From Preferences to Scores: Semi-Supervised Multi-Trait Essay Scoring with LLM Pairwise Preferences and Few Human Anchors

Takumi Shibata and Yuichi Miyamura
Innovation & Technology R&D, Deloitte Touche Tohmatsu LLC
{takumi.shibata, yuichi.miyamura}@tohmatsu.co.jp

Abstract

Multi-trait automated essay scoring (AES) provides trait-level scores as feedback but requires costly annotations across multiple traits to train accurate scoring models. To alleviate this annotation burden, we propose **MTPAS** (**M**ulti-**T**rait **P**reference-to-**A**bsolute **S**coring), a semi-supervised framework that learns relative score orderings from LLM-generated pairwise preferences and calibrates them to absolute rubric scales using a small set of human-scored *anchor essays* in an end-to-end manner. To this end, MTPAS jointly optimizes three objectives: a Bradley–Terry pairwise preference loss, a regression loss on the anchor essays, and a Sinkhorn divergence-based distribution regularizer. The model adopts a *bifactor*-inspired head parameterizing trait scores as a shared general factor plus trait-specific residuals, empirically improving robustness to noisy LLM preferences. We further introduce *pairwise rubric compression*, which reformulates the original score-level rubrics into concise comparative criteria, reducing the inference cost of collecting LLM preferences. On ASAP and ASAP++, MTPAS achieves 91.4% of the fully supervised SoTA performance with only 30 anchor essays per prompt and further narrows the gap with 200 anchors. Under the same anchor budget, MTPAS outperforms conventional multi-trait AES baselines at every anchor size investigated in this study.

1 Introduction

Automated essay scoring (AES) is the task of automatically assigning scores to student essays using machine learning, and has received growing attention as a means of reducing grading costs and improving assessment consistency (Uto, 2021).

AES research is broadly divided into two paradigms depending on the granularity of the score (Li and Ng, 2025; Zhao et al., 2026): *holistic scoring*, which assigns a single overall score to each essay, and *multi-trait scoring*, which assigns separate scores along multiple rubric-defined traits such as content, organization, word choice, and conventions.

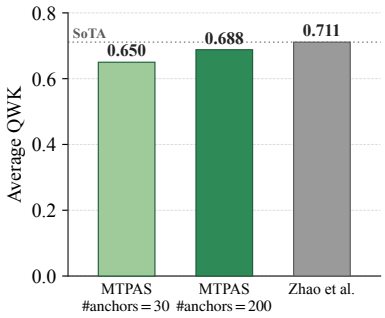


Figure 1: MTPAS uses only 30 or 200 anchor essays, whereas the model of Zhao et al. (2026) is trained on the full dataset.

Holistic scoring has been extensively studied (Taghipour and Ng, 2016; Alikaniotis et al., 2016), with recent methods incorporating hierarchical attention mechanisms (Dong et al., 2017) or ranking-aware objectives. For example, R^2BERT (Yang et al., 2020) augments BERT (Devlin et al., 2019) fine-tuning with a batch-wise ranking loss, while NPCR (Xie et al., 2022) formulates AES as pairwise contrastive regression.

Multi-trait scoring has attracted increasing interest because it can provide richer, more interpretable feedback to learners (Mathias and Bhattacharyya, 2020). Recent methods include text-to-score generation with T5 (Do et al., 2024a; Raffel et al., 2020), graph-based modeling of inter-trait correlations (Li and Ng, 2025), and graph-based scoring enhanced with multiple knowledge sources such as linguistic features and ordinal score structure (Zhao et al., 2026).

Despite this progress, existing multi-trait AES methods require *large amounts* of human-scored training data. This requirement is especially costly in multi-trait scoring, where each essay must be annotated across multiple rubric dimensions, and

Proceedings of the 15th International Joint Conference on Natural Language Processing and the 5th Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics, pages 6041–6057
November 6-10, 2026 ©2026 Association for Computational Linguistics

becomes prohibitive when new prompts (i.e., writing tasks or essay topics) or rubrics are introduced. Reducing this annotation burden is therefore crucial for practical multi-trait AES deployment.

To address this challenge, we propose a semi-supervised framework for high-accuracy multi-trait scoring that leverages a large pool of unscored essays together with only a small set of human-scored essays. Our key idea is to exploit auxiliary supervision from large language models (LLMs). Recent zero-shot AES work, such as LCES (Shibata and Miyamura, 2025), has achieved higher scoring accuracy than direct LLM score generation by aggregating LLM-generated pairwise preferences into scores via a learning-to-rank method (Burges et al., 2005). This suggests that LLM pairwise preferences can provide useful training signals when human scores are scarce.

Building on this insight, we propose **MTPAS** (**M**ulti-**T**rait **P**reference-to-**A**bsolute **S**coring). MTPAS learns trait scores by jointly leveraging LLM-generated pairwise preferences for relative ordering and a small number of human-scored *anchor essays* for absolute calibration to rubric-defined scales. These signals are integrated end-to-end through a Bradley–Terry pairwise preference loss (Bradley and Terry, 1952) for ranking, together with an anchor regression loss and a Sinkhorn divergence-based distribution regularizer for calibration.

However, methods that rely on LLM-generated preferences face two key limitations: the inference cost of LLM pairwise comparisons, and label noise from imperfect LLM judgments. To mitigate these, we further propose *pairwise rubric compression*, which condenses the original score-level rubrics into short comparative criteria to reduce inference cost, and a scoring model inspired by the bifactor model (Holzinger and Swineford, 1937; Reise, 2012). The model decomposes each trait score into a general factor and a trait-specific residual, improving robustness to label noise.

On ASAP and ASAP++ (Mathias and Bhattacharyya, 2018), we evaluate MTPAS at several anchor sizes against two references: the fully supervised state-of-the-art (SoTA)[1], and conventional multi-trait scoring models trained under matched anchor budgets. As shown in Figure 1, MTPAS reaches 91.4% of the fully supervised SoTA ceiling with only 30 anchors per prompt, corresponding to 2.3% of the scored essays used to train conventional fully supervised methods. With 200 anchors per prompt, corresponding to 15.4%, a gap remains but is not large enough to reach statistical significance under a paired t-test at the 5% level. In matched-anchor-size comparisons, MTPAS outperforms conventional multi-trait AES baselines in average QWK across all anchor sizes examined.

Our contributions are as follows: (1) We propose MTPAS, a semi-supervised multi-trait AES framework that integrates LLM pairwise preferences with a small set of human-scored anchor essays. (2) We also introduce pairwise rubric compression to reduce LLM inference cost and a scoring model inspired by the bifactor model to improve robustness to noisy preference labels. (3) We demonstrate that MTPAS approaches the fully supervised ceiling with limited anchors and consistently outperforms baselines trained with the same number of anchor essays.

2 Related Work

2.1 Multi-Trait AES

Multi-trait AES assigns separate scores along multiple rubric-defined traits for a single essay (Hussein et al., 2020; Ridley et al., 2020, 2021; Shibata and Uto, 2022; Chen and Li, 2024; Li and Pan, 2025). ArTS (Do et al., 2024a) reframed multi-trait scoring as a text-to-text generation problem using T5, where the model receives an essay as input and generates a string containing all trait scores. Several extensions have built on this generation paradigm: RMTS (Chu et al., 2025) augments the input with LLM-generated rationales for each trait, and SaMRL (Do et al., 2024b) applies reinforcement learning to improve generation quality. GAT-AES (Li and Ng, 2025) models inter-trait correlations using graph attention networks, capturing the relational structure among traits. T-MES (Wang and Liu, 2025) introduces a trait-aware mixture-of-experts encoder that routes essays through expert sub-networks specialized for each trait. Zhao et al. (2026) couple a graph neural scorer with multiple knowledge sources such as linguistic features and the ordinal structure of scores, and, to our knowledge, currently report the highest QWK on ASAP/ASAP++.

Unlike these methods, which rely on large amounts of human-scored essays, MTPAS operates with scarce human scores and compensates with LLM pairwise preferences, an auxiliary signal

[1]Because the training paradigms and data splits differ, MTPAS and the fully supervised baselines are not evaluated on identical test sets; see Section 4.3 for details.

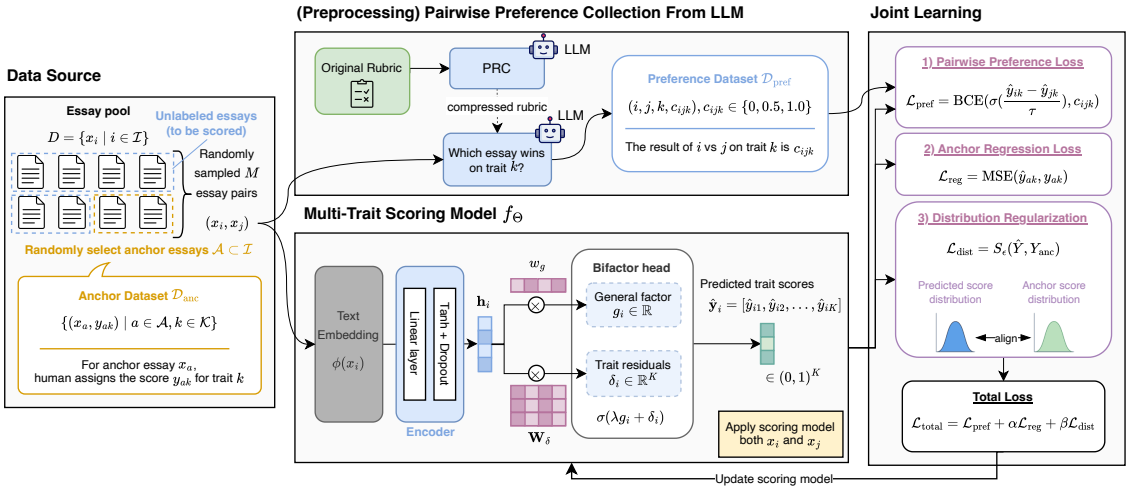


Figure 2: Framework of MTPAS. Given the human-scored anchor set $\mathcal{D}_{\text{anc}}$ and the preference set $\mathcal{D}_{\text{pref}}$ constructed beforehand from LLM-generated pairwise preferences, MTPAS learns a multi-trait scoring model $f_{\boldsymbol{\Theta}}$ by jointly optimizing three loss functions.

unexploited by prior multi-trait AES.

2.2 LLM-Assisted Semi-Supervised Learning

Several prior studies use LLMs as annotators in semi-supervised settings. Zhang et al. (2023) train named entity recognition and relation extraction models using LLM-generated annotations. Park and Caragea (2024) use LLM-generated examples and their associated pseudo-labels for semi-supervised natural language inference. Zhao et al. (2024) use an LLM teacher's outputs as pseudo-labels to train student models for semi-supervised generation.

However, these methods supplement scarce human labels by using LLMs as direct labelers, which is unreliable in AES due to the low accuracy of LLM-based absolute scoring (Yancey et al., 2023; Mansour et al., 2024; Yoshida, 2024, 2025). MTPAS adopts a different paradigm: it elicits LLM pairwise preferences, a more reliable signal that aligns with human judgment (Zheng et al., 2023; Lee et al., 2024; Liusie et al., 2024; Shibata and Miyamura, 2025), and uses a small set of human-labeled anchors to calibrate them to absolute rubric scales, while jointly modeling multiple traits.

3 Proposed Method

MTPAS jointly learns relative score orderings from LLM-generated pairwise essay preferences and calibrates them to the absolute rubric scale using a small set of human-scored anchor essays. The overall framework of MTPAS is illustrated in Figure 2. We begin with the task formulation.

3.1 Task Formulation

Given a set of essays $\mathcal{D} = \{x_i \mid i \in \mathcal{I}\}$, where $\mathcal{I} = \{1, \ldots, N\}$ is the index set and N denotes the number of essays, our goal is to predict trait-specific scores $\{y_{ik} \mid i \in \mathcal{I}, k \in \mathcal{K}\}$, where y_{ik} is the score of essay x_i on the k-th trait, $\mathcal{K} = \{1, \ldots, K\}$ is the set of traits, and K is the total number of traits.

We assume that a small subset of essays $\{x_a \mid a \in \mathcal{A} \subset \mathcal{I}\}$, referred to as *anchor essays* and sampled uniformly at random from $\mathcal{D}$, has been human-scored, yielding gold-standard scores $\{y_{ak} \mid a \in \mathcal{A}, k \in \mathcal{K}\}$ with $|\mathcal{A}| \ll N$. Our objective is to train a scoring model $f_{\boldsymbol{\Theta}}$ that predicts accurate trait-specific scores $\hat{y}_{ik}$ for each non-anchor essay x_i, where $i \in \mathcal{I} \setminus \mathcal{A}$.

Since different traits may have different score ranges, during training we apply min–max normalization to all gold-standard scores to $[0, 1]$ using the rubric-defined minimum and maximum scores. Throughout this paper, y_{ik} denotes the normalized score unless otherwise stated. At inference time, predictions are rescaled to the original rubric range.

3.2 Collecting Pairwise Preferences

Before training the scoring model $f_{\boldsymbol{\Theta}}$, we collect pairwise preferences over a large set of essay pairs to provide a ranking signal. Following LCES (Shibata and Miyamura, 2025), we randomly sample M essay pairs $\mathcal{P}_{\text{sub}}$ from the $N(N-1)/2$ possible pairs and present each pair to the LLM.

For each sampled pair $(i, j) \in \mathcal{P}_{\text{sub}}$ and each trait k, the LLM determines which essay is superior,

producing a comparison label $c_{ijk} \in \{0, 0.5, 1\}$: $c_{ijk} = 1$ if x_i is judged better than x_j on trait k, $c_{ijk} = 0$ if x_j is better, and $c_{ijk} = 0.5$ if the two are judged equivalent. This yields the preference dataset $\mathcal{D}_{\text{pref}} = \{(i, j, k, c_{ijk}) \mid (i, j) \in \mathcal{P}_{\text{sub}}, k \in \mathcal{K}\}$ with $|\mathcal{P}_{\text{sub}}| = M$.

Pairwise Rubric Compression. A naive approach queries the LLM separately for each trait, requiring K queries per essay pair and $M \times K$ queries in total. Furthermore, original rubrics are often verbose, inflating per-query token consumption. To mitigate both costs, we propose *pairwise rubric compression* (PRC).

The key observation is that standard rubrics contain detailed score-level descriptors (e.g., "A score of 4 indicates…") that, while essential for absolute scoring, are redundant for pairwise comparison: it suffices to describe what factors make one essay better than another and when the two should be considered equivalent. PRC therefore distills each trait's rubric into a *win condition* and a *tie rule* via an LLM rewriting step applied to the original rubric. The LLM prompt for PRC is provided in Appendix A. Using the compressed rubric, we elicit preferences for all traits in a single LLM call per pair, so the total number of LLM queries remains M regardless of K.

3.3 Multi-Trait Scoring Model

We equip the scoring model $f_{\boldsymbol{\Theta}}$ with an output head inspired by bifactor models (Holzinger and Swineford, 1937; Reise, 2012; Brunner et al., 2012). Developed in educational and psychological measurement, these models decompose observed scores into general and trait-specific factors. Our head similarly computes each trait score from a weighted shared general factor and a trait-specific residual.

Encoder. Given an essay $x_i \in \mathcal{D}$, we first map it to a fixed embedding $\boldsymbol{\phi}(x_i) \in \mathbb{R}^{d_{\text{in}}}$ using a frozen pre-trained text embedding model ϕ. This embedding is then projected into a latent representation through a single fully-connected layer with $\tanh$ activation and dropout:

$$\boldsymbol{h}_i = \text{Dropout}(\tanh(\boldsymbol{W}\,\boldsymbol{\phi}(x_i) + \boldsymbol{b})), \quad (1)$$

where $\boldsymbol{W} \in \mathbb{R}^{d \times d_{\text{in}}}$ is a weight matrix, $\boldsymbol{b} \in \mathbb{R}^d$ is a bias vector, d_{in} is the dimensionality of the input embedding, and d is the hidden dimension.

Bifactor Decomposition. From the representation $\boldsymbol{h}_i$, we extract a scalar general factor $g_i \in \mathbb{R}$ and a vector of trait-specific residuals $\boldsymbol{\delta}_i \in \mathbb{R}^K$:

$$g_i = \boldsymbol{w}_g^\top \boldsymbol{h}_i + b_g, \quad (2)$$

$$\boldsymbol{\delta}_i = \boldsymbol{W}_\delta \boldsymbol{h}_i + \boldsymbol{b}_\delta, \quad (3)$$

where $\boldsymbol{w}_g \in \mathbb{R}^d$, $b_g \in \mathbb{R}$, $\boldsymbol{W}_\delta \in \mathbb{R}^{K \times d}$, and $\boldsymbol{b}_\delta \in \mathbb{R}^K$.

The predicted score vector $\hat{\boldsymbol{y}}_i = [\hat{y}_{i1}, \ldots, \hat{y}_{iK}]^\top$ is then computed as:

$$\hat{\boldsymbol{y}}_i = \sigma(\boldsymbol{\lambda} g_i + \boldsymbol{\delta}_i), \quad (4)$$

where $\boldsymbol{\lambda} = [\lambda_1, \ldots, \lambda_K]^\top \in \mathbb{R}^K$ is a learnable vector of factor loadings that modulates the contribution of the general factor to each trait, and $\sigma(\cdot)$ denotes the element-wise sigmoid function that maps scores to $(0, 1)$.

In practice, the model is trained on essay pairs (x_i, x_j) for the ranking-based objectives introduced in the following section. For each pair, we apply the same scoring model $f_{\boldsymbol{\Theta}}$ independently to both essays, yielding a pair of predicted score vectors $(\hat{\boldsymbol{y}}_i, \hat{\boldsymbol{y}}_j)$.

The full set of learnable parameters is $\boldsymbol{\Theta} = \{\boldsymbol{W}, \boldsymbol{b}, \boldsymbol{w}_g, b_g, \boldsymbol{W}_\delta, \boldsymbol{b}_\delta, \boldsymbol{\lambda}\}$.

3.4 Model Learning

The model is trained end-to-end by minimizing a composite loss:

$$\mathcal{L} = \mathcal{L}_{\text{pref}} + \alpha \mathcal{L}_{\text{reg}} + \beta \mathcal{L}_{\text{dist}}, \quad (5)$$

where α and β are hyperparameters controlling the contribution of each term. Each loss term is described in turn below.

3.4.1 Pairwise Preference Loss

For each pair (i, j) and trait k in $\mathcal{D}_{\text{pref}}$, we use the Bradley–Terry model (Bradley and Terry, 1952) to map the predicted score difference into the probability that essay x_i outranks x_j on trait k:

$$P(i \succ_k j) = \sigma((\hat{y}_{ik} - \hat{y}_{jk})/\tau), \quad (6)$$

where $\tau > 0$ is a temperature that compensates for the limited dynamic range of $\hat{y}_{ik} \in [0, 1]$, sharpening the mapping from score differences to preference probabilities.

The standard Bradley–Terry model describes only binary win/loss outcomes and does not by itself define a tie outcome. To accommodate the tie label $c_{ijk} = 0.5$, we treat $c_{ijk} \in \{0, 0.5, 1\}$ as a soft target and define the pairwise preference loss as a soft-label binary cross-entropy (BCE) surrogate

built on top of the Bradley–Terry score difference:

$$\mathcal{L}_{\text{pref}} = -\frac{1}{|\mathcal{D}_{\text{pref}}|} \sum_{\mathcal{D}_{\text{pref}}} \big[c_{ijk} \log P(i \succ_k j) + (1 - c_{ijk}) \log(1 - P(i \succ_k j))\big]. \quad (7)$$

This loss drives $\hat{y}_{ik}$ to be consistent with the LLM's pairwise preferences and operates over *all* essays in $\mathcal{D}$, including the anchor essays whose predictions are anchored to the rubric scale by the regression loss introduced next.

3.4.2 Anchor Regression Loss

Pairwise preference learning captures only *relative* score orderings. To calibrate predictions to the absolute rubric scale, we minimize the mean squared error (MSE) between predicted scores and gold-standard scores on the anchor set $\mathcal{D}_{\text{anc}} = \{(x_a, y_{ak}) \mid a \in \mathcal{A}, k \in \mathcal{K}\}$:

$$\mathcal{L}_{\text{reg}} = \frac{1}{|\mathcal{A}|K} \sum_{a \in \mathcal{A}} \sum_{k \in \mathcal{K}} (\hat{y}_{ak} - y_{ak})^2. \quad (8)$$

Beyond absolute calibration, $\mathcal{L}_{\text{reg}}$ also mitigates LLM-preference noise by directly supervising a subset of predictions with ground-truth scores.

3.4.3 Distribution Regularization

$\mathcal{L}_{\text{reg}}$ directly supervises only the $|\mathcal{A}|$ anchor predictions, leaving non-anchor essays constrained only by the relative orderings from $\mathcal{L}_{\text{pref}}$. Since anchors are randomly sampled, their empirical distribution approximates the true score distribution, motivating a distribution-level signal that aligns predicted scores with this distribution. Moreover, aligning the joint distribution over K traits regularizes inter-trait dependencies.

We instantiate this signal as the Sinkhorn divergence (Cuturi, 2013; Cuturi and Peyré, 2016; Feydy et al., 2019) between predicted and anchor score vectors viewed as point clouds in $\mathbb{R}^K$. Let $\hat{\boldsymbol{Y}} = \{\hat{\boldsymbol{y}}_i\}_{i \in \mathcal{I}}$ denote the set of predicted score vectors and $\boldsymbol{Y}_{\text{anc}} = \{\boldsymbol{y}_a\}_{a \in \mathcal{A}}$ denote the set of anchor score vectors, where $\boldsymbol{y}_a = [y_{a1}, \ldots, y_{aK}]^\top$ collects the gold-standard trait scores for anchor essay x_a:

$$\mathcal{L}_{\text{dist}} = S_\varepsilon(\hat{\boldsymbol{Y}}, \boldsymbol{Y}_{\text{anc}}), \quad (9)$$

where S_ε denotes the Sinkhorn divergence with entropic regularization parameter ε.

4 Experiments

We evaluated the scoring performance of MTPAS and compared it against established multi-trait scoring models.

4.1 Datasets

We evaluated on the **ASAP**[2] and **ASAP++** (Mathias and Bhattacharyya, 2018) datasets, the most widely used benchmarks for multi-trait AES. ASAP contains 12,978 student essays across 8 prompts (P1–P8), each with a holistic score; trait scores are provided only for prompts 7 and 8. ASAP++ extends ASAP by providing trait scores for the same essays on prompts 1–6, with 5–6 traits per prompt. Table 9 in Appendix B provides per-prompt details.

4.2 Baselines

We compared MTPAS against six representative multi-trait scoring methods reviewed in Section 2: **ArTS** (Do et al., 2024a), **SaMRL** (Do et al., 2024b), **RMTS** (Chu et al., 2025), **GAT-AES** (Li and Ng, 2025), **T-MES** (Wang and Liu, 2025), and the model of Zhao et al. (2026).

4.3 Implementation Details

LLMs for Pairwise Comparison. We used three LLMs as pairwise judges: **Gemma 4 E4B**[3], **Qwen 3.5 9B** (Qwen Team, 2026)[4], and **GPT-5.4 mini** (gpt-5.4-mini-2026-03-17); we abbreviate them as Gemma 4, Qwen 3.5, and GPT-5.4m hereafter. For each prompt, we sampled $M = 5{,}000$ essay pairs and queried each LLM to compare all traits simultaneously using rubrics generated by PRC. PRC rewriting itself was performed by GPT-5.4 (gpt-5.4-2026-03-05).

Hyperparameters. The text embedding model ϕ was OpenAI's `text-embedding-3-large` ($d_{\text{in}} = 3{,}072$). The fully-connected layer had hidden dimension $d = 16$ with $\tanh$ activation and dropout rate 0.5. We used AdamW (Loshchilov and Hutter, 2017) with learning rate 5×10^{-4} and weight decay 10^{-3}, training for 15 epochs. Loss weights were set to $\alpha = 4.0$ and $\beta = 2.5$, with temperature $\tau = 0.2$. Sinkhorn divergence used entropic regularization parameter $\varepsilon = 2.5 \times 10^{-3}$. Appendix D lists the full training hyperparameters.

Evaluation Metric. We adopted quadratic weighted kappa (QWK) (Cohen, 1968), the standard measure in AES research (Taghipour and Ng, 2016; Zhao et al., 2026), as the primary evaluation metric.

[2] https://www.kaggle.com/c/asap-aes
[3] https://huggingface.co/google/gemma-4-E4B-it
[4] https://huggingface.co/Qwen/Qwen3.5-9B

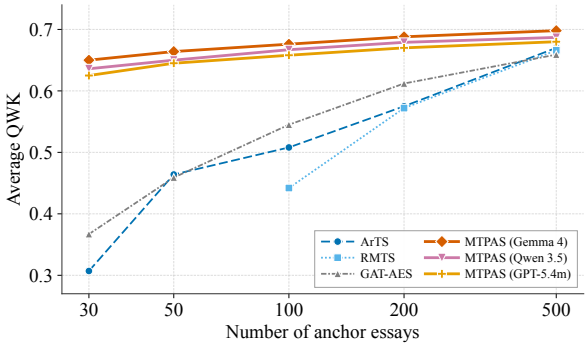


Figure 3: Average QWK across all traits and eight prompts as a function of the anchor size $|\mathcal{A}|$. RMTS was omitted at $|\mathcal{A}| \in \{30, 50\}$, where the training data were too scarce for convergence.

Method	$\|\mathcal{A}\|$=30	$\|\mathcal{A}\|$=50	$\|\mathcal{A}\|$=100	$\|\mathcal{A}\|$=200	$\|\mathcal{A}\|$=500
ArTS	43.2	65.3	71.5	80.9	94.3
RMTS	-	-	62.2	80.5	93.6
GAT-AES	51.7	64.5	76.6	86.1	92.7
MTPAS (Gemma 4)	**91.4**	**93.4**	**95.1**	**96.8**	**98.1**
MTPAS (Qwen 3.5)	89.4	91.4	93.8	95.5	96.6
MTPAS (GPT-5.4m)	87.9	90.7	92.6	94.2	95.6

Table 1: QWK ratio to SoTA (%): fraction of Full-Sup. Zhao et al. (2026) performance achieved by each method at varying anchor size $|\mathcal{A}|$. **Bold** indicates the best and underline the second best in each column.

Evaluation Procedure. Our experiments investigated how closely MTPAS can approach the ceiling attained under full human supervision when only a small fraction of essays is scored. To simulate this on ASAP/ASAP++, we treated the human trait scores as observed for only a randomly drawn subset of essays per prompt—the *anchor essays* $\mathcal{A}$—while masking them on the remaining essays, which form the *non-anchor essays*. We swept five anchor sizes per prompt, $|\mathcal{A}| \in \{30, 50, 100, 200, 500\}$, and evaluated MTPAS's predictions on the non-anchor essays.

We compared MTPAS against baselines through two complementary experiments. First, since ArTS, RMTS, and GAT-AES provide public training code, we re-trained each of them on the anchor essays alone at every $|\mathcal{A}|$ following its original recipe, and evaluated it on the same non-anchor essays as MTPAS. We repeated this experiment, including MTPAS, five times per setting with independently drawn anchor samples, and reported the mean QWK and standard deviation. Second, we compared against **Full-Sup.**, which comprises the QWK values reported by ArTS, SaMRL, RMTS, GAT-AES, T-MES, and Zhao et al. (2026) under the standard ASAP/ASAP++ 6:2:2 split, whose training set contains $12{,}978 \times 0.8 \approx 10{,}382$ scored essays across the 8 prompts (for reference, $|\mathcal{A}| = 30$ and 200 per prompt totaling 240 and 1,600 essays across the 8 prompts, or 2.3% and 15.4% of 10,382).

Note that the Full-Sup. baselines and MTPAS used different splits, and therefore different test sets, so the two did not constitute a strict head-to-head comparison. However, because anchors were sampled uniformly at random per prompt, the non-anchor essays match in expectation the distribution from which the Full-Sup. test sets are drawn. We therefore treat the Full-Sup. figures as an approximate *ceiling estimate* against which to measure recovery, and the matched-budget ArTS/RMTS/GAT-AES results as the directly comparable baseline.

4.4 Main Results

Human-Score Efficiency Figure 3 plots QWK as a function of the anchor size $|\mathcal{A}|$, where the reported value was obtained by first averaging QWK across the prompts in which each trait is defined, and then averaging across traits. Table 1 complements this view by reporting the *QWK ratio to SoTA*, which quantifies how closely each method approaches the Full-Sup. ceiling at a given anchor size $|\mathcal{A}|$. Among the Full-Sup. baselines, we adopted the strongest one—Zhao et al. (2026), whose average QWK of 0.711 is the highest—as the reference, and defined this metric as the ratio of a method's QWK to this value.

Two findings stand out. First, every MTPAS variant outperformed every baseline at every anchor size $|\mathcal{A}|$, with the largest margin at small anchor sizes ($|\mathcal{A}| \leq 100$). With only $|\mathcal{A}| = 30$ anchor essays per prompt (about 2.3% of the Full-Sup. training data), MTPAS (Gemma 4) already reached QWK 0.650, attaining 91.4% of SoTA performance, whereas ArTS and GAT-AES attained only 43.2% and 51.7%, respectively; even the weakest variant, MTPAS (GPT-5.4m), attained 87.9%, exceeding GAT-AES, the strongest baseline. Second, as the anchor size $|\mathcal{A}|$ grew, the baselines closed the gap: at $|\mathcal{A}| = 500$, all baselines exceeded 90%, and the absolute QWK differences shrank to a few points. MTPAS therefore delivers its largest benefit at small anchor sizes.

Per-Trait Comparison. Table 2 reports per-trait QWK at $|\mathcal{A}| = 200$, with each column averaged across the prompts where the trait is defined. MTPAS attained competitive accuracy on every trait

Method	Data	Over.	Cont.	Org.	WC	SF	Conv.	PA	Lang.	Narr.	Style	Voice	Avg. (SD)
ArTS (Do et al., 2024a)	Full-Sup.	0.754	0.730	0.751	0.698	0.725	0.672	0.668	0.679	0.678	**0.721**	0.570	0.695 (±0.018)
T-MES (Wang and Liu, 2025)	Full-Sup.	**0.774**	0.730	0.750	0.702	0.730	0.685	0.686	0.679	0.675	0.693	0.590	0.700 (-)
SaMRL (Do et al., 2024b)	Full-Sup.	0.754	0.735	0.751	0.703	0.728	0.682	0.685	0.688	0.691	0.710	0.627	0.705 (±0.013)
RMTS (Chu et al., 2025)	Full-Sup.	0.755	0.737	0.752	**0.713**	**0.744**	0.682	0.690	0.705	0.694	0.702	0.612	0.708 (±0.043)
GAT-AES (Li and Ng, 2025)	Full-Sup.	0.771	**0.742**	0.749	0.678	0.726	**0.694**	0.686	**0.709**	0.692	0.699	0.649	0.709 (±0.011)
Zhao et al. (2026)	Full-Sup.	0.770	0.736	**0.759**	0.708	**0.744**	0.691	0.686	0.691	0.699	0.714	0.623	**0.711** (-)
ArTS	$\|\mathcal{A}\|$=200	0.646	0.581	0.579	0.554	0.570	0.532	0.625	0.614	0.645	0.515	0.466	0.575 (±0.036)
RMTS	$\|\mathcal{A}\|$=200	0.588	0.607	0.552	0.553	0.555	0.498	0.658	0.652	0.680	0.556	0.394	0.572 (±0.025)
GAT-AES	$\|\mathcal{A}\|$=200	0.690	0.662	0.571	0.621	0.602	0.548	0.678	0.610	0.659	0.594	0.495	0.612 (±0.006)
MTPAS (Gemma 4)	$\|\mathcal{A}\|$=200	0.721	0.722	0.679	0.696	0.694	0.659	0.740	0.674	0.717	0.616	**0.656**	0.688 (±0.004)
MTPAS (Qwen 3.5)	$\|\mathcal{A}\|$=200	0.711	0.712	0.662	0.676	0.681	0.654	0.739	0.684	0.718	0.606	0.629	0.679 (±0.006)
MTPAS (GPT-5.4m)	$\|\mathcal{A}\|$=200	0.706	0.714	0.658	0.678	0.679	0.627	**0.743**	0.639	**0.721**	0.592	0.615	0.670 (±0.005)

Table 2: Per-trait QWK at $|\mathcal{A}| = 200$. Over.=Overall, Cont.=Content, Org.=Organization, WC=Word Choice, SF=Sentence Fluency, Conv.=Conventions, PA=Prompt Adherence, Lang.=Language, Narr.=Narrativity.

and achieved the best QWK on prompt adherence, narrativity, and voice. MTPAS also consistently outperformed the conventional models re-trained on the same anchor essays across almost all traits, regardless of the LLM. Among the Full-Sup. baselines, Zhao et al. (2026) obtained the highest average QWK, while MTPAS (Gemma 4) was the strongest MTPAS variant. A paired t-test between these two yielded $p = 0.136$ when paired by trait (11 traits) and $p = 0.093$ when paired by prompt (8 prompts) (Appendix F); neither test detected a statistically significant difference at the 5% level. Although MTPAS trailed the strongest baseline by a small absolute margin, it reached SoTA-level multi-trait scoring with only $|\mathcal{A}| = 200$ anchors per prompt, on average 15.4% of the Full-Sup. training data.

5 Ablation Study

We conducted ablation studies to analyze each MTPAS component. Unless otherwise stated, we report results at two representative budgets: $|\mathcal{A}| = 30$ (weak supervision) and $|\mathcal{A}| = 200$ (moderate supervision).

5.1 Pairwise Rubric Compression

We evaluated four configurations that vary along two axes: query strategy, either querying all traits jointly (**All**) or each trait independently (**Indep.**); and rubric format, either compressed rubrics (**PRC**) or original rubrics (**Original**)[5]. We report pairwise comparison accuracy and downstream QWK at $|\mathcal{A}| = 30$ and $|\mathcal{A}| = 200$. Acc was computed over all sampled pairs, treating LLM ties as correct only when the two gold scores are equal. $\text{Acc}_{\neq}$ was computed only on pairs with unequal gold scores, since equal-scored pairs admit no well-defined ground-truth preference direction.

[5]For GPT-5.4m, we omitted the **Original** configurations due to API cost.

LLM	Config	Acc (%)	$\text{Acc}_{\neq}$ (%)	Q_{30}	Q_{200}
Gemma 4	All+PRC	58.2	79.8	0.650	0.688
	All+Orig.	58.0	80.1	0.643	0.683
	Indep.+PRC	58.6	81.4	0.654	0.691
	Indep.+Orig.	58.6	81.4	0.655	0.693
Qwen 3.5	All+PRC	57.1	78.8	0.636	0.679
	All+Orig.	56.7	78.6	0.638	0.683
	Indep.+PRC	57.3	79.3	0.633	0.676
	Indep.+Orig.	57.5	79.8	0.630	0.675
GPT-5.4m	All+PRC	56.2	77.3	0.625	0.670
	Indep.+PRC	57.6	79.7	0.633	0.675

Table 3: Effect of pairwise rubric compression. Acc is over all pairs; $\text{Acc}_{\neq}$ excludes equal-gold pairs. Q_n denotes QWK at label budget n.

Config	Tokens/pair	Relative cost
All+PRC	1,899	1.00×
All+Original	3,942	2.08×
Indep.+PRC	6,628	3.49×
Indep.+Original	8,671	4.57×

Table 4: Average tokens per pair and relative cost per configuration, based on the Gemma 4 tokenizer.

Table 3 shows that, within each query strategy, PRC-compressed rubrics and the original rubrics yielded nearly identical pairwise accuracy and downstream QWK. This indicates that PRC preserves the information needed for pairwise comparison while improving token efficiency, making it preferable under both **All** and **Indep.** querying.

Table 4 confirms this efficiency gain. PRC reduced the token count from 3,942 to 1,899 tokens per pair in the **All** setting, and from 8,671 to 6,628 tokens per pair in the **Indep.** setting. Thus, PRC consistently lowered the token budget without a clear loss in performance.

Across query strategies, **Indep.** tended to perform slightly better than **All**, but the gains were small relative to the added cost. We therefore

LLM	Output Layer	Q_{30}	Q_{200}
Gemma 4	Linear	0.637	0.686
	Bifactor	**0.650**	**0.688**
Qwen 3.5	Linear	0.624	0.678
	Bifactor	**0.636**	**0.679**
GPT-5.4m	Linear	0.613	0.668
	Bifactor	**0.625**	**0.670**

Table 5: Effect of the bifactor-inspired output head.

	Gemma 4		Qwen 3.5		GPT-5.4m	
Config	Q_{30}	Q_{200}	Q_{30}	Q_{200}	Q_{30}	Q_{200}
Full	**.650**	**.688**	**.636**	**.679**	**.625**	**.670**
w/o $\mathcal{L}_{\text{pref}}$	.390	.659	.390	.659	.390	.659
w/o $\mathcal{L}_{\text{reg}}$	.647	.668	.631	.654	.622	.644
w/o $\mathcal{L}_{\text{dist}}$	.641	.679	.635	.671	.621	.660
$\mathcal{L}_{\text{reg}}$ only	.357	.629	.357	.629	.357	.629

Table 6: Loss function ablation across three LLMs.

set **All+PRC** as the default, which requires only $1/4.57$ of the tokens of **Indep.+Original** while maintaining comparable QWK. With a larger computational or API budget, **Indep.+PRC** is a viable alternative; in either case, PRC remains effective.

5.2 Bifactor-Inspired Output Head

We compared the bifactor output layer (Eq. 4) against a **standard linear** output layer $\hat{\boldsymbol{y}}_i = \sigma(\boldsymbol{W}_{\text{out}}\boldsymbol{h}_i + \boldsymbol{b}_{\text{out}})$ with $\boldsymbol{W}_{\text{out}} \in \mathbb{R}^{K\times d}$. Table 5 shows that the bifactor architecture consistently outperformed the linear layer across all three LLMs, with the largest gains at $|\mathcal{A}| = 30$ where LLM preferences dominate the training signal and anchors provide little noise correction.

Although the two heads can represent the same input–output mappings, the bifactor parameterization may bias learning toward variation shared across traits, improving robustness to noisy LLM preferences. As $|\mathcal{A}|$ grew, clean anchor supervision increasingly offset preference noise, allowing the linear layer to become competitive.

5.3 Loss Components

Table 6 ablates the three loss terms. The $\mathcal{L}_{\text{reg}}$-only configuration corresponds to a standard supervised multi-output regressor trained solely on the anchor set. Configurations without $\mathcal{L}_{\text{pref}}$ are independent of the LLM and therefore yield identical values across the three LLMs.

The full objective performed best at both budgets for all three LLMs, confirming that each loss contributes to performance. Removing $\mathcal{L}_{\text{pref}}$ caused the largest drop, particularly at $|\mathcal{A}| = 30$, demonstrating the importance of LLM preferences under weak supervision. Removing $\mathcal{L}_{\text{dist}}$ produced a smaller but consistent degradation. Its contribution was generally more pronounced at $|\mathcal{A}| = 200$, as a larger anchor set provides a more accurate empirical target distribution.

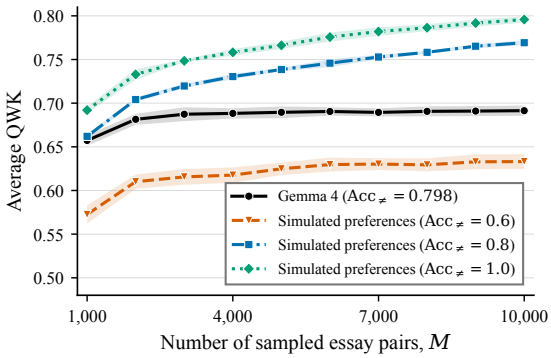


Figure 4: Average QWK across all traits and eight prompts versus the number of sampled pairs M for Gemma 4 and synthetic preferences at three accuracy levels, with 95% confidence intervals across five seeds.

6 Analysis

6.1 Factors Affecting Model Performance

Number of Sampled Essay Pairs. We examined how the number of sampled essay pairs, M, affects MTPAS performance. Although the default setting uses $M = 5{,}000$, we varied M from 1,000 to 10,000 in increments of 1,000, retaining all other default hyperparameters. We used MTPAS with Gemma 4 as the pairwise judge and $|\mathcal{A}| = 200$, the best-performing configuration in Table 2.

As shown in Figure 4, the average QWK across all traits and prompts plateaued at approximately 4,000 pairs. This result indicates that the default $M = 5{,}000$ is sufficient for this configuration.

Pairwise Preference Accuracy. We next examined how the accuracy of LLM-generated pairwise preference labels affects performance. To control this factor, we generated synthetic preferences at varying accuracy levels, measured by $\text{Acc}_{\neq}$ (Section 5.1). Given the true essay scores, we first used a Bradley–Terry model to compute the probability that essay i outperforms essay j on trait k:

$$P(i \succ_k j) = \sigma\left((y_{ik} - y_{jk})/\tau_{\text{sim}}\right). \quad (10)$$

We then sampled a synthetic preference from a Bernoulli distribution parameterized by this proba-

Aggregation	Calibration	Q_{30}	Q_{200}
Bradley–Terry	Linear	0.491	0.514
Bradley–Terry	Isotonic	0.495	0.520
RankNet	Linear	0.588	0.604
RankNet	Isotonic	0.582	0.610
MTPAS	–	**0.650**	**0.688**

Table 7: Average QWK for two-stage pipelines and MTPAS at two anchor budgets, using Gemma 4 preferences.

LLM	Few-shot	MTPAS
Gemma 4	0.436	**0.650**
Qwen 3.5	0.458	**0.636**
GPT-5.4m	0.496	**0.625**

Table 8: Average QWK at $|\mathcal{A}| = 30$: few-shot LLM scoring versus MTPAS, across three LLMs.

bility:

$$\tilde{c}_{ijk} \sim \text{Bernoulli}\left(P(i \succ_k j)\right), \qquad (11)$$

where $\tilde{c}_{ijk} \in \{0, 1\}$ denotes a synthetic preference label. We omitted the tie label $\tilde{c}_{ijk} = 0.5$ because generating many ties from the true scores would indirectly reveal those scores. The temperature τ_{sim} controls preference accuracy. As τ_{sim} increases, the scaled score differences approach zero and the win probabilities approach 0.5, producing increasingly random outcomes. Conversely, a smaller τ_{sim} yields more deterministic outcomes that reflect the true relative scores. We selected τ_{sim} so that the expected $\text{Acc}_{\neq}$ matched each target; Appendix C describes the procedure.

We evaluated $\text{Acc}_{\neq} \in \{0.6, 0.8, 1.0\}$ while varying M as in the preceding analysis. As shown in Table 3, $\text{Acc}_{\neq} = 0.8$ approximates the accuracy of Gemma 4, whereas 0.6 represents lower-quality labels and 1.0 represents perfectly accurate labels. The targets 0.6 and 0.8 correspond to $\tau_{\text{sim}} = 0.76$ and 0.19, respectively, while 1.0 is obtained as $\tau_{\text{sim}} \to 0$.

Figure 4 shows that QWK increased with M under every synthetic-label setting and was higher for more accurate labels. Although the $\text{Acc}_{\neq}$ of Gemma 4 was close to 0.8, its performance plateaued at approximately 4,000 pairs, whereas performance with the synthetic labels continued to improve. This difference may arise because synthetic-label errors are random and thus diminish as M increases, whereas Gemma 4 may make systematic errors that persist with additional pairs. Such errors may reflect LLM biases, including verbosity bias, halo effects, and the overrating of particular essays. Reducing systematic errors in pairwise judgments could therefore further improve MTPAS. Indeed, with sufficiently accurate synthetic labels, MTPAS exceeded the QWK of 0.711 reported by Zhao et al. (2026), suggesting a promising direction for future work.

6.2 Comparison with Simple Alternatives

Two-Stage Pipeline. We evaluated a pipeline that first aggregates LLM pairwise preferences into latent trait scores and then calibrates them to the rubric scale using anchors. We used two representative aggregation methods, Bradley–Terry and RankNet (Burges et al., 2005), followed by either an affine linear map or isotonic regression fit on the anchors. Both stages were performed separately for each trait. Table 7 reports average QWK across all traits and prompts. At both budgets, even the best pipeline underperformed MTPAS. This result highlights the benefit of MTPAS's end-to-end learning, which jointly integrates preference learning with anchor- and distribution-based calibration.

Few-Shot Direct Scoring. A natural alternative under the $|\mathcal{A}| = 30$ regime is to supply the 30 anchor essays to the LLM as in-context demonstrations and elicit absolute trait scores independently per trait. We compared MTPAS against this simple prompting-only baseline, with no calibration or other post-processing. Table 8 reports average QWK across all traits and prompts. Few-shot direct scoring lagged behind MTPAS substantially on every LLM, with gaps of 0.214 (Gemma 4), 0.178 (Qwen 3.5), and 0.129 (GPT-5.4m). Given the *same* 30 anchor essays, routing the LLM's judgments through pairwise preferences and the scoring model $f_{\boldsymbol{\Theta}}$ is markedly more effective than asking the LLM for absolute scores directly.

7 Conclusion

We proposed MTPAS, a semi-supervised framework for multi-trait AES that learns from LLM-generated pairwise preferences and calibrates predictions to the rubric scale using a small set of human-scored anchors. On ASAP/ASAP++, MTPAS achieved 91.4% of the fully supervised SoTA ceiling with only 30 anchors per prompt, and at 200 anchors, the remaining gap was not statistically significant under a paired t-test at the 5% level. MTPAS also outperformed conventional multi-trait AES baselines in average QWK across all matched-anchor-size comparisons, demonstrating its potential to reduce the annotation burden.

Limitations

Our study has several limitations. First, MTPAS is evaluated only on the ASAP/ASAP++ datasets, which consist of English essays from a specific educational context. Generalization to other languages, genres, and assessment contexts remains to be validated. Second, our comparison against published Full-Sup. numbers follows a different evaluation protocol (6:2:2 split) from our setup, in which MTPAS is evaluated on the non-anchor essays; we therefore treat the Full-Sup. figures as an approximate ceiling estimate against which to measure progress rather than as a strict head-to-head comparison, and the matched-anchor-size comparisons against re-trained ArTS, RMTS, and GAT-AES constitute the directly comparable baseline. Third, the PRC prompt template is manually designed, although the rubric compression itself is performed by an LLM; further automation of the prompt design could improve scalability. Fourth, our bifactor-inspired head imposes no explicit psychometric constraints, such as regularizing trait-specific residuals or enforcing orthogonality between general and trait-specific factors. It can therefore represent the same input–output mappings as a standard linear head. Future work could incorporate these constraints and explore multiple group factors for assessments with many heterogeneous traits. Fifth, although PRC reduces the per-pair query cost, the number of candidate essay pairs grows quadratically with the corpus size, and the total LLM inference cost can still become substantial at scale. Finally, the quality of pairwise preferences depends on the LLM used, and we have not exhaustively explored the impact of LLM size, architecture, or instruction tuning methodology on preference quality.

Ethics Statement

We use publicly available, de-identified datasets (ASAP and ASAP++) that are standard benchmarks in AES research. Personally identifiable information in the essays was replaced by the dataset providers with placeholders such as PERSON, LOCATION, and DATE, and our LLM prompts (Appendix A) explicitly instruct the model not to penalize these placeholders. We did not collect new annotations and relied on the human trait scores already released with these benchmarks.

All datasets and models used in this work were accessed under their respective public terms: ASAP via the Kaggle ASAP-AES competition, ASAP++ as released by Mathias and Bhattacharyya (2018), Gemma 4 and Qwen 3.5 via their official Hugging Face releases under their respective licenses, and GPT-5.4, GPT-5.4-mini, and text-embedding-3-large through the OpenAI API. Our use of these artifacts in a research context is consistent with their intended use.

All MTPAS training and inference were run on a single Apple M1 Max workstation (64 GB unified memory), with Gemma 4 and Qwen 3.5 served locally via vLLM and the OpenAI models accessed through the API. Gemma 4 E4B has 4.5B active parameters, Qwen 3.5 has 9B parameters, and the OpenAI models are proprietary. For Gemma 4, pairwise preference collection over 5,000 pairs $\times$ 8 prompts consumed approximately 76M input tokens and 13.4M output tokens. We implemented MTPAS in PyTorch, computed the Sinkhorn divergence with `geomloss`, and computed QWK with scikit-learn's `cohen_kappa_score`.

Our work aims to reduce the cost of human annotation, but we emphasize that MTPAS is intended to assist human graders, not replace them. Automated scores should always be subject to human oversight, particularly in high-stakes assessment contexts. We acknowledge that LLM preferences may encode biases present in their training data, and recommend careful bias auditing before deployment.

Acknowledgments

We thank Deloitte Touche Tohmatsu LLC for its financial support of this research.

We used ChatGPT to assist with proofreading and editing the manuscript, and to assist with debugging the experimental code. All scientific content, experimental design, analysis, and final wording are the authors' own.

References

Dimitrios Alikaniotis, Helen Yannakoudakis, and Marek Rei. 2016. Automatic text scoring using neural networks. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pages 715–725.

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

Martin Brunner, Gabriel Nagy, and Oliver Wilhelm.

2012. A tutorial on hierarchically structured constructs. *Journal of Personality*, 80(4):796–846.

Chris 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*, pages 89–96.

Yuan Chen and Xia Li. 2024. PLAES: Prompt-generalized and level-aware learning framework for cross-prompt automated essay scoring. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation*, pages 12775–12786.

SeongYeub Chu, Jong Woo Kim, Bryan Wong, and Mun Yong Yi. 2025. Rationale behind essay scores: Enhancing S-LLM's multi-trait essay scoring with rationale generated by LLMs. In *Findings of the Association for Computational Linguistics, 2025 Annual Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics*, pages 5811–5829.

Jacob Cohen. 1968. Weighted kappa: Nominal scale agreement with provision for scaled disagreement or partial credit. *Psychological Bulletin*, 70(4):213–220.

Marco Cuturi. 2013. Sinkhorn Distances: Lightspeed Computation of Optimal Transport. In *Advances in Neural Information Processing Systems*, volume 26, pages 2292–2300.

Marco Cuturi and Gabriel Peyré. 2016. A smoothed dual approach for variational wasserstein problems. *SIAM Journal on Imaging Sciences*, 9(1):320–343.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics*, pages 4171–4186.

Heejin Do, Yunsu Kim, and Gary Lee. 2024a. Autoregressive score generation for multi-trait essay scoring. In *Findings of the Association for Computational Linguistics, the 18th Conference of the European Chapter of the Association for Computational Linguistics*, pages 1659–1666.

Heejin Do, Sangwon Ryu, and Gary Lee. 2024b. Autoregressive multi-trait essay scoring via reinforcement learning with scoring-aware multiple rewards. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 16427–16438.

Fei Dong, Yue Zhang, and Jie Yang. 2017. Attention-based recurrent convolutional neural network for automatic essay scoring. In *Proceedings of the 21st Conference on Computational Natural Language Learning*, pages 153–162.

Jean Feydy, Thibault Séjourné, François-Xavier Vialard, Shun-ichi Amari, Alain Trouve, and Gabriel Peyré. 2019. Interpolating between optimal transport and mmd using sinkhorn divergences. In *Proceedings of the 22nd International Conference on Artificial Intelligence and Statistics*, volume 89, pages 2681–2690.

Karl J. Holzinger and Frances Swineford. 1937. The bi-factor method. *Psychometrika*, 2:41–54.

Mohamed A. Hussein, Hesham A. Hassan, and Mohammad Nassef. 2020. A trait-based deep learning automated essay scoring system with adaptive feedback. *International Journal of Advanced Computer Science and Applications*, 11(5).

Harrison Lee, Samrat Phatale, Hassan Mansoor, Thomas Mesnard, Johan Ferret, Kellie Ren 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 *Proceedings of the 41st International Conference on Machine Learning*, pages 26874–26901.

Shengjie Li and Vincent Ng. 2025. Graph-based multi-trait essay scoring. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 33325–33351.

Xia Li and Wenjing Pan. 2025. KAES: Multi-aspect shared knowledge finding and aligning for cross-prompt automated scoring of essay traits. In *Proceedings of the 39th Association for the Advancement of Artificial Intelligence Conference on Artificial Intelligence*, volume 39, pages 24476–24484.

Adian Liusie, Vatsal Raina, Yassir Fathullah, and Mark Gales. 2024. Efficient LLM comparative assessment: A product of experts framework for pairwise comparisons. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 6835–6855.

Ilya Loshchilov and Frank Hutter. 2017. Fixing weight decay regularization in Adam. *ArXiv*, abs/1711.05101.

Watheq Ahmad Mansour, Salam Albatarni, Sohaila Eltanbouly, and Tamer Elsayed. 2024. Can large language models automatically score proficiency of written essays? In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation*, pages 2777–2786.

Sandeep Mathias and Pushpak Bhattacharyya. 2018. ASAP++: Enriching the ASAP automated essay grading dataset with essay attribute scores. In *Proceedings of the 11th International Conference on Language Resources and Evaluation*.

Sandeep Mathias and Pushpak Bhattacharyya. 2020. Can neural networks automatically score essay traits? In *Proceedings of the 15th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 85–91.

Seo Yeon Park and Cornelia Caragea. 2024. VerifyMatch: A semi-supervised learning paradigm for natural language inference with confidence-aware MixUp. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 19319–19335.

Qwen Team. 2026. Qwen3.5: Towards native multimodal agents.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Steven P. Reise. 2012. The rediscovery of bifactor measurement models. *Multivariate Behavioral Research*, 47(5):667–696.

Robert Ridley, Liang He, Xin-yu Dai, Shujian Huang, and Jiajun Chen. 2021. Automated cross-prompt scoring of essay traits. In *Proceedings of the Thirty-Fifth Association for the Advancement of Artificial Intelligence Conference on Artificial Intelligence*, volume 35, pages 13745–13753.

Robert Ridley, Liang He, Xinyu Dai, Shujian Huang, and Jiajun Chen. 2020. Prompt agnostic essay scorer: A domain generalization approach to cross-prompt automated essay scoring. *Preprint*, arXiv:2008.01441.

Takumi Shibata and Yuichi Miyamura. 2025. LCES: Zero-shot automated essay scoring via pairwise comparisons using large language models. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 29988–30001.

Takumi Shibata and Masaki Uto. 2022. Analytic automated essay scoring based on deep neural networks integrating multidimensional item response theory. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 2917–2926.

Kaveh Taghipour and Hwee Tou Ng. 2016. A neural approach to automated essay scoring. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1882–1891.

Masaki Uto. 2021. A review of deep-neural automated essay scoring models. *Behaviormetrika*, 48(2):459–484.

Jiong Wang and Jie Liu. 2025. T-MES: Trait-aware mix-of-experts representation learning for multi-trait essay scoring. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 1224–1236.

Jiayi Xie, Kaiwei Cai, Li Kong, Junsheng Zhou, and Weiguang Qu. 2022. Automated essay scoring via pairwise contrastive regression. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 2724–2733.

Kevin P. Yancey, Geoffrey Laflair, Anthony Verardi, and Jill Burstein. 2023. Rating short L2 essays on the CEFR scale with GPT-4. In *Proceedings of the 18th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 576–584.

Ruosong Yang, Jiannong Cao, Zhiyuan Wen, Youzheng Wu, and Xiaodong He. 2020. Enhancing automated essay scoring performance via fine-tuning pre-trained language models with combination of regression and ranking. In *Findings of the Association for Computational Linguistics, the 2020 Conference on Empirical Methods in Natural Language Processing*, pages 1560–1569.

Lui Yoshida. 2024. The impact of example selection in few-shot prompting on automated essay scoring using GPT models. In *Artificial Intelligence in Education. Posters and Late Breaking Results, Workshops and Tutorials, Industry and Innovation Tracks, Practitioners, Doctoral Consortium and Blue Sky*, pages 61–73.

Lui Yoshida. 2025. Are the reasoning models good at automated essay scoring? In *Findings of the Association for Computational Linguistics, the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 8388–8394.

Ruoyu Zhang, Yanzeng Li, Yongliang Ma, Ming Zhou, and Lei Zou. 2023. LLMaAA: Making large language models as active annotators. In *Findings of the Association for Computational Linguistics, the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 13088–13103.

Jiachen Zhao, Wenlong Zhao, Andrew Drozdov, Benjamin Rozonoyer, Md Arafat Sultan, Jay-Yoon Lee, Mohit Iyyer, and Andrew McCallum. 2024. Multistage collaborative knowledge distillation from a large language model for semi-supervised sequence generation. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*, pages 14201–14214.

Shiman Zhao, Siyuan Liu, and Zhiqi Shen. 2026. Multi-knowledge enhanced graph neural network for multi-trait essay scoring. *Proceedings of the AAAI Conference on Artificial Intelligence*, 40(34):28733–28741.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, Hao Zhang, Joseph Gonzalez, and Ion Stoica. 2023. Judging LLM-as-a-judge with MT-bench and chatbot arena. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, volume 36, pages 46595–46623.

A LLM Prompts

This appendix lists the prompts used by the LLM components of MTPAS: (i) the pairwise essay comparison prompt in the *All*-trait mode (Section 3.2),

which compares all rubric traits in a single LLM call; (ii) the pairwise essay comparison prompt in the per-trait *Independent* mode used as an ablation in Section 5.1; and (iii) the pairwise rubric compression (PRC) prompt of Section 3.2. Each of (i) and (ii) is split into a developer (system) message and a user message in the OpenAI chat format. The placeholders `{prompt}`, `{rubric}`, `{essay1}`, and `{essay2}` are filled in at query time, and `[TRAIT_NAME]` and `[PASTE RUBRIC]` are the inputs supplied to PRC.

A.1 Pairwise Comparison Prompt: All-Trait Mode

In the All-trait mode, the rubric placeholder is filled with the concatenation of every trait's PRC-compressed rubric, and the LLM returns one preference judgment per trait in a single call.

Developer message.

```
You are an English writing evaluator for
middle school exam essays.

Your task is to compare Essay 1 and Essay 2
on multiple rubric traits.
For each trait, decide which essay is
better: "essay1", "essay2", or "tie".

Evaluation rules:
- Judge each trait independently.
- Treat "overall" as one of the rubric
  traits if it appears in the rubric, and
  evaluate it independently according to the
  same rules.
- Do not let strength or weakness in one
  trait affect another trait.
- Use only evidence relevant to the current
  trait.
- Do not decide an overall winner first.
- If evidence is mixed and neither essay is
  clearly better for a trait, output "tie".
- Do not penalize anonymization placeholders
  such as PERSON, ORGANIZATION, LOCATION,
  DATE, TIME, MONEY, PERCENT, CAPS, and NUM.

Before writing the final answer, follow this
procedure internally:
1. Identify all rubric traits.
2. Compare the two essays for one trait at a
   time using only evidence relevant to that
   trait.
3. Record the decision for that trait before
   moving to the next trait.
4. Do not revise earlier trait decisions
   just to make the full set of judgments
   more consistent.
5. Output the final JSON only.

Output rules:
- Return valid JSON only.
- Output exactly one judgment for every
  trait in the rubric.
- Each preference must be exactly one of:
  "essay1", "essay2", or "tie".
- Keep each reasoning short and
  trait-specific.

Output schema:
{
  "trait_judgments": [
    {
      "trait": "TRAIT_NAME",
      "reasoning":
        "One short trait-specific sentence.",
      "preference": "essay1"
    }
  ]
}
```

User message.

```
Prompt:
{prompt}

Rubric:
{rubric}

Essay 1:
{essay1}

Essay 2:
{essay2}
```

A.2 Pairwise Comparison Prompt: Independent Mode

In the Independent mode, the LLM is queried separately for each trait, so the rubric placeholder is filled with a single trait's rubric (compressed or original) and the LLM returns one preference per call. The developer and user messages adapted from Shibata and Miyamura (2025) are shown below.

Developer message.

```
As an English teacher, your primary
responsibility is to evaluate the writing
quality of essays written by middle school
students on an English exam. During the
assessment process, you will be provided
with a prompt and an essay. First, you
should provide comprehensive and concrete
feedback that is closely linked to the
content of the essay. It is essential to
avoid offering generic remarks that could be
applied to any piece of writing. To create a
compelling evaluation for both the student
and fellow experts, you should reference
specific content of the essay to
substantiate your assessment.

Your task is to determine which essay, Essay
1 or Essay 2, receives a higher score based
on the provided rubric guidelines. If both
essays receive the same score, please
respond with "tie".
```

User message.

```
# Prompt
{prompt}

# Rubric Guidelines
```

````
{rubric}

# Note
I have made an effort to remove personally
identifying information from the essays
using the Named Entity Recognizer (NER). The
relevant entities are identified in the text
and then replaced with a string such as
"PERSON", "ORGANIZATION", "LOCATION",
"DATE", "TIME", "MONEY", "PERCENT", "CAPS"
(any capitalized word) and "NUM" (any
digits). Please do not penalize the essay
because of the anonymizations.

# Essay1
{essay1}

# Essay2
{essay2}

Provide your reasoning and final decision in
json format:
```json
{
    "reasoning":
      "Your reasoning in one sentence here.",
    "preference":
      "essay1" or "essay2" or "tie"
}
```
````

A.3 Pairwise Rubric Compression Prompt

PRC rewrites a verbose Likert-style rubric for one trait into a compact pairwise comparison rubric, while explicitly forbidding the LLM from adding new criteria, examples, or quality dimensions beyond those in the source. The full prompt is given below; the resulting compressed rubrics are concatenated across traits and supplied as the `{rubric}` placeholder of the All-trait pairwise comparison prompt.

```
You are a rubric compressor, not a rubric
improver.

Goal:
Convert a Likert-style rubric for one trait
into a compact pairwise rubric for A-vs-B
essay comparison.

Compression principle:
The goal is to remove ordinal redundancy
that is unnecessary for pairwise judgment.
Do NOT enrich, generalize, or improve the
rubric.
Do NOT add new evaluation criteria,
subcriteria, assumptions, or examples.
Do NOT make the rubric more "complete" than
the original.
If the original rubric is already short,
keep the output minimal and close to the
source.

Allowed operations only:
1. Merge repeated score-level wording.
2. Rewrite score-level descriptions into
   pairwise win conditions.
3. Shorten wording while preserving original
   meaning.
4. Convert explicit score differences into a
   tie rule only when directly supported by
   the original rubric.

Strict constraints:
- Preserve information; do not add
  information.
- Stay lexically close to the original
  whenever possible.
- Use only content explicitly present in the
  source rubric.
- Do not infer hidden requirements.
- Do not introduce quality dimensions not
  named in the source.
- Do not add explanatory commentary.
- Do not add pitfall text unless a pitfall
  is directly recoverable from the source.
- If a field cannot be filled without adding
  information, use an empty string or empty
  list.
- If the rubric is already concise, output a
  near-identity transformation.

Token minimization rule:
Among all valid outputs, prefer the shortest
one that preserves the original meaning for
pairwise evaluation.

Field-specific rules:
- trait_name: copy exactly the provided
  trait name.
- focus: one short phrase, copied or
  minimally rewritten from the source.
- win_conditions: 1-3 short rules only; each
  must be directly traceable to the source.
- tie_rule: only describe "similar
  performance" using source-supported
  wording; do not introduce thresholds like
  "minor differences" unless stated in the
  source.
- evidence_cues: short noun phrases taken
  from explicit rubric language only.
- pitfall: usually empty; fill only if
  explicitly supported by the source rubric.

Input trait name:
[TRAIT_NAME]

Input rubric:
[PASTE RUBRIC]

Output JSON schema:
{
  "trait_name": "...",
  "focus": "...",
  "win_conditions": ["...", "..."],
  "tie_rule": "...",
  "evidence_cues": ["...", "..."],
  "pitfall": "..."
}
```

B Dataset Details

Table 9 provides details of the ASAP and ASAP++ datasets.

Prompt	# Essays	Avg. Words	Overall Range	Trait Range	Traits
P1	1,783	350	2–12	1–6	Content, Organization, Word Choice, Sentence Fluency, Conventions
P2	1,800	350	1–6	1–6	Content, Organization, Word Choice, Sentence Fluency, Conventions
P3	1,726	150	0–3	0–3	Content, Prompt Adherence, Language, Narrativity
P4	1,772	150	0–3	0–3	Content, Prompt Adherence, Language, Narrativity
P5	1,805	150	0–4	0–4	Content, Prompt Adherence, Language, Narrativity
P6	1,800	150	0–4	0–4	Content, Prompt Adherence, Language, Narrativity
P7	1,569	250	0–30	0–6	Content, Organization, Conventions, Style
P8	723	650	0–60	2–12	Content, Organization, Word Choice, Sent. Fluency, Conventions, Voice

Table 9: ASAP and ASAP++ dataset statistics. The Overall Range column shows the rubric-defined range of the holistic score, and the Trait Range column shows the (common) range of the per-trait scores listed in the rightmost column.

Category	Hyperparameter	Value
Input representation	Embedding model	`text-embedding-3-large`
	Embedding dimensionality d_{in}	3,072
Shared encoder	Hidden dimension d	16
	Activation	$\tanh$
	Dropout probability	0.5
Preference data	Pairwise preference pairs per prompt M	5,000
Optimization	Optimizer	AdamW
	Learning rate	5×10^{-4}
	Weight decay	10^{-3}
	Epochs	15
Loss	Anchor loss weight α	4.0
	Distribution loss weight β	2.5
	Bradley–Terry temperature τ	0.2
	Sinkhorn parameters	$p = 2$, blur $\varepsilon = 0.05$, scaling 0.9

Table 10: MTPAS hyperparameters.

C Controlling Synthetic Preference Accuracy

In Section 6.1, we generated synthetic preference labels using a Bradley–Terry model. This appendix describes how we selected τ_{sim} to match a target value for the expected $\text{Acc}_{\neq}$, the accuracy on essay pairs with unequal gold scores.

For a pair of essays x_i and x_j with $y_{ik} \neq y_{jk}$, let $g_{ijk} = \mathbf{1}[y_{ik} - y_{jk} > 0]$ denote the gold preference direction. The probability that the synthetic label agrees with this direction is

$$P(\tilde{c}_{ijk} = g_{ijk}) = \sigma\left(\frac{|y_{ik} - y_{jk}|}{\tau_{\text{sim}}}\right). \tag{12}$$

Let $\mathcal{P}_{\text{sub}}^{(p)}$ denote the set of essay pairs sampled for prompt p, and let $\mathcal{N}_{pk} = \{(i,j) \in \mathcal{P}_{\text{sub}}^{(p)} \mid y_{ik} \neq y_{jk}\}$ denote the subset with unequal gold scores for trait k. Further, let $\mathcal{Q}_k$ denote the set of prompts for which trait k is defined. The expected $\text{Acc}_{\neq}$ is then

$$\bar{\text{Acc}}_{\neq}(\tau_{\text{sim}}) = \frac{1}{|\mathcal{K}|}\sum_{k\in\mathcal{K}}\frac{1}{|\mathcal{Q}_k|}\sum_{p\in\mathcal{Q}_k} \times \frac{1}{|\mathcal{N}_{pk}|}\sum_{(i,j)\in\mathcal{N}_{pk}}\sigma\left(\frac{|y_{ik} - y_{jk}|}{\tau_{\text{sim}}}\right). \tag{13}$$

For a target accuracy a^*, we used the bisection method to solve $\bar{\text{Acc}}_{\neq}(\tau_{\text{sim}}) = a^*$ for τ_{sim}. For target accuracies of 0.60 and 0.80, the realized $\text{Acc}_{\neq}$ values of the sampled synthetic labels were 0.601 and 0.799, respectively.

D Hyperparameters

Table 10 summarizes the hyperparameters used for MTPAS. Unless explicitly overridden for the label-budget experiments, these values correspond to the default bifactor configuration used in our implementation.

E Algorithm

Algorithm 1 summarizes the full MTPAS training procedure.

Method	Data	P1	P2	P3	P4	P5	P6	P7	P8	Avg. (SD)
ArTS (Do et al., 2024a)	Full-Sup.	0.708	0.706	0.704	0.767	0.723	**0.776**	0.749	0.603	0.717 (±0.025)
T-MES (Wang and Liu, 2025)	Full-Sup.	0.728	0.684	0.702	0.771	0.726	0.754	**0.755**	0.629	0.719 (–)
SaMRL (Do et al., 2024b)	Full-Sup.	0.702	**0.711**	0.708	0.766	0.722	0.773	0.743	0.649	0.722 (±0.012)
RMTS (Chu et al., 2025)	Full-Sup.	0.716	0.704	**0.723**	0.772	**0.737**	0.769	0.736	0.651	0.726 (±0.042)
GAT-AES (Li and Ng, 2025)	Full-Sup.	0.722	0.698	0.707	0.771	0.720	0.767	0.744	**0.680**	0.726 (±0.009)
Zhao et al. (2026)	Full-Sup.	**0.732**	0.704	0.719	**0.782**	0.734	0.771	**0.755**	0.636	**0.729** (–)
ArTS	$\|\mathcal{A}\|$=200	0.589	0.629	0.628	0.674	0.644	0.600	0.533	0.488	0.598 (±0.033)
RMTS	$\|\mathcal{A}\|$=200	0.590	0.557	0.661	0.696	0.684	0.637	0.541	0.432	0.600 (±0.020)
GAT-AES	$\|\mathcal{A}\|$=200	0.640	0.610	0.639	0.696	0.657	0.667	0.628	0.558	0.637 (±0.004)
MTPAS (Gemma 4)	$\|\mathcal{A}\|$=200	0.694	0.695	0.702	0.758	0.702	0.730	0.661	**0.680**	0.703 (±0.003)
MTPAS (Qwen 3.5)	$\|\mathcal{A}\|$=200	0.682	0.708	0.699	0.762	0.703	0.730	0.647	0.639	0.696 (±0.004)
MTPAS (GPT-5.4m)	$\|\mathcal{A}\|$=200	0.658	0.696	0.692	0.754	0.690	0.731	0.641	0.646	0.688 (±0.003)

Table 11: Per-prompt average QWK at $|\mathcal{A}| = 200$ anchor essays. Each prompt column reports the QWK averaged across all traits associated with that prompt. A paired t-test over the eight prompts between MTPAS (Gemma 4) and the strongest full-data baseline, Zhao et al. (2026), yields $p = 0.093$, which is not significant at the 5% level.

F Detailed Comparison at $|\mathcal{A}| = 200$

Table 11 complements the per-trait comparison in Table 2 by reporting per-prompt average QWK at our representative operating point of $|\mathcal{A}| = 200$ anchor essays. We use the same set of baselines as Table 2: six fully supervised methods trained on all data, and the three multi-trait methods with publicly available training code re-trained at $|\mathcal{A}| = 200$.

Algorithm 1 MTPAS Training Procedure

Require: Essay set $\mathcal{D}$, anchor set $\mathcal{D}_{\text{anc}}$, LLM $\mathcal{M}$, hyperparameters $\alpha, \beta, \tau, \varepsilon, M$

// Phase 1: Preference Collection
Sample M essay pairs $\mathcal{P}_{\text{sub}}$ from $\mathcal{D}$
Apply pairwise rubric compression to the rubric
for each pair $(i, j) \in \mathcal{P}_{\text{sub}}$ **do**
  Query LLM $\mathcal{M}$ for all-trait comparison $\{c_{ijk}\}_{k \in \mathcal{K}}$
end for
// Phase 2: Model Training (5-Fold Ensemble)
Split anchors $\mathcal{A}$ into 5 folds
for each fold $f = 1, \ldots, 5$ **do**
  Set training anchors $\mathcal{A}_{\text{train}}^{(f)}$, validation anchors $\mathcal{A}_{\text{val}}^{(f)}$
  Initialize bifactor model $f_{\boldsymbol{\Theta}}$
  for each epoch **do**
    for each batch of pairs **do**
      Compute $\mathcal{L}_{\text{pref}}$ (Eq. 7) over batch pairs
      Compute $\mathcal{L}_{\text{reg}}$ (Eq. 8) on training anchors in batch
      Compute $\mathcal{L}_{\text{dist}}$ (Eq. 9) between predictions and training anchor distribution
      Update $\boldsymbol{\Theta}$ by minimizing $\mathcal{L}_{\text{pref}} + \alpha\mathcal{L}_{\text{reg}} + \beta\mathcal{L}_{\text{dist}}$
    end for
    Evaluate on $\mathcal{A}_{\text{val}}^{(f)}$; save the best model
  end for
end for
// Phase 3: Ensemble Prediction
Average predictions from 5 best-fold models
Rescale to original score range and round