-mini	74.0	76.0	75.5	**76.5**	75.5	76.0	**76.5**	82.5
AIME	GPT-5-mini	40.0	**56.7**	46.7	50.0	46.7	55.6	**56.7**	56.7

Table 15: Best-of-8 accuracy (%) on math with Majority Vote. AbsS = AbsScore; PCe = PC-Elim; PSe = PS-Elim; Orac = Oracle (best of 8, upper bound); wMV-AbsS/PCe = weighted MV using AbsScore/PC-Elim confidence weights. **Bold** = best non-oracle.

G.1 GPT-4.1-NANO

Table 16 shows best-of-8 results for GPT-4.1-nano. The pattern mirrors the stronger models: pairwise methods gain most on code and open-ended tasks. AIME was not evaluated for nano.

	Math	*Code*		*Knowledge QA*		*Open*
Method	**Math 500**	**HuEval+**	**MBPP+**	**GPQA**	**Trivia QA**	**Alpaca**
Random	57.8	73.8	**48.0**	43.4	80.0	3.38
AbsScore	59.8	72.0	46.4	**52.0**	78.0	3.77
PC-Elim	61.2	**75.0**	47.2	47.5	**81.0**	4.38
PS-Elim	**62.6**	**75.0**	46.2	45.5	**81.0**	**4.70**
Oracle	73.4	81.1	51.8	85.4	92.0	6.38

Table 16: Best-of-8 selection for GPT-4.1-nano. Accuracy (%) for math/code/QA; avg. judge score (0–7) for AlpacaEval. AIME not evaluated for this model. **Bold** = best non-oracle.

G.2 EXAMPLES ON ALPACAEVAL

Tables 17–19 show three best-of-8 examples on AlpacaEval (GPT-5-mini generator), each illustrating a different failure mode of AbsScore.

Example 1: Explanation quality. The task is to classify a LinkedIn invitation email as Promotions or Social. All 8 responses give the correct label, but differ in whether they justify it. AbsScore assigns 100 to two responses and picks *[2]*, which gives the bare label with no explanation (judge = 0). Both pairwise methods select responses with clear justifications (judge = 6).

Example 2: Attribution. The task is to complete the aphorism "two things are infinite…" All 8 responses produce the correct quote, but only *[5]* includes the attribution to Albert Einstein. AbsScore assigns 100 to all responses and picks *[0]* (no attribution; judge = 3). Both pairwise methods select *[5]* (judge = 7).

Example 3: Completeness. The task is to state when the Yamato battleship was built. AbsScore assigns 100 to two responses and picks *[0]*: "The Yamato Battleship was built in 1941," which gives only the commissioning year (judge = 1). The more complete answer distinguishes the laying-down date (1937) from the commissioning date (1941); PairScore assigns it the highest mean score and PC-Elim selects it (judge = 5 or 7).

#	Response	AbsS	PS	Judge
0	Social. It's a LinkedIn connection request, not a promotional offer.	95	94.3	5
1	Social	95	84.3	2
2	Social	**100**	89.3	**0 ←AbsS**
3	Social	95	87.9	1
4	Social	95	87.1	3
5	Social. It's a LinkedIn connection invitation, not a promotional message.	**100**	95.0	6
6	Social. It's a LinkedIn connection request: a personal networking invitation, not a promotion.	95	95.0	6 ←**PCe**
7	Social. It's a LinkedIn network invitation (personal/social networking), not a promotional message.	95	95.3	5 ←**PSe**

Table 17: AlpacaEval example 1 (GPT-5-mini). **Prompt:** "Decide if the given email belongs to the Promotions or Social category. Subject: You have an invitation. Email: Hi, I'd like to join your LinkedIn network." AbsS scores are 95 except where shown. Judge scores on 0–7.

#	Response	PS	Judge
0	"two things are infinite: the universe and human stupidity; and I'm not sure about the universe."	98.6	3 ←**AbsS**
1	[same as 0]	97.1	4
2	[same as 0]	98.6	2
3	[same as 0, with quotation marks]	98.6	6
4	[same as 0]	98.6	2
5	"Two things are infinite: the universe and human stupidity; and I'm not sure about the universe." – Albert Einstein	**100.0**	**7 ←PCe, PSe**
6	[same as 0]	97.1	2
7	[same as 0]	98.6	2

Table 18: AlpacaEval example 2 (Gemini-2.5-FL). **Prompt:** "Complete the aphorism and write the original quote as the answer. two things are infinite" All AbsS scores are 100. Judge scores on 0–7.

#	Response	AbsS	PS	Judge
0	The Yamato Battleship was built in 1941.	**100**	80.0	**1 ←AbsS**
1	The Yamato battleship was commissioned in 1941. Construction began in 1937.	95	90.0	3
2	The Yamato was laid down in 1937 and commissioned in 1941. Construction began in 1937; it entered service in 1941.	95	**97.1**	**7 ←PSe**
3	The Yamato battleship was laid down in 1937 and commissioned in 1941.	95	97.1	6
4	The Yamato battleship was commissioned in 1941.	95	79.3	2
5	The Yamato Battleship was built in 1941.	**100**	85.0	0
6	The Yamato Battleship was laid down in 1937 and commissioned in 1941.	95	97.1	4
7	The Yamato battleship was laid down in 1937 and commissioned in 1941.	95	97.1	5 ←**PCe**

Table 19: AlpacaEval example 3 (Gemini-2.5-FL). **Prompt:** "What year was the Yamato Battleship built?" Judge scores on 0–7.

H Study Selection and Research-Proposal Ranking

We evaluate the same elicitation paradigms on two external applications:

1. **Study selection for systematic reviews.** We evaluate title-and-abstract screening, the high-volume first stage of study selection, in which reviewers exclude clearly irrelevant records before full-text assessment. We use Hall, Walker, and Leenaars from the SYNERGY study-selection benchmark (De Bruin et al., 2023; Hall et al., 2012; Walker et al., 2018; Leenaars et al., 2020), the Bladon butterfly-and-moth conservation synopsis (Bladon et al., 2022), and the Chausson Nature-based Solutions systematic map (Chausson et al., 2020). The labels indicate final inclusion in the corresponding evidence synthesis. Each method receives a budget of $5N$ calls: AbsScore queries each abstract five times, while PairChoice and PairScore sample $5N$ unique pairs uniformly without replacement. PairChoice outcomes are aggregated with a regularized Bradley–Terry model and PairScore with the per-item mean. The abstract sample is fixed and A/B order is randomized independently. Table 4 reports the saved $N = 500$ run for each API model (Bladon uses its full reconstructed corpus, $N = 457$).

   PairScore has smaller losses than PairChoice when pairing does not help, while retaining gains where AbsScore leaves headroom. For example, PairScore improves Walker/GPT-4o-mini from 79.1 to 83.1 AUC ($p = .026$, correlated DeLong test) and Leenaars/Gemini-FL from 76.1 to 78.8 ($p = .004$), but its small Chausson gains are not significant ($p = .606$ and .577). These domains also differ in prevalence, prompt length, and criterion complexity, so the results support transfer across applications but do not identify discipline itself as a causal factor.

2. **Research-proposal ranking.** We use the expert judgments released by Si et al. (2025): 93 human- and LLM-generated NLP proposals across seven topics, each assigned an overall-quality score by expert reviewers. We sample 656 within-topic pairs and evaluate Claude Sonnet 4.5 against the mean expert score. In the cheap, single-order, non-CoT regime, for pairs separated by at least one expert-score point ($n = 427$), terse PairScore attains 60.7% accuracy versus 50.6% for terse PairChoice; their win-rate rankings have within-topic Spearman correlations of .213 and .030, respectively. On the same pairs, terse PairChoice selects the second-presented proposal in all 656 calls, so randomized order turns this positional rule into chance-level accuracy.

   For the stricter CoT + order-counterbalanced control, we give both methods 2–4 sentences of reasoning space and query both A/B orders. PairChoice yields an order-consistent preference for 533/656 pairs (81.2%). On the 334 matched, non-tied pairs separated by at least one expert-score point, PairScore and PairChoice reach 58.7% and 57.5% accuracy, respectively (difference $+1.2$ points; 95% bootstrap CI $[-1.5, +3.9]$; McNemar $p = .503$). This control rules out a general accuracy advantage under equal scaffolding. It instead indicates that numerical pairwise scores are more robust in the terse, low-token regime, whereas binary comparison requires additional scaffolding and remains sensitive to order.

I Prompt Templates

AbsScore

```
How confident are you that the following statement is correct.
Statement: "{text}"
Provide a single number between 0 (certainly wrong) and 100 (certainly
correct). Output ONLY the number.
```

PairChoice

```
Which of the following statements are you more confident of being
correct?
A: "{text_a}"
B: "{text_b}"
Output ONLY 'A' or 'B'.
```

PairScore

```
How confident are you that each of the following statements is
factually correct? Provide a score on a scale of 0 to 100.
A: "{text_a}"
B: "{text_b}"
Output format: "A: [score], B: [score]"
```

Best-of-N, AbsScore: verifiable tasks (math, code, knowledge QA)

```
Here is a {task label} and a proposed answer.
Question: {question}
{Answer/Solution}: {response}
How confident are you that {quality condition}? Output a single
integer between 0 (certainly wrong) and 100 (certainly correct).
Output ONLY the number.
```

Best-of-N, AbsScore: open-ended instruction following

```
Here is an instruction and a response. Rate the response quality as a
percentage from 0 to 100.
Instruction: {question}
Response: {response}
Consider: accuracy, completeness, clarity, helpfulness, and whether it
fully addresses the instruction. Be critical - most responses should
score between 30 and 80. A score above 90 means near-perfect.
Output ONLY a single integer (0-100).
```

Best-of-N, PairChoice

```
Here is a {task label} and two proposed answers.
Question: {question}
{Answer/Solution} A: {response_a}
{Answer/Solution} B: {response_b}
Which answer is {quality condition}? Output ONLY 'A' or 'B'.
```

For open-ended tasks the final question is "Which response better answers the instruction?".

Best-of-N, PairScore

```
Here is a {task label} and two proposed answers.
Question: {question}
{Answer/Solution} A: {response_a}
{Answer/Solution} B: {response_b}
How confident are you that each answer is correct on a scale of 0 to
100.
Output format: "A: [score], B: [score]"
```

For open-ended tasks the scoring instruction is "Rate the quality of each response on a scale of 0 to 100. Consider accuracy, completeness, clarity, and helpfulness."

The {task label}, {Answer/Solution}, and {quality condition} slots are task-dependent, e.g., "math problem" / "Solution" / "more likely to have the correct final answer" for math, and "programming problem" / "Solution" / "this code is correct and handles all edge cases" for code.

The judge score of a response is its number of wins in pairwise comparisons against the other $N-1$ candidates (0–7 for $N=8$), with presentation order randomized per comparison:

AlpacaEval judge: response scoring

```
Compare two responses to the same instruction. Consider accuracy,
helpfulness, detail, and clarity.
Instruction: {instruction}
Response A: {response_a}
Response B: {response_b}
Which response is better? Output ONLY 'A' or 'B'.
```

For the head-to-head evaluation (Table 13), the judge compares the two selected responses in both orderings; agreement gives a win, disagreement counts as a tie:

AlpacaEval judge: head-to-head

```
Here is an instruction and two responses. Which response better
answers the instruction? Consider accuracy, helpfulness, detail, and
clarity.
Instruction: {instruction}
Response A: {response_a}
Response B: {response_b}
Output ONLY 'A', 'B', or 'Tie'.
```

I.1 DIAGNOSTIC PAIRS (SECTION 2)

PairScore, diagnostic wording

```
How confident are you that each of the following statements is
factually correct? Provide a score for each statement on a scale of 0
to 100.
A: "{claim_a}"
B: "{claim_b}"
Output format: "A: [score], B: [score]"
```

PairChoice

```
Which of the following statements are you more confident of being
correct?
A: "{claim_a}"
B: "{claim_b}"
Output ONLY 'A' or 'B'.
```

Tie option, first wording (lines added to the PairChoice prompt)

```
If you are equally confident in both, answer C.
Output ONLY 'A', 'B' or 'C'.
```

Tie option, second wording (lines added to the PairChoice prompt)

```
If you cannot tell the two apart, answer C.
Output ONLY 'A', 'B' or 'C'.
```

I.2 k-WISE ELICITATION (SECTION 4.3)

k-score

```
How confident are you that each of the following statements is
factually correct? Provide a score for each statement on a scale of 0
to 100.
A: "{claim_a}"
B: "{claim_b}"
...
Output format: "A: [score], B: [score], ..."
```

k-rank

```
Rank the following statements from most to least likely to be correct.
A: "{claim_a}"
B: "{claim_b}"
...
Output ONLY the ranking as comma-separated letters.
```