Pref2Point: Distilling Pairwise Rationales into Interpretable Pointwise Scoring Prompts

Anonymous ACL submission

Abstract

Pointwise metrics are essential for scalable single-item evaluation, but reliable scoring functions are difficult to design. When evaluation criteria are subjective or task-specific, absolute scores depend heavily on the rubric and prompt, while pairwise judgments are typically more consistent for both humans and LLMs. Pairwise evaluation, however, does not directly provide the single-item scores needed for thresholds, ranking metrics, or downstream decision making. We introduce Pref2Point, an unsupervised method which derives interpretable pointwise rubrics from pairwise preference data. It extracts recurring evaluation dimensions from natural-language rationales and produces a single-item scorer that preserves the ranking structure implied by pairwise judgments. On diverse benchmarks Pref2Point improves over manually designed LLM evaluation prompts and is competitive with prompts optimized using gold labels, particularly on subjective, multi-dimensional tasks.[1]

1 Introduction

Pointwise assessment of accuracy, quality, and related attributes is standard in domains such as educational testing and information retrieval. Educational testing commonly relies on scoring rubrics for essays and exams, often with significant consequences for students' academic and professional outcomes (Jonsson and Svingby, 2007; Wiggins, 1998; Hanson, 1992). Information retrieval evaluation similarly depends on pointwise relevance judgments in metrics such as nDCG (Harman, 2011).

More recently, LLM-based pointwise evaluation has become common for tasks such as assessing document coherence and factuality, evaluating extracted entities and relations, and grading free-form model outputs. In many of these settings, no widely accepted evaluation metric exists. As a result, practitioners often construct evaluation procedures by prompting a general-purpose LLM to generate scoring criteria, which are then used for pointwise judgment.

Pointwise scoring systems are difficult to design and apply consistently in subjective evaluation. They ask evaluators to compress multidimensional judgments into fixed numerical scales and often require extensive pretesting for calibration (Collins, 2003). Absolute scores are sensitive to rubric and prompt design for both humans and models (Hanson, 1992; Qin et al., 2023). Humans and LLMs are typically more consistent on *pairwise* comparisons. This pattern is not unique to LLM evaluation. In chess, win–loss records yield stable strength estimates; in information retrieval, assessors make pairwise relevance judgments faster, with higher agreement and closer alignment to user behavior than graded relevance scales, especially when many items cluster at the highest relevance levels (Carterette et al., 2008; Clarke et al., 2021; Chen et al., 2013; Qin et al., 2023; Bennett et al., 2008; Sakai and Zeng, 2020).

Pairwise evaluation alone is impractical in many settings. Approximate transitivity can reduce the number of required comparisons far below $O(n^2)$ (Carterette et al., 2008; Rorvig, 1990), but pairwise judgments do not directly yield single-instance decisions. Educational assessment, for example, requires pass/fail thresholds, grade boundaries, and longitudinal comparability, none of which follows from comparisons such as "essay A is better than essay B." Pairwise judgments may be more reliable, but pointwise scores remain necessary for efficiency, interpretation, and downstream decisions. Existing methods do not reliably convert pairwise preferences into calibrated, interpretable scoring functions without loss of fidelity.

Empirical studies suggest that human preference judgments often have low-dimensional latent struc-

[1] Code available at https://anonymous.4open.science/r/anon-48BD.

ture (Rentschler et al., 1999). When pairwise preference data have such structure, the corresponding comparison axes can be recovered from judgments and converted into interpretable pointwise metrics.

Contributions. We introduce Pref2Point, an unsupervised framework that distills pairwise preference rationales into interpretable pointwise scoring prompts. We propose two variants: Pref2Point-Z, which synthesizes a rubric in a single long-context LLM call, and Pref2Point-O, which iteratively refines the rubric via reflective prompt evolution guided by pairwise disagreement feedback. Experiments across five benchmarks and three LLMs of varying size show consistent improvements over manually designed prompts (Table 2) and competitive performance with supervised optimization using gold labels, which are often unavailable in industry and domain-specific settings, particularly on subjective, multi-dimensional tasks.

2 Related Work

Our work draws on classical paired-comparison scaling, preference learning and learning-to-rank, psychometrics and rubric design, and recent research on LLM-based evaluation. We focus on the lines of work most relevant to low-rank, axis-based representations and position Pref2Point in relation to existing approaches.

Preference-based IR Evaluation and Partial Orders. A long line of IR work advocates preference judgments as a superior alternative to graded relevance, showing higher assessor consistency, better alignment with user behavior, and improved sensitivity at top ranks (Rorvig, 1990; Carterette et al., 2008; Clarke et al., 2021). Clarke et al. (2021) further demonstrate that partial preference orderings can expose distinctions among top-ranked items that are invisible to pointwise nDCG evaluation. This literature also emphasizes two practical issues that shape data collection: (i) preference cycles and other non-transitivities may occur reducing the need to do, and (ii) judging budgets are managed by focusing comparisons on the top items and by allowing ties and weak orderings to reduce effort (Carterette et al., 2008; Clarke et al., 2021; Sakai and Zeng, 2020).

Paired-comparison and ranking models. Paired-comparison and ranking models are constrained forms of latent-trait measurement for comparative judgments. Bradley–Terry and Thurstone models assume a single latent utility per item, while Plackett–Luce extends the same idea to full rankings (Bradley and Terry, 1952; Thurstone, 1927; Plackett, 1975). In IR, several works convert preference judgments into scalar relevance estimates, including Elo-style adaptations (Sakai and Zeng, 2020).

Preference learning and alignment. Preference learning has long used pairwise or listwise supervision to learn order-preserving rankers (Burges et al., 2005; Burges, 2010; Xia et al., 2008), typically with opaque scoring functions. In LLM alignment, RLHF trains a reward model from pairwise human comparisons and optimizes the policy under a KL constraint, often with PPO (Ouyang et al., 2022; Schulman et al., 2017). Recent work argues that RL-based and supervised preference optimization differ mainly in data source and feedback mechanism, whether reward models, direct preference labels, or LLM judges (Gao et al., 2024). This links RLHF to offline objectives such as DPO, which optimizes directly from preference pairs without an RL loop (Rafailov et al., 2023), and to AI-feedback pipelines such as RLAIF and UltraFeedback, which scale preference data but may introduce judge bias (Lee et al., 2024; Cui et al., 2023).

LLMs as evaluators and explanation-based supervision. Recent work studies LLMs as automatic judges for open-ended text quality using pairwise comparisons and pointwise scoring (Zheng et al., 2023; Kocmi and Federmann, 2023). Although effective, these methods typically treat pairwise outcomes as black-box labels or directly elicit scalar scores without converting explanations into reusable rubrics. Relatedly, OnlineRubrics (Rezaei et al., 2025) shows that rubric criteria can be elicited online from pairwise comparisons and expanded during RL, while Rubicon (Huang et al., 2025) uses large, fixed banks of rubric dimensions to anchor RL in open domains. Together, these results suggest that pairwise judgments are valuable not only as supervision signals, but also as a mechanism for discovering and maintaining explicit evaluation dimensions for robust pointwise grading and rubric-based reward design.

Prompt optimization and reflective search. Frameworks for prompt search and optimization, such as DSPy and GEPA, automate the discovery and refinement of prompt programs using task-level objectives and feedback (Khattab et al., 2023; Agrawal et al., 2025). GEPA in particular treats prompt improvement as reflective evolution, where candidate prompts are mutated using textual feedback and selected through sampling to preserve

diversity and avoid early local optima from greedy best-candidate selection (Agrawal et al., 2025). This procedure yields compact, high-performing instructions that are often substantially shorter than those produced by other prompt optimizers.

Interpretability, sample efficiency, and emergent structure. Methods that extract interpretable concepts, relations, or rules from learned models can support auditing, intervention, and alignment. Concept bottleneck models make this structure explicit by routing predictions through human-interpretable variables, enabling concept-level debugging and revision (Koh et al., 2020; Marconato et al., 2022). Pref2Point instead induces shared comparison axes through explanation-aligned prompt optimization, with fewer annotations and calibrated pointwise outputs that can be audited.

Positioning and contributions. Pref2Point differs from prior work by converting pairwise preferences and rationales into an interpretable pointwise rubric. Classical paired-comparison models recover scalar utilities but leave comparison dimensions uninterpreted, while learning-to-rank and reward-modeling methods typically optimize opaque scorers. Existing LLM-as-judge approaches either elicit pointwise scores directly or use pairwise explanations without turning them into reusable metrics. In contrast, Pref2Point uses pairwise rationales as semantic supervision to discover, merge, and weight recurring evaluation axes, producing a compact single-item scoring prompt with auditable justifications.

3 Pref2Point

Pref2Point induces interpretable pointwise scoring prompts from pairwise preference data. Structured pairwise reasoning exposes task-specific evaluation axes that can be distilled into a pointwise rubric. We consider two methods. A zero-shot method referred to as Pref2Point-Z (Section 3.3) uses long-context LLMs to synthesize rubrics directly from pairwise examples. The optimization-based method, Pref2Point-O, (Section 3.4) uses GEPA to refine prompts for pairwise agreement, using pairwise reasoning as textual feedback. Both methods aim to preserve the ranking structure of the pairwise data while scoring single items.

3.1 Problem Formulation

Let $X = \{x_1, \ldots, x_n\}$ be a set of items to evaluate. We seek a pointwise metric

$$s : X \to \mathcal{R}, \qquad \mathcal{R} = \{1, \ldots, K\}$$

where $\mathcal{R}$ is a discrete, finite *rating space*. The function s assigns each item an integer score on a fixed K-point scale.

We are given a set of pairwise preferences

$$\mathcal{P} = \{(x_i, x_j, y_{ij})\}, \qquad y_{ij} = 1 \text{ if } x_i \succ x_j,$$

and our objective is to learn s such that

$$x_i \succ x_j \Rightarrow s(x_i) > s(x_j)$$

for as many pairs as possible. Because $\mathcal{R}$ is finite, some preferred pairs will inevitably receive the same score $s(x_i) = s(x_j)$. We treat these cases explicitly as *ties* and penalize them softly in our objective.

Formally, we minimize the *pairwise disagreement rate*

$$\mathcal{L}_{\text{rank}}(s) = \frac{1}{|\mathcal{P}|} \sum_{(i,j) \in \mathcal{P}} \mathbf{1}\big[s(x_i) \le s(x_j) \wedge y_{ij} = 1\big], \tag{1}$$

This objective measures how well the induced pointwise scores preserve the observed pairwise preferences. **This is the core optimization objective we use for prompt optimization**: we evolve candidate pointwise prompts to minimize disagreement with collected pairwise labels, ensuring that single-item scores maintain global ranking fidelity.

3.2 Structured Pairwise Judgment Collection

Before describing our pointwise prompt generation methods, we describe the pairwise data collection protocol used to elicit comparison reasoning for distillation into pointwise rubrics.

Large language models perform pairwise comparisons reliably, particularly state-of-the-art (SOTA) models. Unlike pointwise scoring, which requires absolute calibration and is sensitive to prompt design, pairwise judgments rely on direct comparison and remain more stable across prompt formulations. We therefore use SOTA models for pairwise annotation even when the resulting pointwise prompt is intended for smaller models. This produces more reliable preference labels and higher-quality reasoning for pointwise prompt design.

Our pairwise prompt enforces a three-step reasoning structure rather than allowing unconstrained comparison:

Step 1: Identify pair-specific comparison dimensions. Given a pair of objects, the model identifies the relevant comparison dimensions. For example, when comparing two web documents for a news-seeking query, relevant dimensions may include recency, source authority, comprehensiveness, factual accuracy, and geographic specificity. The model generates a task-specific set of comparison dimensions.

Step 2: Structured comparative reasoning. The model explains why one option is preferred over the other, or declares a tie when appropriate. The reasoning must satisfy two conditions: (A) it explicitly addresses each dimension identified in Step 1, and (B) it specifies how those dimensions are weighted in the final decision. This reasoning exposes both the relevant dimensions and their relative importance in the judgment process.

Step 3: Final decision. The model outputs its preference decision or declares a tie.

We apply this protocol to all pairs in the evaluation set, producing an augmented dataset

$$\mathcal{D} = \{(x_i, x_j, y_{ij}, E_{ij})\},$$

where $(x_i, x_j, y_{ij}) \in \mathcal{P}$ and E_{ij} is the structured reasoning generated by the pairwise model, including both the identified comparison dimensions and the weighting underlying the final preference decision. This dataset forms the basis of the two pointwise prompt generation methods described in the following subsections.

3.3 Zero-Shot Pointwise Prompt Generation

Our first method, (Pref2Point-Z), uses the long-context capacity of SOTA LLMs to synthesize a pointwise prompt directly from collected pairwise reasoning, without iterative optimization.

A pointwise prompt can act as a rubric. It specifies the dimensions the model should assess, then combines those assessments into a final decision using appropriate weights. In pointwise document evaluation, for example, the model may check timeliness, response comprehensiveness, factual correctness, location specificity, and other task-relevant factors. The final score should reflect the relative importance of these intermediate judgments.

The pairwise reasoning dataset $\mathcal{D}$ provides the information needed to derive such a rubric. Each pairwise explanation E_{ij} contains: (1) the comparison dimensions used for that pair, and (2) the way those dimensions contributed to the preference decision. Aggregating across many pairwise explanations lets us recover recurring dimensions and estimate how they should affect pointwise evaluation.

We prompt a long-context LLM with all pairwise reasoning examples from $\mathcal{D}$. The meta-prompt instructs the model to:

1. analyze the provided pairwise reasoning examples;
2. identify recurring comparison dimensions;
3. extract patterns in how these dimensions are weighted and combined; and
4. synthesize a pointwise evaluation prompt that produces single-item scores whose induced pairwise preferences align with the pairwise measurements.

The resulting pointwise prompt P^* evaluates individual items along the recovered dimensions and aggregates those evaluations so that, when applied to pairs of items, the induced scores preserve the preference structure observed in $\mathcal{D}$.

Pref2Point-Z requires no optimization: a single LLM call produces a deployment-ready prompt. It uses all available pairwise examples up to the context limit and, as shown in Section 5, often matches the optimization-based variant at lower cost. See Appendix for a formal description of all procedures.

3.4 Optimization-Based Pointwise Prompt Generation

While Pref2Point-Z is fast and effective, an optimization-based approach can further refine the pointwise prompt through iterative improvement guided by the pairwise disagreement objective defined in Section 3.1. We refer to this method as Pref2Point-O.

Prompt optimization framework. We employ GEPA (Generative Evolution of Prompts with Adaptive Refinement) (Agrawal et al., 2025), which maintains a population of candidate prompts and improves them through mutation and selection. GEPA fits this task because it can use scalar performance metrics together with textual feedback during optimization.

Candidate prompt evaluation. Each candidate pointwise prompt P defines a scoring function $\hat{s}_P : X \to \mathcal{R}$ over individual items. To evaluate a candidate, we apply it to both items in each pair $(x_i, x_j) \in \mathcal{P}$, producing scores $\hat{s}_P(x_i)$ and $\hat{s}_P(x_j)$. We measure performance using the pairwise disagreement rate in Equation 1, which tests whether the pointwise scores preserve the pairwise preferences.

Incorporating pairwise reasoning as feedback. We use the pairwise reasoning $\{E_{ij}\}$ as textual feedback for prompt evolution. Instead of adding these explanations as a separate loss term, we pass them to GEPA's **reflection model**.

When a candidate prompt P is evaluated on a minibatch $\mathcal{B}$, we construct feedback containing:

1. Scalar performance: pairwise agreement rate on $\mathcal{B}$
2. Failure cases: pairs where $\hat{s}_P(x_i) \leq \hat{s}_P(x_j)$ but $y_{ij} = 1$
3. The corresponding pairwise reasoning E_{ij} for failed pairs
4. Recurring dimensions in the pairwise reasoning for $\mathcal{B}$

This feedback gives GEPA's reflection model both the failed comparisons and the reasoning associated with them, enabling mutations such as adding missing dimensions, adjusting weights, refining the aggregation mechanism, or merging redundant criteria.

Emergent axis consolidation. Over successive iterations, candidate prompts tend to retain a smaller, more stable set of evaluation dimensions. Prompts with consistent dimension names and weighting schemes obtain higher pairwise agreement and are more likely to survive selection. After exposure to many pairwise reasoning examples, the reflection model can identify near-synonymous dimensions, such as "grammar" and "linguistic quality," and merge them into canonical forms without explicit clustering or manual intervention.

Advantages over zero-shot. Iterative refinement directly optimizes the pairwise agreement objective, can recover from weak initial prompts through evolutionary search, and aggregates signal across rounds when rationales are noisy or inconsistent. The cost is additional computation: each round requires LLM calls to score items and generate feedback. For high-stakes tasks, this cost is justified empirically in Section 5. Algorithm 2 gives the formal procedure.

4 Experimental Setup

We evaluate whether explanations collected with pairwise preferences reveal a compact set of comparison axes that a refinement LLM can distill into an interpretable pointwise prompt, measured by its performance on held-out pointwise test sets.

Dataset	Domain	Scale	$\lvert K \rvert$	Train	Test
DREsS	EFL essays	3–15	13	1000	1000
LLMJudge	Doc. relevance	0–3	4	1000	1000
HelpSteer2	LLM responses	0–12	13	1000	1000
SummEval	News summaries	4–20	17	485	800
WMT DA	MT quality	0–100	101	983	990

Table 1: Dataset summary. $|K|$: distinct score levels. Train/Test: pointwise instances. 300 pairwise instances are randomly sampled for training the two Pref2Point methods.

Datasets. We evaluate on five benchmarks spanning a wide range (4 to 101 distinct values) in scale granularity, three annotation paradigms, and both single-aspect and multi-dimensional composite tasks (Table 1). In all experiments we use only the total composite score and disregard individual rubric dimensions. The datasets are:

- **DREsS** (Yoo et al., 2025): EFL essays scored by trained raters on content, organization, and language (each 1–5; composite 3–15). We train on DREsS_Std and test on DREsS_New.[2]
- **LLMJudge** (Rahmani et al., 2024): query–document pairs from TREC 2024, annotated for relevance (0–3); test split is class-balanced (250 per level).
- **HelpSteer2** (Wang et al., 2024): human ratings of LLM responses; we use the composite of helpfulness, correctness, and coherence (each 0–4), excluding complexity and verbosity. Consecutive rows share a prompt, providing natural pairs.
- **SummEval** (Fabbri et al., 2021): machine summaries of news articles rated by three annotators on four dimensions (1–5 each; composite is the rounded sum of dimension means, yielding integers in [4, 20]). We split at the article level to prevent leakage.
- **WMT DA** (Kocmi et al., 2022): direct assessment of MT quality (0–100); pairs are constructed within the same source sentence. This tests our methods at the extreme of rating granularity.

LLMs. We use three models: Claude Sonnet 4.5 (also used for prompt refinement), and two

[2] 1,813 essays after filtering rows with empty essay text.

open-weight models, gpt-oss-120b and gpt-oss-20b, spanning a range of model sizes to test cross-model robustness.

4.1 Data collection protocol

For each dataset we collect a set of annotated pairs $\mathcal{D} = \{(x_i, x_j, y_{ij}, E_{ij})\}$. Each pair shares the same task context (same essay prompt or same retrieval query). An annotator (human or LLM acting as a pairwise judge) is asked to:

1. indicate a binary preference y_{ij} (A preferred / B preferred), and
2. provide a short natural-language explanation E_{ij} that lists the 2–6 most salient comparison axes and a one-sentence justification per axis.

Annotations intentionally do not include numeric axis scores or a pre-defined axis vocabulary. Explanations therefore capture the raw, possibly noisy language people use to justify preferences; the refinement LLM is then tasked with synthesizing this raw evidence into a stable pointwise rubric.

4.2 Baselines and experimental conditions

We compare the two Pref2Point methods with two baselines that reflect real-life settings:

- **Manual baseline:** The user asks an LLM to create a prompt for the task which is often the case in company specific or personalized tasks.
- **Pointwise Supervised:** Prompt is optimized with access to gold labels using GEPA which is assumed to achieve the upper-bound performance.

4.3 Metrics and evaluation protocol

Given that the tasks in this work are ordinal rating tasks, we use Quadratic Weighted Kappa (QWK) (Cohen, 1968) to measure performance. Additionally, to have an interpretable direct numeric insight into performance numbers, we also report Mean Squared Error (MSE) in all experiments. All reported numbers are the mean and standard deviation in six runs and are measured in holdout test sets.

For reproducibility, full prompt templates and hyperparameter settings is available[3].

4.4 Ablation studies

We experiment with different LLM budgets (light, medium, heavy), including or exclusion of the rationales in Pref2Point-Z, the choice of the initial prompt and stratified pair selection methodology. We found that single-line prompts versus the manual prompts do not change the performance of Pref2Point-Z and Pref2Point-O. For pair selection, we first annotated the training data set with manual pointwise prompts and performed stratified pair selection based on score difference buckets. We did not observe a statistically significant change in performance. The ablation studies on the budget and the impact of rationales are provided in the result section.

[3] https://anonymous.4open.science/r/anon-48BD/

5 Results

Table 2 reports MSE and QWK across five datasets, three models, and six seeds. Across all settings, a consistent pattern emerges: Pref2Point-Z is the best choice for well-defined or single-axis tasks (LLMJudge, WMT), where a single synthesis pass reliably recovers the relevant criteria. Pref2Point-O is preferable for multi-dimensional tasks with interacting criteria (SummEval, HelpSteer2, DREsS), where iterative refinement corrects misweighted axes that a single pass may miss.

Dataset trends. On LLMJudge, all methods fall within a narrow range, consistent with document relevance being a familiar evaluation task for LLMs. Pref2Point-Z nevertheless obtains the best MSE across all three models, suggesting that pairwise rationales provide calibration signal beyond generic pointwise prompting.

On DREsS, both Pref2Point variants consistently improve over Manual across all models. On HelpSteer2, the gains are concentrated on the gpt-oss models—Pref2Point-Z cuts MSE nearly in half for gpt-oss-120b and gpt-oss-20b—while on Claude-4.5 Pref2Point-Z degrades (MSE 21.08 vs. 14.14). This failure occurs because the single-pass synthesis misweights interacting criteria when the scorer is already well-calibrated; Pref2Point-O recovers to match Manual in this case.

SummEval is the main setting where Pref2Point-Z is unstable (std up to 50 on gpt-oss-20b), indicating that a single synthesis pass does not reliably recover the relative weights among four quality dimensions. Pref2Point-O performs markedly better: on gpt-oss-120b and gpt-oss-20b it substantially outperforms Manual and closes much of the gap with the supervised oracle.

On WMT, the 100-point scale requires fine-grained score placement, not only ordinal separation. Manual is the strongest unsupervised method on two of three models (significantly so). This repre-

		Supervised		Unsupervised (no gold labels)					
		Oracle		Manual		Pref2Point-Z		Pref2Point-O	
Dataset	Model	MSE↓	QWK↑	MSE↓	QWK↑	MSE↓	QWK↑	MSE↓	QWK↑
LLMJudge	Claude-4.5	1.11±0.07	0.49±0.03	1.03±0.05	0.47±0.02	**0.74**±0.04*	**0.51**±0.02*	1.04±0.09	0.46±0.03
	gpt-oss-120b	1.19±0.11	**0.45**±0.02	1.03±0.07	0.45±0.02	**0.92**±0.22*	0.44±0.10	1.06±0.07	0.44±0.03
	gpt-oss-20b	1.11±0.13	0.47±0.03	1.07±0.09	0.47±0.03	**0.95**±0.20*	0.45±0.05	1.00±0.08*	**0.48**±0.02*
DREsS	Claude-4.5	**6.83**±0.52	**0.24**±0.04	11.60±3.00	0.19±0.06	8.69±0.99*	0.18±0.03	8.50±1.77*	0.23±0.05*
	gpt-oss-120b	**8.02**±3.12	0.20±0.04	18.94±5.26	0.12±0.05	10.85±4.03*	**0.20**±0.07*	15.36±5.21*	0.15±0.08*
	gpt-oss-20b	**11.70**±3.97	**0.13**±0.05	18.13±7.08	0.10±0.05	14.86±5.00*	0.12±0.07*	15.14±5.94*	0.11±0.06*
HelpSteer2	Claude-4.5	**13.62**±3.45	**0.42**±0.07	14.14±5.11	0.42±0.09	21.08±2.11	0.30±0.02	14.07±4.76	0.42±0.08
	gpt-oss-120b	11.85±3.94	0.44±0.07	11.87±4.29	0.45±0.06	**8.77**±2.43*	**0.48**±0.04*	10.69±3.32*	0.47±0.05*
	gpt-oss-20b	15.97±4.75	0.35±0.06	16.92±5.19	0.34±0.06	**10.90**±2.06*	**0.42**±0.05*	15.32±5.06*	0.36±0.06
SummEval	Claude-4.5	**11.78**±5.81	**0.53**±0.10	15.48±3.04*	0.43±0.08	35.93±6.49	0.26±0.05	17.58±7.57	0.43±0.12
	gpt-oss-120b	**5.35**±1.03	**0.61**±0.02	26.58±21.40	0.35±0.20	27.78±16.92	0.32±0.15	7.78±4.62*	0.56±0.12*
	gpt-oss-20b	**6.67**±1.39	**0.52**±0.05	15.11±4.84	0.36±0.08	41.76±50.05	0.27±0.16	8.12±2.37*	0.48±0.07*
WMT DA	Claude-4.5	846.88±105.12	0.46±0.04	**810.28**±62.41*	**0.47**±0.03*	979.59±89.62	0.44±0.04	864.15±81.74	0.45±0.02
	gpt-oss-120b	1012.05±152.14	0.43±0.05	**851.41**±72.48*	**0.47**±0.04*	972.59±141.77	0.44±0.03	887.60±74.61	0.45±0.03
	gpt-oss-20b	994.32±88.80	0.42±0.04	929.82±56.71	**0.44**±0.03	**917.40**±107.47	0.43±0.03	944.95±73.83	0.44±0.05

Table 2: Performance comparison (mean ± std) across six random seeds for light budget. ↓ lower is better, ↑ higher is better. **Bold** marks the best result per row for each metric (including Supervised). Underline marks the best unsupervised result. Stars (*): Unsupervised methods are starred when significantly better than manual and vice versa. Significance uses an instance-level paired bootstrap (1k resamples); $^*p < 0.05$.

	QWK ↑			MSE ↓		
Dataset	+Rat.	−Rat.	Δ	+Rat.	−Rat.	Δ
LLMJudge	.464	**.486**	+.022	1.04	**0.96**	−0.1
DREsS	**.230***	.197	−.033	**8.5***	10.9	+2.4
HelpSteer2	**.418***	.389	−.029	**14.1***	16.2	+2.1
SummEval	**.432***	.406	−.026	**17.6***	18.5	+0.9
WMT	.454	**.462**	+.008	864.1	**840.7**	−23.5

Table 3: Ablation: removing pairwise rationales from Pref2Point-O feedback (Claude-4.5, light budget, 6 seeds) results in regression on most datasets.

Aspect	**Manual**	**Ours**
Weighting	Symmetric	Asymmetric
Content share	Implicit	33% (highest)
Grammar rule	Equal weight	If impedes meaning
Anchors	Abstract	Per-level behavioral
Output	Free-form	Per-criterion points

Table 4: Manual vs. Pref2Point-O prompts on DREsS.

sents the main limitation of our approach: pairwise rationales provide ordinal signal but not the absolute calibration that very fine-grained scales demand.

Model-size trends. Performance generally degrades from the larger models to gpt-oss-20b, though gpt-oss-120b occasionally surpasses Claude-4.5 (e.g. on HelpSteer2 and SummEval). Across this variation, Pref2Point typically retains its advantage over Manual, suggesting that the induced prompts encode task-level evaluation criteria rather than scorer-specific artifacts.

Ablation: role of rationales and optimization budget. Table 3 isolates the contribution of pairwise rationales in Pref2Point-O. Removing rationales significantly degrades both QWK and MSE on DREsS (−0.033 QWK, +2.4 MSE), HelpSteer2 (−0.029 QWK, +2.1 MSE), and SummEval (−0.026 QWK, +0.9 MSE)—all multi-dimensional tasks where the relevant quality axes and their weighting are not given explicitly (p<0.05, paired bootstrap). On LLMJudge and WMT, removing rationales has no significant effect or slightly improves performance, suggesting that binary preference feedback is sufficient when the evaluation task is well-specified (single-axis relevance) or when the scale is too fine-grained (0–100) for qualitative explanations to provide actionable guidance. We also ablated the GEPA optimization budget and found no meaningful differences across light, medium, and heavy settings; we therefore use the light budget in all experiments.

Qualitative analysis. Table 4 compares the Manual and Pref2Point prompts on DREsS. Both use an additive scoring structure, but the Manual distributes points symmetrically across four IELTS criteria (each described at equal depth with no relative priority), while Pref2Point learns an

Essay Assessment Rubric (0–15 Scale)
Score the essay on a 0–15 scale. For each criterion, pick the descriptor that best matches and award the corresponding points; then sum.
Task Response — 0 to 5 points
5: Fully addresses all parts of the prompt; clear, relevant position; well-supported, extended examples. **4**: Mostly clear position with relevant development (minor gaps). **3**: Partly addresses the task; limited or uneven development. **2**: Fails to address key parts; underdeveloped or off-topic. **1**: Barely attempts; mostly irrelevant or incoherent. **0**: No relevant response.
Coherence & Cohesion — 0 to 4 points
4: Logical organization, clear paragraphing, smooth progression, effective cohesive devices. **3**: Generally logical; some lapses in flow or linking. **2**: Weak organization; ideas jump; linking repetitive or mechanical. **1**: Poor organization; little paragraphing; hard to follow. **0**: No organization / unreadable.
Lexical Resource — 0 to 3 points
3: Wide range, precise word choice, few errors. **2**: Sufficient to express meaning; some inappropriate choices but meaning clear. **1**: Limited; frequent inappropriate choices that limit clarity. **0**: No useful vocabulary.
Grammatical Range & Accuracy — 0 to 3 points
3: Wide range of structures; errors rare, do not impede understanding. **2**: Mix of structures with some errors; meaning clear despite mistakes. **1**: Limited range; frequent errors sometimes obscure meaning. **0**: Grammar so poor meaning is largely incomprehensible.
Key design choices
▷ Asymmetric weighting: Task Response (33%) > Coherence (27%) > Lexical = Grammar (20% each).
▷ Additive structure forces per-criterion justification, preventing holistic impression bias.
▷ Grammar penalized only when errors "obscure meaning"—surface errors tolerated.
Output: per-criterion points + 1-sentence justification each, then summed integer (0–15).

Table 5: Pref2Point-O's induced prompt for DREsS (seed 1). The optimization discovers an additive point system with asymmetric weighting favoring content over mechanics, and concrete behavioral anchors at each level. For brevity, only a summary is shown; full prompt released with the code.

asymmetric allocation from pairwise preferences: Task Response receives the most points (5/15), Grammar the fewest (3/15). Crucially, the induced prompt provides concrete behavioral anchors at each score level and tolerates grammar errors unless they "obscure meaning"—matching the trade-offs observed in pairwise judgments, where essays with stronger arguments outrank cleaner but shallower essays (see Table 5).

6 Discussion

Across datasets, pairwise rationales are most useful for new, subjective, or domain-specific scoring tasks where gold pointwise labels are unavailable and the scoring criteria are not already fixed. Pref2Point-Z is a low-cost option because it produces a rubric in one synthesis step and works well on familiar or moderately structured tasks. Pref2Point-O is more useful when criteria interact, rationales are noisy, or the initial synthesized rubric needs correction from preference failures.

The main limitation is calibration. Pairwise evidence identifies which dimensions matter and how they trade off, but it does not determine precise locations on a fine-grained numeric scale. This is most visible on WMT, where the 100-point scale makes absolute score placement more important than rank preservation. Anchors, score-distribution matching, or a small number of calibrated pointwise examples may be needed in such settings.

The ablation results also indicate when rationales matter. Removing them hurts DREsS and HelpSteer2 more than LLMJudge, where the evaluation concept is already clear. Pref2Point is therefore best viewed as a method for inducing pointwise rubrics when labels are scarce, rather than a substitute for supervised pointwise optimization when reliable gold labels are available.

7 Conclusion

We presented Pref2Point, which combines paired-comparison scaling with reflective prompt evolution (GEPA) to construct interpretable pointwise rating prompts from pairwise judgments. Experiments on five datasets with three LLMs show that refined prompts align well with preference data and produce attribute-based rubrics competitive with pointwise prompts optimized with gold labels.

The method is especially valuable in industry settings, where new rating tasks often lack gold labels and prompt quality cannot be measured directly. In these cases, Pref2Point can use pairwise preferences to induce a pointwise rubric, including for industry-specific or personalized tasks where the LLM lacks relevant search or parametric knowledge.

Limitations

Dataset Suitability Our proposed methods are most useful for tasks for which the parametric knowledge of the LLMs does not cover the information required to derive a prompt or to annotate instances with little effort. Such tasks are abundant in industry settings or for specialized tasks; however, we were not able to experiment with such tasks since they are either a) proprietary or b) there are no labeled data for evaluation.

LLM judge. Our pairwise data is LLM-generated, so judge biases can propagate into the induced rubric. We did not validate against human preferences or audit for lexical biases such as favoring longer outputs.

Scale and calibration. The objective learns pairwise ordering, not calibrated scores. We also assume the dataset rating range K rather than choosing K or thresholds. In addition, our method does not take prior distribution of ratings into account and assumes a uniform score distribution dependent on pairwise samples.

Omitted baselines. We did not test agentic/ReAct evaluators, training-free relative-reward methods, or classical pairwise-to-pointwise models such as Bradley–Terry, Thurstone, and Plackett–Luce Bradley and Terry, 1952; Thurstone, 1927; Plackett, 1975. The classical methods would require utility estimation, content-to-utility regression, and thresholding into $\{1,...,K\}$, and do not use rationales.

References

Lakshya A. Agrawal, Shangyin Tan, Dilara Soylu, Noah Ziems, Rishi Khare, Krista Opsahl-Ong, Arnav Singhvi, Herumb Shandilya, Michael J. Ryan, Meng Jiang, Christopher Potts, Koushik Sen, Alexandros G. Dimakis, Ion Stoica, Daniel Klein, Matei Zaharia, and Omar Khattab. 2025. GEPA: reflective prompt evolution can outperform reinforcement learning. *CoRR*, abs/2507.19457.

Paul N. Bennett, Ben Carterette, Olivier Chapelle, and Thorsten Joachims. 2008. Beyond binary relevance: preferences, diversity, and set-level judgments. *SIGIR Forum*, 42(2):53–58.

Ralph Allan Bradley and Milton E. Terry. 1952. Rank analysis of incomplete block designs. *Biometrika*.

Christopher J. C. Burges. 2010. From ranknet to lambdarank to lambdamart: An overview. Technical Report MSR-TR-2010-82, Microsoft Research.

Christopher J. C. Burges, Tal Shaked, Erin Renshaw, Ari Lazier, Matt Deeds, Nicole Hamilton, and Greg Hullender. 2005. Learning to rank using gradient descent. In *Proceedings of the 22nd International Conference on Machine Learning*.

Ben Carterette, Paul N Bennett, David Maxwell Chickering, and Susan T Dumais. 2008. Here or there: Preference judgments for relevance. In *European Conference on Information Retrieval*, pages 16–27. Springer.

Xi Chen, Paul N Bennett, Kevyn Collins-Thompson, and Eric Horvitz. 2013. Pairwise ranking aggregation in a crowdsourced setting. In *Proceedings of WSDM*, pages 193–202.

Charles L. A. Clarke, Alexandra Vtyurina, and Mark D. Smucker. 2021. Assessing top- preferences. *ACM Trans. Inf. Syst.*, 39(3):33:1–33:21.

Jacob Cohen. 1968. Weighted kappa: Nominal scale agreement provision for scaled disagreement or partial credit. *Psychological Bulletin*, 70:213–220.

Debbie Collins. 2003. Pretesting survey instruments: An overview of cognitive methods. *Quality of Life Research*, 12:229–238.

Ganqu Cui, Lifan Yuan, Ning Ding, Guanming Yao, Wei Zhu, Yuan Ni, Guotong Xie, Zhiyuan Liu, and Maosong Sun. 2023. Ultrafeedback: Boosting language models with high-quality feedback. *CoRR*, abs/2310.01377.

Alexander R Fabbri, Wojciech Kryściński, Bryan McCann, Caiming Xiong, Richard Socher, and Dragomir Radev. 2021. Summeval: Re-evaluating summarization evaluation. *Transactions of the Association for Computational Linguistics*, 9:391–409.

Bofei Gao, Feifan Song, Yibo Miao, Zefan Cai, Zhe Yang, Liang Chen, Helan Hu, Runxin Xu, Qingxiu Dong, Ce Zheng, Wen Xiao, Ge Zhang, Daoguang Zan, Keming Lu, Bowen Yu, Dayiheng Liu, Zeyu Cui, Jian Yang, Lei Sha, and 5 others. 2024. Towards a unified view of preference learning for large language models: A survey. *CoRR*, abs/2409.02795.

F. Allan Hanson. 1992. *Testing Testing: Social Consequences of the Examined Life*. University of California Press.

Donna Harman. 2011. *Information retrieval evaluation*. Morgan & Claypool Publishers.

Zenan Huang, Yihong Zhuang, Guoshan Lu, Zeyu Qin, Haokai Xu, Tianyu Zhao, Ru Peng, Jiaqi Hu, Zhanming Shen, Xiaomeng Hu, Xijun Gu, Peiyi Tu, Jiaxin Liu, Wenyu Chen, Yuzhuo Fu, Zhiting Fan, Yanmei Gu, Yuanyuan Wang, Zhengkai Yang, and 2 others. 2025. Reinforcement learning with rubric anchors. *CoRR*, abs/2508.12790.

Anders Jonsson and Gunilla Svingby. 2007. The use of scoring rubrics: Reliability, validity, and educational consequences. *Educational Research Review*, 2:130–144.

O. Khattab, Arnav Singhvi, Paridhi Maheshwari, Zhiyuan Zhang, Keshav Santhanam, Sri Vardhamanan, Saiful Haq, Ashutosh Sharma, Thomas T. Joshi, Hanna Moazam, Heather Miller, Matei Zaharia, and Christopher Potts. 2023. Dspy: Compiling declarative language model calls into self-improving pipelines. *ArXiv*, abs/2310.03714.

Tom Kocmi, Rachel Bawden, Ondřej Bojar, Anton Dvorkovich, Christian Federmann, Mark Fishel, Thamme Gowda, Yvette Graham, Roman Grundkiewicz, Barry Haddow, and 1 others. 2022. Findings of the 2022 conference on machine translation (wmt22). In *Proceedings of the Seventh Conference on Machine Translation (WMT)*, pages 1–45.

Tom Kocmi and Christian Federmann. 2023. Large language models are state-of-the-art evaluators of translation quality. In *Proceedings of the 24th Annual Conference of the European Association for Machine Translation, EAMT 2023, Tampere, Finland, 12-15 June 2023*, pages 193–203. European Association for Machine Translation.

Pang Wei Koh, Thao Nguyen, Yew Siang Tang, Stephen Mussmann, Emma Pierson, Been Kim, and Percy Liang. 2020. Concept bottleneck models. In *International conference on machine learning*, pages 5338–5348. PMLR.

Harrison Lee, Samrat Phatale, Hassan Mansoor, Thomas Mesnard, Johan Ferret, Kellie Lu, Colton Bishop, Ethan Hall, Victor Carbune, Abhinav Rastogi, and Sushant Prakash. 2024. RLAIF vs. RLHF: scaling reinforcement learning from human feedback with AI feedback. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net.

Emanuele Marconato, Andrea Passerini, and Stefano Teso. 2022. Glancenets: Interpretabile, leak-proof concept-based models. *CoRR*, abs/2205.15612.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F. Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

Robin L Plackett. 1975. The analysis of permutations. *Journal of the Royal Statistical Society Series C: Applied Statistics*, 24(2):193–202.

Zhen Qin, Rolf Jagerman, Kai Hui, Honglei Zhuang, Junru Wu, Le Yan, Jiaming Shen, Tianqi Liu, Jialu Liu, Donald Metzler, and 1 others. 2023. Large language models are effective text rankers with pairwise ranking prompting. *arXiv preprint arXiv:2306.17563*.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D. Manning, Stefano Ermon, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

Hossein A. Rahmani, Emine Yilmaz, Nick Craswell, Bhaskar Mitra, Paul Thomas, Charles L. A. Clarke, Mohammad Aliannejadi, Clemencia Siro, and Guglielmo Faggioli. 2024. Llmjudge: Llms for relevance judgments. In *Proceedings of The First Workshop on Large Language Models for Evaluationin Information Retrieval (LLM4Eval 2024)*, volume 3752, pages 1–3.

Ingo Rentschler, Martin Jüttner, Alexander Unzicker, and Theodor Landis. 1999. Innate and learned components of human visual preference. *Current Biology*, 9:665–671.

MohammadHossein Rezaei, Robert Vacareanu, Zihao Wang, Clinton Wang, Bing Liu, Yunzhong He, and Afra Feyza Akyürek. 2025. Online rubrics elicitation from pairwise comparisons. *CoRR*, abs/2510.07284.

Mark E. Rorvig. 1990. The simple scalability of documents. *J. Am. Soc. Inf. Sci.*, 41(8):590–598.

Tetsuya Sakai and Zhaohao Zeng. 2020. Good evaluation measures based on document preferences. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval, SIGIR 2020, Virtual Event, China, July 25-30, 2020*, pages 359–368. ACM.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *ArXiv*, abs/1707.06347.

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

Zhilin Wang, Yi Dong, Olivier Delalleau, Jiaqi Zeng, Gerald Shen, Daniel Egert, Jimmy J Zhang, Makesh N Sreedhar, and Oleksii Kuchaiev. 2024. Helpsteer 2: Open-source dataset for training top-performing reward models. *Advances in Neural Information Processing Systems*, 37:1474–1501.

Grant Wiggins. 1998. *Educative Assessment: Designing Assessments to Inform and Improve Student Performance*. Jossey-Bass Publishers.

Fen Xia, Tie-Yan Liu, Jue Wang, Wensheng Zhang, and Hang Li. 2008. Listwise approach to learning to rank: theory and algorithm. In *International Conference on Machine Learning*.

Haneul Yoo, Jieun Han, So-Yeon Ahn, and Alice Oh. 2025. DREsS: Dataset for rubric-based essay scoring on EFL writing. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13439–13454,

Vienna, Austria. Association for Computational Linguistics.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

Algorithm 1 Pref2Point-Z: Zero-Shot Pointwise Prompt Generation

Input: Pairwise dataset $\mathcal{D} = \{(x_i, x_j, y_{ij}, E_{ij})\}$, meta-prompt template Π, SOTA LLM $\mathcal{M}$
Output: Pointwise prompt P^*
Collect pairwise reasoning: $\mathcal{E} = \{E_{ij} : (x_i, x_j, y_{ij}, E_{ij}) \in \mathcal{D}\}$
Construct meta-prompt: $\pi \leftarrow \Pi(\mathcal{E})$ ▷ Include all reasoning examples
Generate pointwise prompt: $P^* \leftarrow \mathcal{M}(\pi)$ ▷ Single LLM call
return P^*

A Algorithm Pseudocode

B Prompt Evolution (SummEval)

In Table 6, we compare the Manual baseline prompt with the Pref2Point-O's prompt on SummEval (seed 0, Claude Sonnet 4.5) to illustrate what PREF2POINT discovers from pairwise preferences. MSE improves from 18.7 (Manual) to 15.6 (Pref2Point-O), QWK from 0.349 to 0.439.

C Score Calibration

Figure 1 reports the Expected Calibration Error normalized by each dataset's scale span (mean over 6 seeds). Pref2Point-Z achieves the lowest calibration error on LLMJudge (9.8%) and DREsS (12.0%, approaching Oracle at 6.7%), but degrades on HelpSteer2 (28.4%) and SummEval (32.9%). Pref2Point-O recovers calibration on these datasets to near-Manual levels (20.0% and 20.8% vs. 20.0% and 19.5%). After normalization, WMT's calibration (14–17%) is comparable to narrower scales, indicating that scale width alone does not determine calibration difficulty.

D Cross-Model Prompt Transferability

In our main experiments, each LLM uses a prompt optimized specifically for that model. To test whether optimized prompts are model-specific or transferable, we evaluate all source→scorer combinations across three models and five datasets for Pref2Point-O. Table 7 reports QWK averaged over all datasets and seeds.

Off-diagonal entries are competitive with or exceed the diagonal (same-model) entries, indicating that optimized prompts transfer across models without significant degradation. GPT-120B prompts achieve the best performance on all three scorers,

Manual — Generic rubric (MSE = 18.7, QWK = 0.349)	**Pref2Point-O** — Learned from preferences (MSE = 15.6, QWK = 0.439)
"Evaluate based on these criteria: Content Coverage (8 pts), Accuracy (6 pts), Conciseness (3 pts), Coherence (3 pts)."	*"Captures all or nearly all key facts, main events, named entities, statistics, and essential context. Includes critical details like specific numbers, names, locations, and outcomes that make the story newsworthy."*
Criteria (one-line descriptions)	**Criteria (operationalized per level)**
1. Content Coverage: "Captures key facts, main events, and essential information."	1. Content Coverage: 4 levels with concrete indicators (e.g., 7–8: "named entities, statistics, outcomes"; 3–4: "only basic facts").
2. Accuracy: "No factual errors, misrepresentations, or hallucinations."	2. Accuracy: 4 levels distinguishing severity (e.g., "slightly imprecise wording" vs. "fabricated information").
3. Conciseness: "Appropriate length, no redundancy."	3. Conciseness: 3 levels ("every sentence adds value" vs. "repetitive information").
4. Coherence: "Logical flow, clear structure, readable."	4. Coherence: 3 levels ("smooth transitions" vs. "disconnected fragments").
Limitations	**Evaluation principles (absent from Manual)**
▷ No guidance on what counts as a "key fact" vs. secondary detail.	▷ "A single major factual error should significantly lower the accuracy score."
▷ No severity distinctions (minor imprecision = hallucination?).	▷ "Summaries that repeat the same information in slightly different wording should lose conciseness points."
▷ No decision rules for borderline cases.	▷ "Consider what information is essential for understanding why the story is newsworthy."
	Severity taxonomy (absent from Manual)
	▷ Minor: slightly awkward phrasing, missing one secondary detail.
	▷ Major: factual errors, missing multiple key facts, misleading constructions.
	Key discoveries
	▷ Operationalizes each score level with observable features.
	▷ Distinguishes "imprecise wording" from "fabrication" (accuracy calibration).
	▷ Defines newsworthiness as the completeness criterion (names, numbers, outcomes).

Table 6: Manual vs. Pref2Point-O prompts on SummEval. The algorithm discovers from pairwise preferences that (1) each score level needs concrete observable indicators rather than abstract descriptions, (2) explicit severity distinctions are needed to separate minor imprecision from hallucination, and (3) "completeness" must be grounded in newsworthiness (named entities, statistics, outcomes). The Manual says *what* to evaluate; Pref2Point-O says *how to recognize* each quality level in practice.

Prompt \ Scorer	Sonnet 4.5	GPT-120B	GPT-20B
Sonnet 4.5	0.408	0.375	0.353
GPT-120B	**0.420**	**0.421**	**0.378**
GPT-20B	0.398	0.412	0.374

Table 7: Cross-model prompt transfer (mean QWK ↑ across all datasets). Rows = prompt source; columns = scoring model. Bold = best per column.

including outperforming Claude's own prompt on Claude. Inspecting the prompts reveals a structural explanation: GPT-120B's optimization produces more prescriptive rubrics with explicit scoring trade-off rules (e.g., "perfect accuracy but minimal coverage → 10–13 range") and concrete checklists of required content elements, whereas Claude's prompts rely more on abstract evaluation principles. The prescriptive style leaves less ambiguity for any scorer to resolve independently, making it more portable across models of varying capability.

E Pref2Point-O Implementation

Figure 2 provides a minimal self-contained implementation of Algorithm 2 using the DSPy framework and GEPA (Khattab et al., 2023; Agrawal et al., 2025). The key design is that a *single* pointwise scoring prompt is applied to both documents in a pair; GEPA's reflection LM rewrites this prompt based on natural-language feedback derived from pairwise disagreements.

The critical properties of this design are: (1) the optimised artifact is a *pointwise* rubric that requires no pairwise context at inference time; (2) GEPA's reflection LM receives the annotator's natural-language rationale on each disagreement, enabling

Algorithm 2 Pref2Point-O Optimization-Based Pointwise Prompt Generation

Input: Pairwise dataset $\mathcal{D} = \{(x_i, x_j, y_{ij}, E_{ij})\}$, initial prompt population $\mathcal{P}_0$, optimization budget B
Output: Optimized pointwise prompt P^*
Initialize: $\mathcal{P} \leftarrow \mathcal{P}_0$, $t \leftarrow 0$
while $t < B$ and not converged **do**
  Sample minibatch: $\mathcal{B} \subset \mathcal{D}$
  for each candidate prompt $P \in \mathcal{P}$ **do**
    // Evaluate on minibatch
    for each $(x_i, x_j, y_{ij}, E_{ij}) \in \mathcal{B}$ **do**
      Compute scores: $\hat{s}_P(x_i)$, $\hat{s}_P(x_j)$
      Record disagreement: $\mathbf{1}[\hat{s}_P(x_i) \leq \hat{s}_P(x_j) \wedge y_{ij} = 1]$
    end for
    // Compute loss and generate feedback
    $\mathcal{L}(P) \leftarrow$ pairwise disagreement rate on $\mathcal{B}$
    $F(P) \leftarrow$ construct textual feedback including:
      - Scalar performance: $1 - \mathcal{L}(P)$
      - Failed pairs and their reasoning $\{E_{ij}\}$
      - Dimension analysis from $\{E_{ij}\}$
  end for
  // Selection and mutation via GEPA
  Rank candidates by $\mathcal{L}(P)$
  $\mathcal{P}_{\text{elite}} \leftarrow$ select top-k prompts from $\mathcal{P}$
  // Generate new candidates using reflection model
  for each $P \in \mathcal{P}_{\text{elite}}$ **do**
    $P' \leftarrow$ GEPA-mutate$(P, \mathcal{L}(P), F(P))$ ▷ Use feedback
    $\mathcal{P}_{\text{new}} \leftarrow \mathcal{P}_{\text{new}} \cup \{P'\}$
  end for
  $\mathcal{P} \leftarrow \mathcal{P}_{\text{elite}} \cup \mathcal{P}_{\text{new}}$
  $t \leftarrow t + 1$
end while

$P^* \leftarrow \text{argmin}_{P \in \mathcal{P}} \mathcal{L}(P)$ ▷ Return best prompt
return P^*

it to incorporate human quality criteria into the rubric; and (3) the shared `rater` ensures both documents are scored under identical instructions, so preference consistency emerges from calibrated absolute scores.

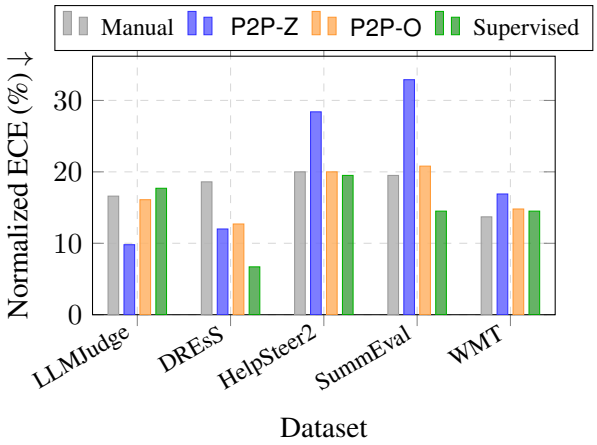

Figure 1: Normalized Expected Calibration Error (ECE as % of rating scale, mean over 6 seeds, Claude-4.5) across the four methods. Lower is better.

```python
import dspy

class PointwiseRater(dspy.Signature):
  """Rate the item given the query."""
   query = dspy.InputField()
   document = dspy.InputField()
   explanation: str = dspy.OutputField()
   rating: int = dspy.OutputField()

class Pref2PointModule(dspy.Module):
   """Apply the SAME rater to both sides of a pair."""
   def __init__(self):
      self.rater = dspy.Predict(PointwiseRater)

   def forward(self, example_A, example_B):
      a = self.rater(query=example_A["query"],
                     document=example_A["document"])
      b = self.rater(query=example_B["query"],
                     document=example_B["document"])
      return dspy.Prediction(
         A_rating=a.rating, B_rating=b.rating)

def pairwise_metric(gold, pred, **kwargs):
   """1 iff induced preference matches gold; else return
   human rationale as feedback to reflection LM."""
   a, b = int(pred.A_rating), int(pred.B_rating)
   pred_pref = "A" if a > b else ("B" if b > a else "tie")
   if pred_pref == gold.true_preference:
      return dspy.Prediction(score=1.0, feedback="")
   return dspy.Prediction(score=0.0,
      feedback=f"Predicted {pred_pref}, gold "
        f"{gold.true_preference}. "
        f"Rationale: {gold.reasoning}")

module = Pref2PointModule()
module.rater.signature = (
   module.rater.signature.with_instructions(seed_rubric))
optimized = dspy.GEPA(
   metric=pairwise_metric,
   reflection_lm=reflection_lm, auto="light"
).compile(module, trainset=pairs, valset=val_pairs)
# Output: standalone pointwise rubric
rubric = optimized.rater.signature.instructions
```

Figure 2: Minimal implementation of Pref2Point-O. The optimised artifact is a pointwise rubric requiring no pairwise context at inference.

F Meta-Prompts

F.1 Pairwise Judge Prompt

The following prompt is used to elicit structured pairwise preferences with rationales (Step 1 of both algorithms). The `{query}`, `{A_doc}`, and `{B_doc}` fields are filled per instance.

Pairwise Judge — Structured Preference Elicitation

Compare two documents (A and B) retrieved for the same user query, and decide which is more relevant.

Calibration guidance

▷ Items will be rated on an integer scale of $[r_{\min}, r_{\max}]$.

▷ Reference how many points apart you would rate them.

▷ Decent-vs-decent pairs: gap is typically 0–1 point.

▷ Only invoke a large gap when one side has a critical failure the other lacks.

▷ Do not escalate a single minor weakness into a multi-point gap.

Decision rule

▷ Return "equal" only if A and B are indistinguishable on every axis.

▷ If either dominates on at least one axis, pick that side.

Required output format

`Axes`: 3–6 short quality-dimension names.

`Per-axis`: For each axis, which side is stronger + concrete evidence. End each with `Winner: A|B|eq`.

`Weighting`: Which axes dominated and why; note close calls.

`Preference: A|B|equal`

Table 8: Pairwise judge meta-prompt (dataset-agnostic structure; the task preamble is adapted per dataset). Produces structured rationales consumed by Pref2Point-Z.

F.2 P2P-Z Distillation Meta-Prompt

The following prompt instructs the rewriter LLM to synthesize a pointwise scoring rubric from a batch of pairwise rationales. In iterative mode, it is called repeatedly with the previous output as `previous_draft`.

Rubric Rewriter — Distillation from Pairwise Rationales

You are a senior evaluation-prompt designer. Given a task description, a previous rubric draft, and a batch of pairwise preference rationales, write a new pointwise scoring rubric.

Objective

▷ Identify quality dimensions and their relative weighting from the rationales.

▷ Produce a rubric that, applied to a single item, yields ratings consistent with the pairwise preferences.

▷ Treat the previous draft as a starting point, not a constraint—rewrite freely.

Required outputs

`axes`: 3–6 short names for quality dimensions.

`axis_weighting`: Which axes matter most and why (grounded in rationales).

`scoring_anchors`: What a top-of-range and bottom-of-range item looks like concretely.

`rubric_body`: Self-contained rubric text for a scorer (no "Rating:" line—appended by code).

Constraints on rubric_body

▷ Must not mention pairwise comparisons, rationales, or training artefacts.

▷ Must not name the dataset or benchmark.

▷ Must not impose absolute score ceilings ("caps at 8", "maximum X").

▷ Bottom of range reserved for items severely deficient across multiple axes.

Table 9: Distillation meta-prompt for Pref2Point-Z. The structured output fields are assembled deterministically into the final rubric with a fixed `Rating: <integer min-max>` footer.

F.3 Pref2Point-O Feedback Template

Algorithm 2 uses GEPA for reflective prompt evolution. The optimizer scores each pairwise example by applying the candidate pointwise rubric to both items and checking whether the predicted ordering matches the gold preference. The following feedback is passed to the reflection LM to guide prompt mutations.

GEPA Pairwise-Disagreement Feedback

On mismatch (predicted ordering $\neq$ gold)

"Predicted ratings indicate document `{pred}` rates higher but a human annotator judged document `{gold}` should rate higher. Score = 0."

"Quality dimensions the annotator weighed: `{axes}`."

"Human explanation: `{rationale}`."

On match (predicted ordering = gold)

"Based on your predictions document `{pred}` rates higher and a human annotator agrees. Score = 1."

"Quality dimensions the annotator weighed: `{axes}`."

On tie (either side predicts equal)

"No feedback. Ignore this sample and do not make any decision based on it."

Anti-meta-pollution guard (appended to all feedback)

"The rubric is a POINTWISE scoring prompt. Add concrete quality criteria, axes, and range anchors that reflect the patterns in the feedback above so the scorer uses the full rating range. Do not phrase the rubric as a pair comparison and do not reference pairs, preferences, or annotator explanations in the rubric itself."

Table 10: Feedback template for Pref2Point-O. The reflection LM sees this per-example feedback and mutates the candidate rubric accordingly. The anti-meta guard prevents the optimized prompt from drifting toward pairwise language. The gold preference here is the pairwise predicted preference but presented to LLM as gold.