MAPLE: A Meta-learning Framework for Cross-Prompt Essay Scoring

Salam Albatarni, May Bashendy, Sohaila Eltanbouly, Tamer Elsayed
Computer Science and Engineering Department, Qatar University
{sa1800633, ma1403845, se1403101, telsayed}@qu.edu.qa

Abstract

Automated Essay Scoring (AES) faces significant challenges in cross-prompt settings, where models must generalize to unseen writing prompts. To address this limitation, we propose *MAPLE*, a meta-learning framework that leverages prototypical networks to learn transferable representations across different writing prompts. Across three diverse datasets (ELLIPSE and ASAP (English), and LAILA (Arabic)), *MAPLE* achieves state-of-the-art performance on ELLIPSE and LAILA, outperforming strong baselines by 8.5 and 3 points in QWK, respectively. On ASAP, where prompts exhibit heterogeneous score ranges, *MAPLE* yields improvements on several traits, highlighting the strengths of our approach in unified scoring settings. Overall, our results demonstrate the potential of meta-learning for building robust cross-prompt AES systems.

1 Introduction

Automated Essay Scoring (AES) has been an active research area for about six decades (Page, 1966). AES systems assess the quality of essays by providing holistic scores, trait-specific scores, or both, offering scalable alternatives to costly manual grading in large-scale assessments (Burstein, 2013).

Current AES research follows two main paradigms: *prompt-specific* and *cross-prompt*.[1] Prompt-specific AES trains and tests models on essays from the same prompt, achieving strong performance but requiring substantial labeled data (Kumar et al., 2022). In contrast, cross-prompt AES trains a model on a set of source prompts and tests it on *unseen* target prompts (Ridley et al., 2021). This setup is more practical, but struggles with performance due to prompt variability.

Despite ongoing efforts, cross-prompt AES remains under-explored, with generalization still a major challenge. To contribute towards addressing this gap, we introduce *MAPLE*, a **M**et**A**-learning framework for **P**rototypica**L** cross-prompt **E**ssay scoring. Unlike prior optimization-based meta-learning approaches for AES (Chen and Li, 2024; Wang et al., 2025), which treat each prompt as a separate meta-task and thus limit task diversity, *MAPLE* adopts a non-parametric approach based on prototypical networks (Snell et al., 2017) and proposes a more flexible meta-task definition that extends beyond prompts to include traits and score distributions. This design increases task heterogeneity and better leverages meta-learning's strength in generalization, as suggested by prior work on heterogeneous task setups (Iwata and Kumagai, 2020; van der Heijden et al., 2021).

Inspired by the work of Wu et al. (2021), we explore reformulating multiclass scoring as a series of binary classification tasks, which enhances task diversity during meta-training and improves the model's generalizability. Additionally, we introduce a gating mechanism that incorporates contextual information from the prompt, trait-specific rubrics, and engineered features previously shown to enhance cross-prompt performance.

In this work, we test *MAPLE* over essays of two different languages, English and Arabic, to assess its robustness. Experimental results show that *MAPLE* achieves SOTA performance on both ELLIPSE English dataset (Crossley et al., 2023) and LAILA Arabic dataset (Bashendy et al., 2026). On ASAP English dataset (Mathias and Bhattacharyya, 2018), which presents additional challenges due to its varying score ranges across prompts, *MAPLE* achieves improvements on traits with unified score ranges. This indicates that the proposed framework is a promising direction for building robust cross-prompt AES models.

The main contribution of this work is four-fold:

1. We introduce *MAPLE*, a *novel* meta-learning framework for cross-prompt AES.

[1] A prompt is the description of a writing task.

Findings of the Association for Computational Linguistics: ACL 2026, pages 19948–19961

2. We evaluate *MAPLE* on both English and Arabic essays to demonstrate its effectiveness and robustness across languages.

3. *MAPLE* achieves SOTA performance on ELLIPSE and LAILA, and exhibits a competitive trait-level performance on ASAP.

4. We release our implementation to support replication for future AES research.[2]

The remainder of this paper is organized as follows. We review related work in §2, provide meta-learning background in §3, and introduce *MAPLE* in §4. The experimental setup is described in §5, followed by results and discussion in §6. Finally, we conclude and suggest future directions in §7.

2 Related Work

In this section, we review cross-prompt and Arabic AES, along with the main meta-learning frameworks and their adoption in existing AES systems.

2.1 Cross-prompt AES

The limited feedback provided by cross-prompt holistic scoring (Cao et al., 2020; Ridley et al., 2020; Zhang et al., 2025) motivated shifting towards trait scoring (Ridley et al., 2021). Recent advances include ProTACT (Do et al., 2023), which combines prompt-aware essay representations with extracted features using POS-embedding-based neural model. Chen and Li (2023) use contrastive learning to align source-target representations. Li and Ng (2024b) adopt multi-task learning with a purely feature-based approach. Xu et al. (2025) integrates syntactic embeddings with an LLM to measure essays' relevance. Eltanbouly et al. (2025) scores essays using LLM-extracted and hand-crafted features. Chen et al. (2025) proposed MOOSE, a Mixture-of-Experts (MoE) architecture, integrating prompt and essay information using experts for overall and relative quality, and essay-prompt relevance, achieving SOTA performance.

2.2 Arabic AES

Research on Arabic AES has recently gained momentum with the release of new publicly available datasets, including QAES (Bashendy et al., 2024) and TAQEEM 2025 (Bashendy et al., 2025). Sayed et al. (2025) pioneered cross-prompt Arabic AES by introducing a comprehensive feature set and evaluating various AES models. TAQEEM 2025 shared task introduced Arabic cross-prompt holistic and trait scoring tasks, with a multitask AraBERT baseline (Bashendy et al., 2025). The top-performing systems employed GPT-4o with few-shot prompting (Almarwani et al., 2025), and GPT-4.1 with 10-shot chain-of-thought prompting (Alnajjar et al., 2025). Most recently, LAILA dataset (Bashendy et al., 2026) was introduced with 7,859 essays annotated with seven traits and a holistic score, and was benchmarked in a cross-prompt setup using feature-based models, adapted English SOTA baselines, and LLM-based approaches.

2.3 Meta-learning for AES

In the AES context, PLAES (Chen and Li, 2024), a framework for cross-prompt AES, was proposed employing MAML (Finn et al., 2017) to capture general knowledge across prompts, and a level-aware contrastive learning to distinguish the different essays' quality. In addition, Wang et al. (2025) proposed MLCAES, a meta-learning framework for holistic cross-prompt AES that guides model generalization toward target prompt distributions using a distribution-guided meta-learner selection mechanism. Both approaches fall under optimization-based methods, where the model is trained to directly optimize its parameters for rapid adaptation. Zeng et al. (2023) used prototypical networks, a non-parametric method, for cross-prompt *short-answer* scoring. While we also adopt prototypical networks, we address *essay* scoring, which differs from short-answer scoring in both length and trait complexity. Moreover, their approach relies on few-shot labels from the target prompt, whereas we assume *completely unseen target prompts*. In addition, we explore multiple task-construction strategies (§4.1) rather than a single-task setup, jointly predict multiple traits instead of a single holistic score, and retain the original prototypical loss while incorporating hand-crafted features, all of which distinguish our approach.

Despite these advances, existing meta-learning approaches for AES tend to model prompts as homogeneous tasks, conflicting with meta-learning's strength in heterogeneous settings (Iwata and Kumagai, 2020; van der Heijden et al., 2021). This limits tasks to available prompts, reducing generalization. We follow (Wu et al., 2021) by exploring reframing the problem as binary classification to increase task diversity and improve model adaptability and effectiveness.

[2] https://github.com/salbatarni/ACL2026_MAPLE

3 Background: Meta-learning

Meta-Learning is a few-shot learning paradigm that enables rapid adaptation from task-independent to task-specific spaces (Finn et al., 2017), with success across different domains (Triantafillou et al., 2019; Tarunesh et al., 2021; Wu et al., 2021).

Meta-learning consists of *two phases*: *meta-training* and *meta-testing*. During meta-training, a model is exposed to a diverse set of tasks, each with labeled support and query sets, to learn adaptable representations that generalize across tasks. In meta-testing, the model supposedly adapts to a new task, leveraging labeled support set to predict labels for unlabeled query set. In our work, we employ *Prototypical Networks* (Snell et al., 2017), a meta-learning framework that learns a shared embedding space for classification problems, where *class prototypes* are computed based on corresponding labeled support set examples. These prototypes enable classification of unlabeled query samples by proximity in the embedding space.

Meta-training During meta-training, the model is trained on a set of classification tasks $\mathcal{T}_{tr} = \{T_i\}$. Each task T_i consists of a pair of labeled subsets: a *support* set S_i and a *query* set Q_i. The support set contains k shots (examples) from every class $c \in C_i$, where C_i is the class set of T_i. All examples are encoded using an encoder or learner f_θ, which maps an input x into an embedding space $f_\theta(x) \in \mathbb{R}^d$.

Formally, the support set S_i is defined as $\{(x_j^s, y_j^s)\}$, where x_j^s is an example and y_j^s is its corresponding class. Similarly, the query set Q_i is a set of labeled examples $\{(x_j^q, y_j^q)\}$; the model uses the support set to predict the classes of the query set by learning an embedding space where each class c is represented by a prototype $c^\star$, computed as the centroid of embeddings of its k-shot examples:

$$c^\star = \frac{1}{k} \sum_{y_j^s = c} f_\theta(x_j^s) \tag{1}$$

For each query example x_j^q, the learner computes the distance D, using a distance function ϕ, between $f_\theta(x_j^q)$ and the class prototypes:

$$D(x_j^q, c) = \phi(f_\theta(x_j^q), c^\star), c \in C_i \tag{2}$$

The final prediction c_j^q is assigned to the class closest to the query sample:

$$c_j^q = \arg\max_{c \in C_i} D(x_j^q, c) \tag{3}$$

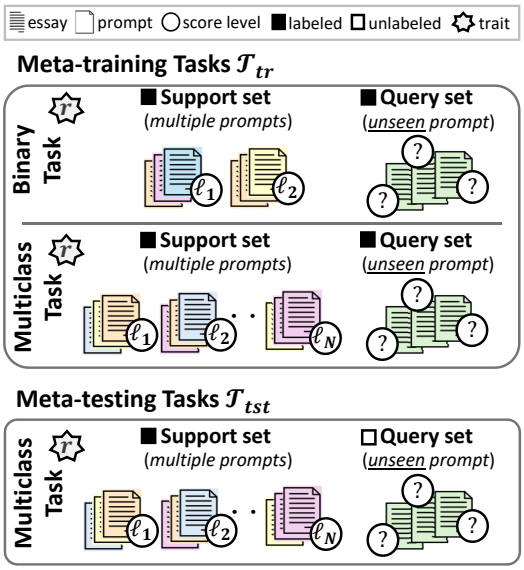


Figure 1: *MAPLE* task generation. In meta-training, we explore two settings, binary and multiclass classification. Each sampled task includes a support set (for C_i computation) and a query set (for evaluation/learner-update). In meta-testing, the task is multiclass, where the support set includes all training data and the query set corresponds to an unseen prompt.

Finally, the model is updated based on its performance on the query set, with the loss function $\mathcal{L}$:

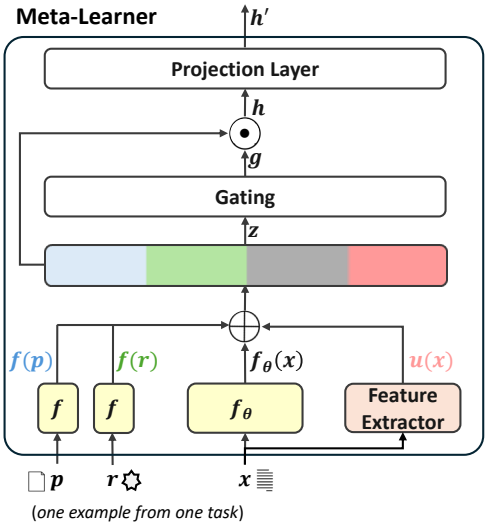
$$\mathcal{L}(x_j^q) = -\log \frac{\exp\left(D(x_j^q, y_j^q)\right)}{\sum_{c \in C_i} \exp\left(D(x_j^q, c)\right)} \tag{4}$$

where y_j^q is the class label of x_j^q.

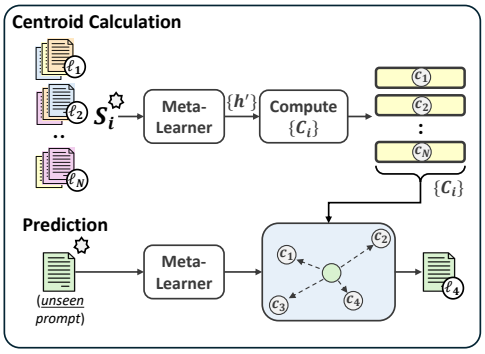
Meta-testing After training the learner f_θ, the model is ready to handle new, unseen tasks $\mathcal{T}_{tst} = \{T_i\}$. Similar to meta-training, each task T_i consists of a support set S_i and a query set Q_i. The key difference, however, is that the query set Q_i in meta-testing is *unlabeled*. The learner f_θ quickly adapts to each new task T_i using the *labeled* support set S_i, then predicts the labels for the examples in the query set Q_i, following the same methodology as in meta-training. This formulation avoids reliance on few-shot parameter adaptation from prompt-specific data and supports robust generalization to completely unseen prompts.

4 *MAPLE* for Cross-Prompt AES

In this section, we describe *MAPLE* by introducing our meta-learning task formulation for cross-prompt AES, outlining our different training strategies (Figure 1), and presenting our model architecture and prediction process (Figure 2).



(a) Meta-learner architecture. The input consists of an essay x, trait rubric r, and prompt p. The output is the essay representation h'.



(b) The prediction process involves two steps. The meta-learner first encodes the support-set for centroids C_i computation. Then, the unseen essay representation is compared with these centroids to predict the score l.

Figure 2: Overview of *MAPLE* showing (a) the meta-learner architecture and (b) the prediction process.

4.1 Meta-training Task Formulation

Cross-prompt AES aims to train a model on essays from a set of *source* prompts $P^s = \{p_i^s\}$ and test its generalization capability on essays from an unseen *target* prompt p^t. A direct adaptation of cross-prompt AES into the meta-learning framework is to frame each prompt p_i^s as a distinct task T. However, this approach presents a significant challenge: the number of unique meta-training tasks is limited by the source prompts, reducing task diversity which is essential for effective meta-learning.

To address this, we propose two training formulations: i) *binary* classification, where the number of tasks is bounded by the combination of prompts, traits, and score levels; and ii) *multiclass* classification, where the number of tasks is bounded by the combination of prompts and traits.[3] These settings are shown in the meta-training tasks in Figure 1.

Binary classification Inspired by Wu et al. (2021), each meta-training task T in this setup is represented as a tuple $(p, r, l^\star)$, where p is a prompt from which essays are sampled, r is a writing trait, and $l^\star$ is a score level. For each sampled task, we construct two classes: a positive class $(p, r, l = l^\star)$ and a negative class $(p, r, \bar{l} \neq l^\star)$ from which we sample k essays each, substantially increasing task diversity for potentially improving meta-learning performance (Iwata and Kumagai, 2020).

Multiclass classification In this setup, each meta-training task T is represented as a tuple (p, r), where p is a prompt and r is a writing trait. For each sampled task, we draw k examples from *each* score level of trait r, representing a class per level. While this reduces the number of tasks compared to the binary formulation, it better mirrors the actual scoring process, which is inherently a multiclass classification task.

In both setups, every task is constructed as a cross-prompt task, i.e., the support set and query set *always* come from different prompts, as illustrated in Figure 1. Specifically, we first sample one prompt for the query set, and then select a different prompt (or prompts) for the support set. In fact, we experiment with two variants for constructing the support set: **one-prompt** (denoted as **1P**), where the support set is drawn from one single prompt different from the query prompt, and **multiple prompts** (denoted as **mP**), where the support set is drawn from all available prompts except the query prompt.

4.2 Meta-testing

After training the learner f_θ, the model is evaluated on new tasks $\mathcal{T}_{tst} = \{T_i\}$ constructed from *unseen* prompts. Each task T_i is defined by a trait r and a prompt p_{tst}, formulated as an N-way classification problem, where N is the number of score levels of trait r. To satisfy the cross-prompt condition, the support set S_i is sampled from training essays written for different prompts other than p_{tst}, while the query set Q_i consists of essays from p_{tst}.

[3]We note that we follow Wu et al. (2021) by excluding tasks that have less than $k + m$ samples, where k and m denote the numbers of samples in the support set and query set, respectively.

For each trait r, prototypes are constructed from the support set by averaging the embeddings of essays that share the same score level, as defined in Equation 1. In this case, k corresponds to all training essays with score l for trait r. Each query essay x_j^q is then scored by assigning it to the nearest prototype centroid, following Equation 3.

4.3 Incorporating More Context

We anticipate a potential bottleneck in the above approach; although the centroids are computed for each trait separately, the representations of the essays are the same for different traits. Furthermore, having essays from different prompts within the same meta-learning task necessitates providing richer contextual information about those prompts. Accordingly, we include the **prompt and rubric** texts as additional inputs to the learner.

Figure 2(a) shows the architecture of the meta-learner model. $f(p)$ and $f(r)$ denote the prompt and rubric representations, respectively, obtained by the *pretrained* encoder f. $f_\theta(x)$ is the learned essay representation by the learner f_θ. To effectively combine these sources of information, we concatenate them into a joint vector, $z = [f_\theta(x); f(p); f(r)]$, and introduce a gating mechanism that allows the model to selectively weight each component in a task-dependent manner.[4] This adaptive fusion ensures that context from prompts and rubrics is emphasized when relevant for each task. The element-wise gates are computed as:

$$g = \sigma(W_z z), \tag{5}$$

where $W_z \in \mathbb{R}^{3d \times 3d}$ is a learnable parameter matrix and $\sigma(\cdot)$ is the sigmoid function. The gated representation h is obtained as:

$$h = z \odot g, \tag{6}$$

where $\odot$ denotes element-wise multiplication. Finally, h is passed through a projection layer:

$$h' = W_2\, ReLU(\mathcal{O}(W_1 h + b_1)) + b_2, \tag{7}$$

where $W_1 \in \mathbb{R}^{3d \times 3d}$, $W_2 \in \mathbb{R}^{3d \times d}$, b_1, and b_2 are learnable parameters, and $\mathcal{O}(\cdot)$ denotes dropout with rate 0.5. The output $h' \in \mathbb{R}^d$ serves as the final representation for scoring.

Moreover, engineered features have proven useful, particularly in cross-prompt setups (Do et al., 2023; Sayed et al., 2025). Hence, we explore incorporating a set of engineered features, $u(x) \in \mathbb{R}^{d_u}$, into the essay representation. Specifically, we extend z to be $[f_\theta(x); f(p); f(r); u(x)]$ and apply the same gating and projection steps to obtain h'. The prediction process is outlined in Figure 2(b).

5 Experimental Setup

This section outlines the experimental setup, covering encoder selection, training and meta-learning specifications, hyper-parameters, datasets, evaluation metric, and baselines.

5.1 Hand-crafted Features

For ELLIPSE and ASAP, we use the 86 feature-set proposed by Ridley et al. (2020), which covers length-based, readability, text complexity, text variation, and sentiment features. For Arabic, we use the 816 features introduced by Sayed et al. (2025), including surface, readability, lexical, semantic, and syntactic aspects.

5.2 Encoder Selection

Encoder model selection was based on GPU compatibility and performance, leading us to choose models under 200M parameters: AraBERTv2[5] for Arabic and RoBERTa[6] for English.

5.3 Training and Hyperparameters

All experiments were conducted using 4 NVIDIA A10-24Q GPUs with PyTorch mixed precision training and Adam optimizer. The number of shots in meta-training was fixed to 5 for both the support (k) and query (m) sets. For the multiclass classification setup, the maximum number of classes was set to 5 to align with computational constraints. The model was trained for 30K meta-training tasks with a batch size of 12 (the maximum our hardware resources could accommodate). The best-performing model on the dev set was selected for evaluation.

During the meta-training setup selection phase, the learning rate and number of learnable encoder layers were initially fixed at $1e^{-5}$ and 3, respectively. Before training the final model for testing, these hyperparameters were tuned over the following values: for AraBERT, learning rate $\in \{1e^{-5}, 5e^{-6}, 1e^{-6}\}$ and number of tunable layers $\in \{3, 6, 9\}$; for RoBERTa, learning rate $\in$

[4] Preliminary experiments with existing attention-based fusion and GLU showed lower performance.

[5] https://huggingface.co/aubmindlab/bert-base-arabertv2

[6] https://huggingface.co/FacebookAI/roberta-base

$\{1e^{-5}, 3e^{-5}, 5e^{-5}\}$ and number of tunable layers $\in \{3, 6, 9\}$. The best hyper-parameters for each dataset are reported in Appendix B.

5.4 Datasets and Evaluation

English Data For experimenting on English data, we use two datasets: ELLIPSE (Crossley et al., 2023) and ASAP[7]/ASAP++ (Mathias and Bhattacharyya, 2018). ELLIPSE serves as our main English dataset, as it provides a consistent score range across prompts, making it well-suited to demonstrate the strengths of *MAPLE*. ASAP is primarily used to compare *MAPLE* to existing baselines; however, its varying score ranges across prompts make some traits more challenging, as certain score levels are unrepresented in the support set.

ELLIPSE comprises about 6.5k essays written by English Learners for 44 distinct prompts. The essays, written by students in grades 8-12, are assessed across 7 traits: cohesion (COH), syntax (SYN), vocabulary (VOC), phraseology (PHR), grammar (GRM), conventions (CNV), and holistically (HOL), with an average length of 427 words. Each trait is scored using a standardized rubric on a 1-5 scale, with 0.5-point increments. ASAP dataset contains approximately 13k essays across 8 prompts, where trait-level annotations are available for prompts 7 and 8. ASAP++ extends ASAP by providing trait annotations for prompts 1–6. The evaluated traits include Content (CNT), Organization (ORG), Word Choice (WC), Sentence Fluency (SF), Conventions (CNV), Prompt Adherence (PA), Language (LNG), and Narrativity (NAR). Since score ranges vary across prompts in ASAP, we shift all scores to start from 0 to align them across prompts. ASAP per-prompt traits and score ranges are described in Appendix A.[8] Although ASAP dataset includes holistic scores, the score ranges vary substantially across prompts. Thus, we exclude holistic scoring and focus on trait-level evaluation, leaving this issue for future work.

Arabic Data For Arabic, we used **LAILA** dataset (Bashendy et al., 2026), comprising 7,859 essays written by native high school students under test-like conditions across 8 different prompts. Each essay is annotated across 7 traits: Relevance (REL), Organization (ORG), Vocabulary (VOC), Style (STY), Development (DEV), Mechanics (MEC), and Grammar (GRM), in addition to a Holistic (HOL) score computed as the sum of all trait scores, with an average length of 171 words. All traits are assessed on a scale of 0-5, except REL on a scale of 0-2, using 1-point increments.

Data Splits We employed different cross-validation strategies for the English and Arabic datasets. For ELLIPSE, we followed the 11-fold cross-validation splits proposed by Eltanbouly et al. (2025), with 40 prompts for training and 4 unseen prompts for testing. Within the training set, each prompt is partitioned into training and development subsets using an 80/20 split. For ASAP, we adopt the 8-fold leave-one-prompt-out cross-validation setup of Ridley et al. (2020), where for each fold, seven prompts are used for training and development, and one unseen prompt is reserved for testing. For LAILA, we use the public cross-prompt splits provided by the dataset authors,[9] following an 8-fold leave-one-prompt-out setup, with 5 training, 2 development, and 1 test prompt for each fold.

Evaluation Metric To assess model performance, we employ Quadratic Weighted Kappa (QWK) (Cohen, 1968), a widely adopted metric in AES that quantifies agreement between human annotators and system predictions.

5.5 Baselines

We compare our proposed approach with SOTA baseline models for both English and Arabic AES.

English Baselines On ELLIPSE, we compare with TRATES (Eltanbouly et al., 2025) (the SOTA method on ELLIPSE) and MOOSE (Chen et al., 2025) (the SOTA method on ASAP), which we ran on ELLIPSE using their publicly available code.[10]

On ASAP, we compare with MOOSE and EPCTS (Xu et al., 2025), the current two best performing models on ASAP. Upon examination of the released code,[10] we note that the reported results of MOOSE on ASAP (Table 3 in (Chen et al., 2025)) were based on tuning on the *test* set, rather than the *dev* set. We confirmed that by running the code and managed to almost reproduce the reported results.[11] However, with properly tuning on the dev set,[12] we obtained an average score of 0.538. Table 2 reports both the original paper's test-optimized

[7] https://www.kaggle.com/c/asap-aes

[8] We refer to ASAP/ASAP++ as ASAP hereafter.

[9] https://gitlab.com/bigirqu/laila

[10] https://github.com/antslabtw/MOOSE-AES

[11] We obtained a score of 0.635 compared to 0.640 (avg w/o overall) reported in the paper.

[12] Using the same set of the fixed hyperparameters outlined in (Chen et al., 2025).

results (which are not comparable to the others, and marked by *) and also the dev-optimized results for fair and transparent comparison.

Arabic Baselines We select the best performing model on LAILA under the cross-prompt setup, specifically MOOSE, where Bashendy et al. (2026) replaces BERT with AraBERT and incorporates features proposed by Sayed et al. (2025). Additionally, we compare with XGB, the second-best model, which also uses the same set of features.

6 Experimental Evaluation

We aim to address the following research questions in the context of cross-prompt AES: **(RQ1)** Which meta-training formulation, multiclass or binary classification, yields the best performance? **(RQ2)** How does incorporating prompt and rubric information influence performance? **(RQ3)** What is the impact of integrating handcrafted features on the performance? **(RQ4)** How does *MAPLE* compare to baseline SOTA models for English and Arabic AES?

This section answers the research questions and discusses the corresponding results. We report performance results on dev sets as we navigate through the best configurations of our framework (the first three RQs). Finally, we compare the best setup with SOTA on test sets (RQ4).

6.1 Meta-training Setups (RQ1)

RQ1 examines the different meta-learning setups we proposed in §4.1. We consider four setups: **binary-1P**, **binary-mP**, **multiclass-1P**, and **multiclass-mP**, where binary/multiclass indicates the task classification type, and 1P/mP specifies whether the support set is sampled from a single or multiple prompts. Datasets' average performance on the dev sets is reported in Table 1 (detailed pertrait results are shown in Table 5 in Appendix C).

On ELLIPSE, multiclass classification tasks outperform the binary tasks and are consistent across different support-set sampling strategies, achieving gains of 3 and 1 points for the **1P** and **mP** strategies, respectively. Over ASAP, multiclass tasks outperform binary tasks; however, the **1P** strategy performs better than **mP** by approximately 2 points. On LAILA, in contrast, binary setups generally perform better. Although this differs from the inference scenario, it supports prior findings that more tasks improve performance (Wu et al., 2021).

Dataset	Setup	Avg
ELLIPSE	Binary-1P	0.548
	Binary-mP	0.574
	Multiclass-1P	0.578
	Multiclass-mP	**0.581**
	+PR	0.587•
	+PR +Features	0.592•
ASAP	Setup	Avg
	Binary-1P	0.469
	Binary-mP	0.474
	Mutliclass-1P	**0.622**
	Multiclass-mP	0.603
	+PR	0.639•
	+PR +Features	0.645•
LAILA	Setup	Avg
	Binary-1P	0.599
	Binary-mP	**0.601**
	Multiclass-1P	0.596
	Multiclass-mP	0.588
	+PR	0.614•
	+PR +Features	0.631•

Table 1: Average performance of the meta-training setups in QWK on the dev sets. Bold indicates the best-performing setup, and • marks improvements over the best setup. PR indicates prompt and rubric.

The differences across datasets arise from the number of prompts and score granularity. LAILA has few prompts with coarse scores, making binary classification effective by reducing confusion and yielding stable prototypes. ASAP also has few prompts but uses heterogeneous score ranges, hence, aggregating across prompts leads to incompatible prototypes, favoring the 1P setup. In contrast, ELLIPSE has many prompts with fine-grained scores and limited samples per score, where multiclass classification better captures subtle distinctions. Based on these observations, we adopt the **Multiclass-mP**, **Mutliclass-1P**, and **Binary-mP** setups for subsequent experiments on ELLIPSE, ASAP, and LAILA, respectively.

6.2 Effect of Adding Writing Context (RQ2)

We next investigate whether incorporating the rubric and prompt information improves performance by providing richer task context and clearer distinctions across traits and topics. Results in Table 1 show that the performance improves over all datasets when prompt and rubric representations are included, but the improvement is more pro-

Dataset	Model	COH	SYN	VOC	PHR	GRM	CVN	HOL	Avg^{-H}	Avg
ELLIPSE	MOOSE	0.207	0.227	0.168	0.209	0.146	0.216	0.226	0.195	0.200
	TRATES	0.519	0.540	0.522	0.525	0.512	0.561	-	0.530	-
	MAPLE	**0.575**	**0.616**	**0.617**	**0.639**	**0.607**	**0.633**	**0.700**	**0.615**	**0.627**
Dataset	Model	CNT	ORG	WC	SF	CNV	PA	LNG	NAR	Avg
ASAP	MOOSE*	0.651	0.652	0.634	0.643	0.604	0.649	0.624	0.665	0.640
	MOOSE	0.559	0.533	0.570	0.559	0.464	0.542	0.506	0.569	0.538
	EPCTS	**0.630**	**0.606**	**0.614**	**0.617**	**0.525**	0.630	0.613	0.647	**0.610**
	MAPLE	0.555	0.465	0.483	0.529	0.482	**0.650**	**0.633**	**0.685**	0.560
Dataset	Model	REL	ORG	VOC	STY	DEV	MEC	GRA	HOL	Avg
LAILA	MOOSE	**0.411**	0.627	0.642	0.649	0.585	0.586	0.623	0.649	0.597
	XGB	0.360	0.645	0.641	0.641	0.583	0.577	0.619	0.679	0.593
	MAPLE	0.290	**0.647**	**0.686**	**0.702**	**0.607**	**0.634**	**0.664**	**0.723**	**0.619**

Table 2: *MAPLE* performance in QWK on the test sets compared to SOTA baselines. Avg^{-H} denotes average performance without HOL scoring. **Bold** values indicate best performance per dataset per trait, excluding MOOSE*.

nounced on LAILA and ASAP than ELLIPSE.

Examining the trait-level results, we note that prompt-independent traits benefit more from incorporating the rubric and prompt information than prompt-dependent traits. Additionally, larger gains are observed on traits with detailed rubrics and datasets with longer prompt texts. More detailed per-trait analyses for each dataset are provided in Appendix C.1.

6.3 Effect of Adding Features (RQ3)

Given the performance boost gained by incorporating the contextual representations on all datasets, we next examine the effect of incorporating a feature-engineered vector into the essay representation. As shown in Table 1, incorporating these features had a positive influence over all datasets, with an improvement of about 0.7 points on ASAP, 0.5 points on ELLIPSE, and 2 points on LAILA.

Examining the trait-level results (detailed in Appendix C.2), we note that improved traits emphasize structural, lexical, or surface-level aspects of writing, such as grammar and conventions on ELLIPSE, organization and content on ASAP, and organization and vocabulary on LAILA. These results suggest that the features capture aspects of clarity, correctness, and structure.

6.4 *MAPLE* vs. SOTA (RQ4)

Based on the above findings, we evaluate *MAPLE* on the test set of the three datasets using the configuration that incorporates the prompt, rubric, and feature information. Table 2 compares *MAPLE* against SOTA baselines over the three datasets.

ELLIPSE *MAPLE* outperforms the best baseline model by 8.5 points on ELLIPSE (on average across all traits but HOL, since TRATES, the best baseline, focuses solely on trait-level scoring). The improvement is pronounced on *all traits* by 5-10 points. These results establish a new important benchmark for ELLIPSE, a dataset that has been underutilized despite addressing limitations in ASAP (Li and Ng, 2024a).

ASAP On average, *MAPLE* outperforms MOOSE but lags behind EPCTS. This comparatively-lower performance is expected, since ASAP prompts have *heterogeneous* score ranges (see Table 4 in Appendix A). For example, prompt 8 has 11 score levels, whereas other prompts have a maximum of 6, leaving 5 score levels unrepresented in the support set.[13] Nevertheless, *MAPLE* demonstrates strong performance on traits that have *unified* score ranges, specifically PA, LNG, and NAR, achieving 2-4 points improvements over SOTA models and setting new SOTA performance for these traits. Detailed ASAP results, including per-prompt comparisons with baselines, as well as per-trait and per-prompt breakdowns, are provided in Appendix D.

LAILA *MAPLE* outperforms the best baseline models by 3 points on LAILA, achieving an average gain of 4 points across all traits except REL. The improvements are most pronounced on HOL with a 7-point gain, and on STY and MEC with 5-point gains each. For transparency, we report per-prompt and per-trait results in Appendix D.

[13]We leave addressing this issue to future work.

Overall, *MAPLE* exhibits SOTA performance when score ranges are unified, as seen in ELLIPSE, LAILA, and the unified-range traits in ASAP, demonstrating its effectiveness in cross-prompt AES. Moreover, these findings underscore the importance of evaluating AES models across diverse datasets to obtain a more comprehensive assessment of their generalization ability.

7 Conclusion and Future Work

In this work, we propose *MAPLE*, a meta-learning framework based on prototypical networks for cross-prompt AES that integrates prompt, rubric, and feature representations with essay embeddings to enhance generalization across prompts and traits. Through systematic experiments on three English and Arabic datasets, results showed that *MAPLE* achieves SOTA performance on two datasets, outperforming strong baselines by an impressive 8.5 points on ELLIPSE and up to 3 points on LAILA, demonstrating the potential of meta-learning for building adaptable, cross-prompt AES systems. ASAP dataset was more challenging due to its varying score ranges; however, for traits unaffected by this issue, improvements reached up to 4 points. For future work, we plan to address the score range variability, explore auxiliary tasks to further enhance model generalizability, and extend the approach to multilingual settings by incorporating a language dimension into task definitions.

Acknowledgments

This work was made possible by NPRP grant# NPRP14S-0402-210127 from the Qatar Research Development and Innovation (QRDI) Council. The statements made herein are solely the responsibility of the authors.

Limitations

While *MAPLE* demonstrates strong performance in cross-prompt AES, several limitations exist.

First, we experimented with a single encoder model for each language, therefore, the effect of varying pre-trained models on *MAPLE*'s performance remains an open question.

Second, we fixed the size of the support and query sets to 5 examples during meta-training for computational efficiency, though tuning this hyperparameter could improve the performance.

Third, one limitation becomes evident in datasets with different score ranges across the prompts (e.g., ASAP dataset), as *MAPLE*, a classification-based approach, does not inherently handle differences in score ranges. Unlike regression models, where scores can be scaled to a unified range during training and rescaled to their original ranges during inference, such scaling is not intuitive in the classification setting. This is because score discretization requires rounding, which can introduce mismatches in the mapping of scores across prompts with different score ranges and when scaling the scores back during evaluation. Nevertheless, *MAPLE* demonstrates superior performance on datasets and prompts with unified score ranges. Adapting *MAPLE* to prompts or traits with varying score ranges remains open for future work.

Finally, although we explored the combined effect of prompt and rubric information, we did not isolate their individual contributions. A more granular investigation of each component could provide deeper insights into their specific roles in enhancing model performance.

References

Nada Almarwani, Alaa Alharbi, and Samah Aloufi. 2025. Taibah at TAQEEM 2025: Leveraging GPT-4o for Arabic essay scoring. In *Proceedings of The Third Arabic Natural Language Processing Conference: Shared Tasks*, pages 989–997, Suzhou, China. Association for Computational Linguistics.

Mohamad Alnajjar, Ahmad Almoustafa, Tomohiro Nishiyama, Shoko Wakamiya, Eiji Aramaki, and Takuya Matsuzaki. 2025. ARxHYOKA at TAQEEM2025: Comparative approaches to Arabic essay trait scoring. In *Proceedings of The Third Arabic Natural Language Processing Conference: Shared Tasks*, pages 977–982, Suzhou, China. Association for Computational Linguistics.

May Bashendy, Salam Albatarni, Sohaila Eltanbouly, Walid Massoud, Houda Bouamor, and Tamer Elsayed. 2025. TAQEEM 2025: Overview of the first shared task for Arabic quality evaluation of essays in multi-dimensions. In *Proceedings of The Third Arabic Natural Language Processing Conference: Shared Tasks*, pages 966–976, Suzhou, China. Association for Computational Linguistics.

May Bashendy, Salam Albatarni, Sohaila Eltanbouly, Eman Zahran, Hamdo Elhuseyin, Tamer Elsayed, Walid Massoud, and Houda Bouamor. 2024. QAES: First publicly-available trait-specific annotations for automated scoring of Arabic essays. In *Proceedings of the Second Arabic Natural Language Processing Conference*, pages 337–351, Bangkok, Thailand. Association for Computational Linguistics.

May Bashendy, Walid Massoud, Sohaila Eltanbouly, Salam Albatarni, Marwan Sayed, Abrar Abir, Houda

Bouamor, and Tamer Elsayed. 2026. LAILA: A large trait-based dataset for Arabic automated essay scoring. In *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2026)*. Association for Computational Linguistics.

Jill Burstein. 2013. *Handbook of automated essay evaluation: Current applications and new directions*. Routledge.

Yue Cao, Hanqi Jin, Xiaojun Wan, and Zhiwei Yu. 2020. Domain-adaptive neural automated essay scoring. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '20, page 1011–1020, New York, NY, USA. Association for Computing Machinery.

Po-Kai Chen, Bo-Wei Tsai, Shao Kuan Wei, Chien-Yao Wang, Jia-Ching Wang, and Yi-Ting Huang. 2025. Mixture of ordered scoring experts for cross-prompt essay trait scoring. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 18071–18084, Vienna, Austria. Association for Computational Linguistics.

Yuan Chen and Xia Li. 2023. PMAES: Prompt-mapping contrastive learning for cross-prompt automated essay scoring. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1489–1503, Toronto, Canada. Association for Computational Linguistics.

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 (LREC-COLING 2024)*, pages 12775–12786, Torino, Italia. ELRA and ICCL.

Jacob Cohen. 1968. Weighted kappa: nominal scale agreement provision for scaled disagreement or partial credit. *Psychological bulletin*, 70(4):213.

Scott Crossley, Yu Tian, Perpetual Baffour, Alex Franklin, Youngmeen Kim, Wesley Morris, Meg Benner, Aigner Picou, and Ulrich Boser. 2023. The English language learner insight, proficiency and skills evaluation (ELLIPSE) corpus. *International Journal of Learner Corpus Research*, 9(2):248–269.

Heejin Do, Yunsu Kim, and Gary Geunbae Lee. 2023. Prompt- and trait relation-aware cross-prompt essay trait scoring. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 1538–1551, Toronto, Canada. Association for Computational Linguistics.

Sohaila Eltanbouly, Salam Albatarni, and Tamer Elsayed. 2025. TRATES: Trait-specific rubric-assisted cross-prompt essay scoring. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 20528–20543, Vienna, Austria. Association for Computational Linguistics.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. 2017. Model-agnostic meta-learning for fast adaptation of deep networks. In *International conference on machine learning*, pages 1126–1135. PMLR.

Tomoharu Iwata and Atsutoshi Kumagai. 2020. Meta-learning from tasks with heterogeneous attribute spaces. *Advances in Neural Information Processing Systems*, 33:6053–6063.

Rahul Kumar, Sandeep Mathias, Sriparna Saha, and Pushpak Bhattacharyya. 2022. Many hands make light work: Using essay traits to automatically score essays. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1485–1495, Seattle, United States. Association for Computational Linguistics.

Shengjie Li and Vincent Ng. 2024a. Automated essay scoring: A reflection on the state of the art. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 17876–17888, Miami, Florida, USA. Association for Computational Linguistics.

Shengjie Li and Vincent Ng. 2024b. Conundrums in cross-prompt automated essay scoring: Making sense of the state of the art. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7661–7681, Bangkok, Thailand. Association for Computational Linguistics.

Sandeep Mathias and Pushpak Bhattacharyya. 2018. ASAP++: Enriching the ASAP automated essay grading dataset with essay attribute scores. In *Proceedings of the eleventh international conference on language resources and evaluation (LREC 2018)*.

Ellis B Page. 1966. The imminence of... grading essays by computer. *The Phi Delta Kappan*, 47(5):238–243.

Robert Ridley, Liang He, Xin-yu Dai, Shujian Huang, and Jiajun Chen. 2021. Automated cross-prompt scoring of essay traits. In *Proceedings of the AAAI 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. *arXiv preprint arXiv:2008.01441*.

Marwan Sayed, Sohaila Eltanbouly, May Bashendy, and Tamer Elsayed. 2025. Feature engineering is not dead: A step towards state of the art for Arabic automated essay scoring. In *Proceedings of the Arabic Natural Language Processing Conference (ArabicNLP 2025)*, China.

Jake Snell, Kevin Swersky, and Richard Zemel. 2017. Prototypical networks for few-shot learning. *Advances in neural information processing systems*, 30.

Ishan Tarunesh, Sushil Khyalia, Vishwajeet Kumar, Ganesh Ramakrishnan, and Preethi Jyothi. 2021. Meta-learning for effective multi-task and multilingual modelling. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 3600–3612, Online. Association for Computational Linguistics.

Eleni Triantafillou, Tyler Zhu, Vincent Dumoulin, Pascal Lamblin, Utku Evci, Kelvin Xu, Ross Goroshin, Carles Gelada, Kevin Swersky, Pierre-Antoine Manzagol, et al. 2019. Meta-dataset: A dataset of datasets for learning to learn from few examples. *arXiv preprint arXiv:1903.03096*.

Niels van der Heijden, Helen Yannakoudakis, Pushkar Mishra, and Ekaterina Shutova. 2021. Multilingual and cross-lingual document classification: A meta-learning approach. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1966–1976, Online. Association for Computational Linguistics.

Jiong Wang, Qing Zhang, Jie Liu, Xiaoyi Wang, Mingying Xu, Liguang Yang, and Jianshe Zhou. 2025. Making meta-learning solve cross-prompt automatic essay scoring. *Expert Systems with Applications*, 272:126710.

Mike Wu, Noah Goodman, Chris Piech, and Chelsea Finn. 2021. Prototransformer: A meta-learning approach to providing student feedback. *arXiv preprint arXiv:2107.14035*.

Jiangsong Xu, Jian Liu, Mingwei Lin, Jiayin Lin, Shenbao Yu, Liang Zhao, and Jun Shen. 2025. EPCTS: Enhanced prompt-aware cross-prompt essay trait scoring. *Neurocomputing*, 621:129283.

Zijie Zeng, Lin Li, Quanlong Guan, Dragan Gašević, and Guanliang Chen. 2023. Generalizable automatic short answer scoring via prototypical neural network. In *International Conference on Artificial Intelligence in Education*, pages 438–449. Springer.

Chunyun Zhang, Jiqin Deng, Xiaolin Dong, Hongyan Zhao, Kailin Liu, and Chaoran Cui. 2025. Pairwise dual-level alignment for cross-prompt automated essay scoring. *Expert Systems with Applications*, 265:125924.

A ASAP Statistics

Table 4 presents the statistics for ASAP dataset, including the traits associated with each prompt and their corresponding score ranges. We include these details primarily to highlight the variability in score ranges across prompts in ASAP dataset.

B *MAPLE* Hyperparameters

We report the best hyperparameters obtained after tuning the learning rate and the number of trainable layers for each dataset in Table 3.

Dataset	Learning rate	Trainable layers
ASAP	$1e^{-5}$	3
ELLIPSE	$1e^{-5}$	3
LAILA	$1e^{-5}$	9

Table 3: Best hyperparameters for each dataset.

C *MAPLE* Detailed Results

This section presents the detailed results of the different meta-training setups at the trait level (Table 5) and analyzes how incorporating writing context (Section C.1) and features (Section C.2) affect the performance of *MAPLE*.

C.1 Adding Writing Context

We particularly investigate the effect of incorporating the prompt and rubric on the performance of *MAPLE*. Table 5 shows that for ELLIPSE dataset, prompt-independent traits, e.g., cohesion, conventions, and grammar, benefit the most, with gains of 1-2 points, while prompt-dependent traits like vocabulary and phraseology show little or no improvement. This difference could be due to the extremely-short prompts of ELLIPSE (often just a couple of words), providing limited topical context for the prompt-dependent traits. In contrast, the rubric mainly benefits prompt-independent traits, which evaluate general writing quality.

On ASAP, traits with longer rubrics (e.g., sentence fluency and word choice) exhibit the largest gains, improving by 2-4 points, whereas traits with shorter rubrics (e.g., organization and narrativity) show only 1-point improvement.

Over LAILA, almost all traits improved. Prompt-independent traits, e.g., organization, style, and grammar, achieved the largest gains (2.2-2.5 points), while prompt-dependent traits like development and vocabulary improved less (1.4-1.8 points). Relevance showed little change, likely due to the narrow scoring range (0-2), which limits measurable improvement. The difference between ELLIPSE and LAILA may be attributed to the richer prompts of LAILA, allowing the model to better leverage both the rubric and the prompts.

Prompt	Scores	Ave Length	Essays	CNT	ORG	WC	SF	CNV	PA	LNG	NAR
P1	1 - 6	350	1783	✓	✓	✓	✓	✓			
P2	1 - 6	350	1800	✓	✓	✓	✓	✓			
P3	0 - 3	100	1726	✓					✓	✓	✓
P4	0 - 3	100	1772	✓					✓	✓	✓
P5	0 - 4	125	1805	✓					✓	✓	✓
P6	0 - 4	150	1800	✓					✓	✓	✓
P7	0 - 3	300	1569	✓	✓			✓			
P8	2 - 12	600	723	✓	✓	✓	✓	✓			

Table 4: A description of the ASAP Datasets: Scores, Average essay length in terms of words, and Traits.

Dataset	Setup	COH	SYN	VOC	PHR	GRM	CVN	HOL		Avg
ELLIPSE	Binary-1P	0.492	0.554	0.541	0.555	0.530	0.542	0.624		0.548
	Binary-mP	0.515	0.575	0.580	0.594	0.548	0.559	0.647		0.574
	Mutliclass-1P	0.514	0.581	0.566	0.597	**0.554**	**0.577**	0.658		0.578
	Multiclass-mP	**0.516**	**0.583**	**0.586**	**0.603**	0.549	0.566	**0.660**		**0.581**
	+PR	0.534•	0.589•	0.581	0.604•	0.557•	0.583•	0.663•		0.587•
	+PR +Features	0.538•	0.590•	0.587•	0.610•	0.565•	0.585•	0.671•		0.592•
ASAP	Setup	CNT	ORG	WC	SF	CNV	PA	LNG	NAR	Avg
	Binary-1P	0.450	0.327	0.357	0.384	0.352	0.631	0.598	0.655	0.469
	Binary-mP	0.429	0.355	0.392	0.427	0.357	0.608	0.587	0.641	0.474
	Multiclass-1P	**0.625**	**0.593**	**0.531**	**0.561**	**0.588**	**0.696**	0.664	**0.718**	**0.622**
	Multiclass-mP	0.616	0.579	0.494	0.502	0.569	0.691	**0.670**	0.703	0.603
	+PR	0.649•	0.604•	0.556•	0.602•	0.597•	0.704•	0.670•	0.727•	0.639•
	+PR +Features	0.661•	0.626•	0.549	0.588	0.612•	0.713•	0.676•	0.737•	0.645•
LAILA	Setup	REL	ORG	VOC	STY	DEV	MEC	GRM	HOL	Avg
	Binary-1P	0.306	0.594	**0.668**	0.665	0.573	0.622	0.649	**0.713**	0.599
	Binary-mP	0.300	**0.606**	0.666	**0.669**	**0.581**	**0.626**	**0.649**	0.709	**0.601**
	Multiclass-1P	**0.308**	0.596	0.657	0.666	0.578	0.620	0.643	0.697	0.596
	Multiclass-mP	0.292	0.593	0.647	0.662	0.573	0.617	0.640	0.682	0.588
	+PR	0.303•	0.628•	0.680•	0.688•	0.606•	0.629•	0.654•	0.724•	0.614•
	+PR +Features	0.275	0.662•	0.715•	0.706•	0.626•	0.638•	0.676•	0.747•	0.631•

Table 5: Performance of the meta-training setups in QWK on the dev sets. Bold indicates the best-performing setup, and • marks configurations that improved over the previous setting. PR indicates prompt and rubric.

C.2 Adding Features

A closer look at the rubrics explains why certain traits benefited more from incorporating the features. The added features primarily improve traits captured by surface-level and structural cues. For example, on ELLIPSE, traits aligned with text structure and correctness specifically, overall and grammar, improved the most, by nearly 1 point. Moreover, on ASAP, organization, which emphasizes logical sequencing of ideas, and conventions, which covers proper grammar and punctuation, showed improvements of about 2 points. In contrast, more substantial improvements were shown on LAILA, ranging from 1-4 points across most traits. The REL trait, however, was an exception, with performance decreasing by about 3 points. We note that relevance primarily depends on the semantic content of the essay, rather than syntactic or surface-level properties. As a result, adding feature-engineered representations, which focused more on structure, style, or readability, may have introduced noise rather than useful information for this trait.

D *MAPLE* vs SOTA Detailed Results

Since per-prompt comparison is a standard practice in AES literature, we report prompt-wise average performance of *MAPLE* on ELLIPSE, ASAP, and LAILA datasets in Tables 6, 7, and 8, respectively. We note that for ASAP dataset, our results are not directly comparable to EPCTS (Xu et al., 2025), as we do not predict the holistic trait in *MAPLE* for ASAP dataset (refer to Section 5.4). Additionally, for ELLIPSE dataset, we report only MOOSE results (as its code is available), since Eltanbouly et al. (2025) do not provide per-fold results for TRATES on ELLIPSE.

Overall, *MAPLE* outperforms the baselines on ELLIPSE and LAILA. On ASAP, it surpasses MOOSE but lags behind EPCTS (though the comparison is not direct). For ASAP dataset, the issue of non-unified score ranges is also evident in the per-prompt results: *MAPLE* exceeds MOOSE on P3, P4, P5, and P6, and even outperforms EPCTS on P3. However, it performs worse than MOOSE on P7 and P8. Finally, for transparency, we detail the results of *MAPLE* on ELLIPSE, ASAP and LAILA datasets per-prompt and per-trait in Tables 9, 10 and 11, respectively.

Fold	MOOSE	MAPLE
1	0.248	0.571
2	0.348	0.583
3	0.036	0.647
4	0.053	0.647
5	0.343	0.601
6	0.122	0.643
7	0.336	0.594
8	0.326	0.651
9	0.391	0.685
10	0.000	0.608
11	-0.004	0.665
Avg	0.200	0.627

Table 6: *MAPLE* performance on ELLIPSE dataset per-fold averaged across all traits.

Prompt	MOOSE	EPCTS*	MAPLE
P1	0.610	0.659	0.633
P2	0.594	0.609	0.550
P3	0.578	0.619	0.698
P4	0.619	0.686	0.681
P5	0.477	0.671	0.648
P6	0.509	0.629	0.591
P7	0.382	0.555	0.315
P8	0.483	0.630	0.360
Avg	0.532	0.632	0.560

Table 7: *MAPLE* performance on ASAP dataset per-prompt averaged across all traits except HOL. EPCTS* is averaged over all traits, including the HOL trait, and is therefore not directly comparable.

Prompt	MOOSE	XGB	MAPLE
P1	0.426	0.400	0.537
P2	0.652	0.656	0.601
P3	0.661	0.710	0.679
P4	0.625	0.564	0.612
P5	0.442	0.549	0.500
P6	0.661	0.653	0.691
P7	0.658	0.561	0.664
P8	0.646	0.653	0.668
Avg	0.596	0.593	0.619

Table 8: *MAPLE* performance on LAILA dataset per-prompt averaged across all traits.

Fold	COH	SYN	VOC	PHR	GRA	CON	HOL	Avg
1	0.483	0.549	0.548	0.580	0.564	0.611	0.660	0.571
2	0.567	0.521	0.611	0.615	0.565	0.559	0.644	0.583
3	0.604	0.644	0.644	0.644	0.652	0.605	0.735	0.647
4	0.628	0.668	0.625	0.656	0.604	0.615	0.735	0.647
5	0.588	0.488	0.600	0.643	0.590	0.631	0.670	0.601
6	0.590	0.667	0.595	0.649	0.641	0.639	0.720	0.643
7	0.530	0.593	0.627	0.604	0.504	0.625	0.673	0.594
8	0.600	0.657	0.626	0.647	0.627	0.676	0.724	0.651
9	0.609	0.716	0.670	0.692	0.655	0.707	0.743	0.685
10	0.533	0.606	0.575	0.623	0.597	0.657	0.664	0.608
11	0.595	0.672	0.670	0.678	0.675	0.639	0.727	0.665
Avg	0.575	0.616	0.617	0.639	0.607	0.633	0.700	0.627

Table 9: Breakdown of *MAPLE* prompt and trait-wise performance on ELLIPSE dataset.

Prompt	CNT	ORG	WC	SF	CNV	PA	LNG	NAR	Avg
P1	0.447	0.648	0.682	0.705	0.684	-	-	-	0.633
P2	0.675	0.501	0.519	0.558	0.499	-	-	-	0.550
P3	0.693	-	-	-	-	0.696	0.679	0.725	0.698
P4	0.677	-	-	-	-	0.700	0.619	0.727	0.681
P5	0.682	-	-	-	-	0.643	0.617	0.648	0.648
P6	0.548	-	-	-	-	0.561	0.617	0.638	0.591
P7	0.373	0.309	-	-	0.262	-	-	-	0.315
P8	0.341	0.403	0.248	0.325	0.484	-	-	-	0.360
Avg	0.555	0.465	0.483	0.529	0.482	0.650	0.633	0.685	0.560

Table 10: Breakdown of *MAPLE* prompt and trait-wise performance on ASAP dataset.

Prompt	REL	ORG	VOC	STY	DEV	MEC	GRA	HOL	Avg
P1	0.270	0.578	0.608	0.632	0.553	0.515	0.527	0.613	0.537
P2	0.392	0.612	0.650	0.687	0.547	0.614	0.622	0.682	0.601
P3	0.216	0.690	0.716	0.792	0.665	0.752	0.791	0.814	0.679
P4	0.365	0.601	0.579	0.662	0.561	0.660	0.673	0.791	0.612
P5	0.217	0.588	0.592	0.569	0.528	0.477	0.516	0.513	0.500
P6	0.182	0.769	0.767	0.800	0.763	0.653	0.742	0.853	0.691
P7	0.277	0.668	0.791	0.726	0.661	0.700	0.736	0.751	0.664
P8	0.401	0.672	0.784	0.750	0.574	0.697	0.706	0.763	0.668
Avg	0.290	0.647	0.686	0.702	0.607	0.634	0.664	0.723	0.619

Table 11: Breakdown of *MAPLE* prompt and trait-wise performance on LAILA dataset.