Cross-Prompt Automated Essay Scoring of Multiple Traits: Making Sense of the State of the Art

Shengjie Li and Vincent Ng
Human Language Technology Research Institute
University of Texas at Dallas
Richardson, TX 75080-0688
sxl180006@utdallas.edu, vince@hlt.utdallas.edu

Abstract

Despite the recent progress made in cross-prompt essay scoring, there is little analysis of what makes a state-of-the-art cross-prompt scorer work well. To this end, we present an empirical analysis of how the key components of a cross-prompt scorer interact with each other and impact its overall performance. In addition, we examine for the first time the application of transductive learning to cross-prompt scoring, which represents an important starting point for providing a practical way to improve cross-prompt scorers for use in the rarely-studied classroom setting without the need for additional labeled training data.[1]

1 Introduction

Traditional work on automated essay scoring (AES) has focused on *within-prompt* scoring, where an AES model is trained on essays written for a prompt and then applied to test essays written for the same prompt. Some have argued that within-prompt scoring is not a practical setting: when within-prompt scorers are applied to essays written for a new prompt, their performance often deteriorates considerably. So, before they are applied to score essays written for a new prompt, they need to be retrained on scored essays written for the new prompt. However, manually scoring essays is time-consuming and requires a lot of expertise.

In light of this concern, AES researchers have recently begun work on the task of *cross-prompt* AES (Ridley et al., 2021), where the goal is to train a model that can offer good performance when it is applied to score essays written for unseen prompts. Despite its broader applicability, cross-prompt AES is arguably much more challenging than its within-prompt counterpart. Specifically, for an AES model to perform well on a new prompt, we need knowledge specific to the new prompt. For example, if the new prompt is "write a persuasive essay on whether capital punishment should be abolished", an AES model needs to distinguish between persuasive arguments and unpersuasive arguments for (or against) capital punishment. While a within-prompt scorer could acquire domain-specific knowledge from the training data, for a cross-prompt scorer such kind of knowledge has to be acquired from external sources.

Another complication concerns the fact that cross-prompt AES is typically addressed in combination with another challenging, relatively unexplored task of *multi-trait* scoring. Specifically, while traditional AES research has focused on *holistic* scoring, which aims to summarize the overall quality of an essay with a *single* number, recent years have seen a surge of interest in scoring different essay *traits* (i.e., dimensions of essay quality such as ORGANIZATION). *Multi-trait* scoring refers to the task of scoring an essay not only holistically but also along different traits. Content-based traits such as PERSUASIVENESS (i.e., how persuasive the main argument made in an essay is), which require an understanding of an essay's content, are particularly difficult to score accurately.

Despite the aforementioned challenges, researchers have made slow but steady progress on cross-prompt AES. While performance numbers continue to improve, our understanding of what makes a cross-prompt scorer work well is arguably not. Specifically, state-of-the-art cross-prompt scorers differ in many implementation details, some of which may not be emphasized or even reported in research papers, thus complicating our understanding of what exactly is contributing to their strong performance. For instance, Scorer B achieved better results than Scorer A by augmenting A's feature set with novel features while re-computing both A's features and the novel features with a different normalization method, without showing any results

[1]Our code is available at https://github.com/samlee946/LRA-AES

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 45384–45406

obtained using the original feature normalization method. This could cast the wrong impression that the improvement was due to the new features, while the improvement could have stemmed from changing the feature normalization method. What we are lacking is an understanding of how the various components of a cross-prompt scorer, including those that are lesser-discussed, interact with each other and impact overall performance.

Our goal in this paper is two-fold. First, given the above discussion, we seek to gain insights into how the different components of a cross-prompt scorer interact by examining four key components of a cross-prompt scorer, namely (1) the feature set, (2) the model architecture, (3) the feature normalization method, and (4) the metric used for scoring development-set essays, evaluating cross-prompt scorers obtained from different, including novel, implementations of each component.

Second, we investigate the deployment of cross-prompt AES scorers in a rarely-studied setting, the *transductive* setting. Recall that in AES research, the typical assumption is that the (unlabeled) test essays cannot be exploited for model training and development. This is understandable: AES research was originally motivated by the need to develop automated methods for *within-prompt* scoring of the student essays written every year for standardized aptitude tests such as GRE and SAT, where the within-prompt models were always trained with a large number of human-scored essays before they were used for within-prompt scoring.

We believe, however, that the time is ripe to determine whether we can broaden the impact of AES technologies by investigating whether AES scorers can be deployed in a *classroom* setting. In a standardized test setting, within-prompt scoring is possible because organizations such as the ETS can easily provide a large number of human-scored essays to train prompt-specific scorers. In contrast, in a classroom setting where school teachers want to employ AES systems to score student essays, cross-prompt AES is more applicable because there may not be enough resources to provide human-labeled essays to retrain within-prompt scorers for every new prompt. When deploying AES technologies in a classroom setting, we are no longer bound by the traditional constraint that the "test" essays (i.e., the essays a teacher wants a cross-prompt model to score) cannot be exploited for model training. Specifically, when the students in a class submit their essays written for a new essay prompt as part of a homework assignment, a teacher can retrain a cross-prompt model on not only the (scored) essays originally in the model's training data, but also the essays submitted by the students. This form of semi-supervised learning where (unlabeled) test data is allowed to be used for model training together with the (labeled) training data is known as *transductive* learning (Vapnik, 1998).

In this paper, we examine whether transductive learning can improve the performance of a cross-prompt AES scorer, focusing on methods that use the test instances to re-weigh the training instances, specifically by giving higher weights to those training instances that are more similar to the test instances. We are by no means claiming that our investigation of transduction for cross-prompt AES is exhaustive, but we believe our work represents a good starting point for providing a practical way to improve cross-prompt AES scorers for use in the (under-investigated) classroom setting without the need for additional labeled training data.

2 Evaluation Setup

2.1 Dataset

Following previous work (Do et al., 2023, 2025), we use ASAP[2]/ASAP++ (Mathias and Bhattacharyya, 2018a) for model training and evaluation. The dataset contains nine human-annotated scores including the holistic (a.k.a. OVERALL) score and eight additional trait scores for essays written in response to eight writing prompts, namely CONTENT, ORGANIZATION (Org), WORD CHOICE (WC), SENTENCE FLUENCY (SF), CONVENTIONS (Conv), PROMPT ADHERENCE (PA), LANGUAGE (Lang), and NARRATIVITY (Narr). However, the set of traits varies across prompts since different prompts correspond to different types of essays (narrative, persuasive, and source-dependent) and different prompts have a different set of applicable traits.[3] We employ the same train/development/test partitions and random seeds as Do et al. (2023).[4]

2.2 Evaluation Metric

We employ Quadratic Weighted Kappa[5] (QWK), the standard metric used for AES evaluation that measures the agreement between model predictions

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

[3] Details of the dataset can be found in Appendix A.

[4] Statistics on these partitions can be found in Appendix B.

[5] See https://www.kaggle.com/competitions/asap-aes/overview/evaluation for details.

and ground truth labels, as our metric. Higher QWK values indicate better model performance.

2.3 Model

We employ as our model a multi-layer neural network, which takes as input a set of features and predicts one or more trait scores. We examine four *components* of this model, as discussed below.

Model architecture: Independent vs. Joint^S vs. Joint^T. We experiment with three architectures. In the INDEPENDENT architecture, we train a separate model to score each trait in the target prompt. To examine whether jointly learning multiple traits can improve the scoring of individual traits, we use the JOINT^S architecture: given a target (i.e., test) prompt, the model jointly predicts all traits that appear in the source (i.e., training) prompts. To further investigate whether traits that are not applicable to the target prompt influence the scoring of applicable traits, we introduce the JOINT^T architecture, where the model jointly predicts only those traits applicable to the target prompt.

Input features: Basic vs. Advanced. We experiment with two feature sets: BASIC and ADVANCED. The BASIC feature set contains the set of features proposed by Li and Ng (2024) that are highly correlated with the target trait(s).[6]

Researchers have begun exploring using large language models (LLMs) to automatically score essay traits in a zero-shot manner without much success (Mansour et al., 2024; Lee et al., 2024). We instead incorporate these LLM-predicted trait scores into our model in an attempt to see whether they would be useful for scoring when used in a *supervised* fashion. In addition, since we can ask LLMs to explain their predictions by prompting, researchers have begun exploring LLM-generated rationales, but only in the context of within-prompt scoring (Chu et al., 2025). Thus, we also incorporate rationale information into our model. Thus, the ADVANCED feature set constitutes three sources of features: all BASIC features, LLM-predicted trait scores, and the embedding of the LLM-given rationale for the predicted trait scores.

Feature normalization: z-score vs. min-max We experiment with two feature normalization methods. Following Ridley et al. (2021), we experiment with z-SCORE normalization which normalizes input features such that the resulting features have zero mean and unit variance. For input feature matrix $\mathbf{F}$ of shape (M, N) where M is the number of essays from a specific essay prompt and N is the number of features, the z-score normalized feature values of essay i are obtained by

$$\mathbf{F}'_{i,j} = \frac{\mathbf{F}_{i,j} - \mu_j}{\sigma_j}$$

where μ_j and σ_j denote the mean and standard deviation of the j-th feature respectively.

Following Uto et al. (2020), we experiment with MIN-MAX normalization, which scales input features into the range of [0, 1]. The min-max normalized feature values of essay i are obtained by

$$\mathbf{F}'_{i,j} = \frac{\mathbf{F}_{i,j} - \min_x \mathbf{F}_{x,j}}{\max_x \mathbf{F}_{x,j} - \min_x \mathbf{F}_{x,j}}$$

where $\min_x \mathbf{F}_{x,j}$ and $\max_x \mathbf{F}_{x,j}$ denote the minimum and maximum feature value of the j-th feature respectively.

Development set metric: Loss vs. QWK^S vs. QWK^T When evaluating a model's performance on the development set, the most straightforward approach is to use the development loss as in Chu et al. (2025). However, AES systems are typically evaluated using QWK, a non-differentiable agreement metric that does not always align with the loss function. Therefore, in addition to using LOSS for hyperparameter selection, and following prior work (Li and Ng, 2024), we also experiment with the QWK^S setting, in which predictions on development data are rescaled to their respective *source* score ranges to account for the varying prompt-specific score ranges in ASAP. QWK is then computed between the rescaled development set predictions and the gold labels. In contrast, Do et al. (2023) rescale predictions to a unified score range, rather than using prompt-specific ones. To investigate the impact of this choice, we additionally consider the QWK^T setting, where development set predictions are rescaled to match the *target* score ranges of the corresponding test prompts.

2.3.1 Implementation Details

Network architecture. All INDEPENDENT and JOINT models are trained with two hidden layers of sizes 128 and 64, respectively, using ReLU as the activation function and mean squared error (MSE) as the loss function.[7] For the JOINT models, the

[6] Details of Li and Ng's (2024) 1535 features can be found in Appendix C.

[7] The number of hidden layers, the number of neurons per layer, and the activation function were determined during preliminary experiments on development data. MSE is a widely-used loss function in AES research.

	Setting	Overall	Content	Org	WC	SF	Conv	PA	Lang	Nar	AVG
					SUPERVISED						
1	ProTACT	**.674**	.596	**.518**	**.599**	.585	.450	.619	.596	.639	.586±.009
2	GAPS	.670	**.597**	.515	.595	**.590**	**.472**	**.621**	**.608**	**.650**	**.591±.011**
					ZERO-SHOT PROMPTING						
3	Llama2	.364	.397	.451	.460	.420	.399	.360	.392	.359	.400
4	Gemma2	.406	.394	.355	.318	.318	.312	.403	.444	.477	.381
5	Gemma3	.498	.483	.413	.337	.309	.202	.533	.478	.550	.422
6	Qwen3	.471	.456	.362	.323	.309	.217	.508	.364	.452	.385
7	Gemini-2.5-flash	.530	.516	.479	.340	.390	.295	.516	.424	.462	.439

Table 1: Baseline trait-wise QWK scores.

total loss is computed as the sum of the MSE losses across all traits. Since not all traits are applicable to every prompt, any predicted score for an inapplicable trait in the JOINT^S setting does not contribute to the loss. Each neuron in the final output layer employs the sigmoid activation function to produce a score between 0 and 1, which is then rescaled to match the valid score range.

Basic features. We construct our BASIC feature set as follows. For the INDEPENDENT model, we compute the Pearson Correlation Coefficient between each feature in Li and Ng's (2024) and the target trait on the training set, and select the top 20 features. For the JOINT models, we compute the Pearson Correlation Coefficient between each feature in Li and Ng's (2024) and each trait in the ASAP dataset, selecting the top 20 features for each trait. The BASIC feature set is then formed by combining the top 20 features across all traits and removing duplicate features.

Advanced features. For the LLM features in the ADVANCED feature set, we utilize two top-performing, open-source, instruction-tuned dense LLMs: Qwen3[8] and Gemma3[9]. Each essay from the ASAP dataset is processed in a zero-shot setting, where the LLMs are prompted using the official ASAP/ASAP++ rubrics to generate both trait-level scores and the corresponding rationales.[10]

For the rationale features in the ADVANCED feature set, we employ the top-ranked embedding model on the MTEB multilingual leaderboard[11], `Qwen3-Embedding-8B`[12], to generate 4096-dimensional embeddings for each rationale. In the INDEPENDENT model, trait-specific rationales are used directly to compute the embeddings. In the JOINT models, we first concatenate the rationales across all traits into a single text input, from which the embeddings are then derived.

Training details. All models are trained for 50 epochs maximum using AdamW optimizer with $\beta_1 = 0.9, \beta_2 = 0.999, \epsilon = 1 \times 10^{-8}$, and weight decay$= 0.01$. Following previous work (Do et al., 2023), we use five random seeds $\{12, 22, 32, 42, 52\}$. We perform a grid search for selecting the best early-stopping epoch and learning rate (by searching out of $\{0.001, 0.0001\}$) based on development set metrics.[13]

Computation cost. It takes 300 hours to obtain all LLM and rationale features, which corresponds to roughly 22 million output tokens, and an additional 24 hours to complete training for all experiments on two NVIDIA RTX A6000 48GB GPUs.

3 Understanding Component Interactions

In this section, we present the results of our first set of experiments, which aim to examine how the performance of a cross-prompt scorer is impacted by changes in the implementation of the four key components we described in the previous section.

3.1 Baseline Results

We begin by presenting baseline results in Table 1. Specifically, rows 1 and 2 show the results of two state-of-the-art cross-prompt scorers, `ProTACT` (Do et al., 2023) and `GAPS` (Do et al., 2025).[14] For each scorer, we show the OVERALL score, the trait scores, as well as the AVG score, which is obtained by taking the unweighted average of all of its trait scores, including the holistic score. Following previous work (Do et al., 2023), we compute each trait-specific score by taking the unweighted average of the QWK scores over the eight ASAP prompts. In addition, since we employ five different seeds to run these experiments, we report in the table the

[8] https://huggingface.co/Qwen/Qwen3-32B

[9] https://huggingface.co/google/gemma-3-27b-it

[10] The full prompt is provided in Appendix D.

[11] https://huggingface.co/spaces/mteb/leaderboard

[12] https://huggingface.co/Qwen/Qwen3-Embedding-8B

[13] The best hyperparameter values are in Appendix E.

[14] Descriptions of the baseline scorers are in Appendix F.

	BASIC			ADVANCED^Q			ADVANCED^G		
	Loss	QWK^S	QWK^T	Loss	QWK^S	QWK^T	Loss	QWK^S	QWK^T
INDEPENDENT									
z-score	.520±.001	.520±.006	.534±.003	.596±.002	.595±.002	**.599±.002**	.592±.001	.592±.001	.597±.004
min-max	.447±.005	.433±.005	.443±.005	.538±.002	.533±.001	.539±.002	.549±.003	.541±.003	.545±.003
JOINT^S									
z-score	.513±.004	.508±.004	.519±.004	.596±.003	**.599±.006**	.597±.004	.589±.005	.594±.004	.593±.005
min-max	.458±.006	.433±.010	.454±.005	.521±.007	.519±.007	.519±.009	.522±.005	.516±.007	.516±.005
JOINT^T									
z-score	.512±.002	.512±.002	.524±.004	.598±.007	.599±.007	.600±.006	.597±.004	.599±.004	**.603±.005**
min-max	.435±.007	.421±.011	.431±.008	.521±.007	.515±.008	.517±.011	.511±.007	.503±.004	.503±.006

Table 2: Trait-wise Average QWK scores for the four components.

results averaged over the five runs together with the corresponding standard deviations. As can be seen, ProTACT and GAPS achieve AVG QWK scores of 0.586 and 0.591, respectively.

Rows 3–8 of Table 1 show the zero-shot results of five LLMs. To get an idea of whether newer LLMs are better at cross-prompt scoring, we selected three relatively new LLMs (Qwen3, Gemma3, Gemini-2.5-Flash (Comanici et al., 2025)), and two older ones (Gemma2[15] and Llama2[16]). We ask each LLM to score each ASAP essay along each trait, including OVERALL, using the scoring rubric we provide as part of the prompt. W.r.t. the AVG score, Gemini-2.5-Flash performs significantly better than the older LLMs. While Gemma3 performs significantly better than the other open LLMs, Qwen3 does not show superior performance despite being one of the newer LLMs.[17] Nevertheless, even Gemma3 is significantly outperformed by Llama2 on some of the traits, such as ORGANIZATION, WORD CHOICE, and SENTENCE FLUENCY. Moreover, these zero-shot results are significantly worse than those of ProTACT and GAPS on all of the traits.

3.2 AVG Results

Recall that we examined four components of a cross-prompt scorer: two options for the feature set (BASIC and ADVANCED), three options for the model architecture (INDEPENDENT, JOINT^S and JOINT^T), two options for feature normalization (z-score and min-max) and three options for development set metric (Loss, QWK^S and QWK^T). If we compose cross-prompt scorers using all possible combinations of the options for the four components, we will end up with 36 scorers, whose results are shown in Table 2.

A few points about these results deserve mention. First, using z-score normalization yields far better results than using min-max normalization: regardless of how the remaining three components are implemented, we see a significant increase of 0.06-0.10 points in AVG QWK scores when min-max normalization is replaced with z-score normalization. To our knowledge, no one has quantified the impact of different feature normalization methods on cross-prompt scoring performance.

Second, using the ADVANCED features (computed using Gemma3 or Qwen3) yields far better results than using the Basic features. Specifically, regardless of how the remaining three components are implemented, we see a statistically significant increase of 0.06-0.10 points in AVG scores when the BASIC features are replaced with the ADVANCED features. While it is perhaps not surprising that better scores can be obtained using a richer set of features, what is somewhat unexpected is the consistently large performance improvements obtained with the rich features.

Third, contrary to common wisdom, JOINT models do not always outperform their INDEPENDENT counterparts. Specifically, keeping the feature normalization method and the development set metric unchanged, INDEPENDENT performs at the same level as or significantly better than its JOINT counterparts when the BASIC feature set is used. In contrast, when the ADVANCED feature set is used, JOINT^T performs at the same level as or significantly better than the other two model architectures, again assuming that the feature normalization method and the development set metric remain unchanged. In addition, these results suggest that (1) the two JOINT models enjoy bigger performance improvements than the INDEPENDENT model when the BASIC features are replaced with the ADVANCED features, and (2) JOINT^T is generally superior to JOINT^S. We speculate that

[15] https://huggingface.co/google/gemma-2-9b-it

[16] https://huggingface.co/meta-llama/Llama-2-13b-chat-hf

[17] All significance tests are paired t-tests ($p < 0.05$).

	Setting	Overall	Content	Org	WC	SF	Conv	PA	Lang	Nar	AVG
						INDEPENDENT					
1	Loss	.639	.599	.475	.564	.592	.487	.675	.605	.693	.592±.001
2	QWK^S	.643	.606	.472	.549	.581	.484	**.690**	.605	**.694**	.592±.001
3	QWK^T	.644	**.613**	.489	.563	.596	**.495**	.686	.601	.683	.597±.004
						JOINT^S					
4	Loss	.635	.602	.487	.565	.606	.449	.671	.617	.670	.589±.005
5	QWK^S	.641	.606	.503	.568	.609	.448	.675	.619	.678	.594±.004
6	QWK^T	.638	.605	.499	.570	**.613**	**.450**	.677	.613	.670	.593±.005
						JOINT^T					
7	Loss	.639	.611	.504	.574	.606	.445	.680	.627	.686	.597±.004
8	QWK^S	.640	.612	.511	.580	.605	.446	.684	.632	.686	.599±.004
9	QWK^T	**.645**	**.613**	**.535**	**.582**	.606	**.449**	.686	**.633**	.680	**.603±.005**

Table 3: Trait-wise QWK scores for ADVANCED^G with z-score normalization.

JOINT^T's superior performance can be attributed to the fact that the learner does not have to waste any effort on learning how to score those traits that are not even present in the target essays, but additional experiments are needed to determine the reason.

Fourth, among the three development metrics we consider, if z-score normalization is used, QWK^T performs either at the same level or significantly better than the other two metrics as we vary the implementations of the feature set and the model architecture. However, if min-max normalization is used, the results are rather mixed.

Finally, comparing these results with the baseline results in Table 1, we can see that any scorer that uses z-score normalization in combination with one of the two ADVANCED feature sets can comfortably outperform the current state of the art established by ProTACT and GAPS.

3.3 Trait-Specific Results

Next, we discuss the trait results. Since we showed in the previous subsection that poor AVG results were obtained when min-max normalization and the BASIC features were used, below we will focus our discussion on the nine cross-prompt scorers where the feature normalization method is z-score and the feature set is ADVANCED^G. Trait-specific results of these nine scorers are shown in Table 3.[18]

A few points deserve mention. First, QWK^T performs at the same level or significantly better than the other development set metrics on the majority of the traits when used in combination with INDEPENDENT and JOINT^T. When JOINT^S is used, there is no clear winner even though Loss offers worse performance than the other two metrics.

Second, different models score well for different traits regardless of the development set metric being used. For instance, INDEPENDENT achieves significantly better results than the other models on NARRATIVITY and CONVENTIONS, whereas JOINT^T achieves significantly better results on LANGUAGE, WORD CHOICE, and ORGANIZATION. These results seem to suggest that different traits are best scored using different models.

Third, comparing the best-performing JOINT model (row 9) with the corresponding INDEPENDENT model (row 3), we see that not all traits benefit from joint modeling. Specifically, ORGANIZATION, WORD CHOICE, SENTENCE FLUENCY, and LANGUAGE benefited from joint modeling, CONVENTIONS was hurt by joint modeling, and the remaining traits were neither improved nor hurt.

Upon a closer look at the traits, we see that ORGANIZATION, WORD CHOICE, and SENTENCE FLUENCY are highly correlated, and the JOINT model has been able to capture the interdependencies among them. In contrast, LANGUAGE only co-occurs with four other traits (CONTENT, PROMPT ADHERENCE, NARRATIVITY, and OVERALL). It is highly correlated with NARRATIVITY but has only a moderate correlation with the other three traits. None of these traits is being improved significantly by joint modeling, so the question is: why can LANGUAGE be improved by joint modeling? A closer examination of the results reveals that fewer extreme LANGUAGE scores were predicted by the JOINT model in comparison to the INDEPENDENT model, and this in turn boosted the LANGUAGE score.

As for CONVENTIONS, we find that its correlation with other traits is relatively moderate: while in many essays the CONVENTIONS scores correlate well with other trait scores, there were also many essays where the opposite was true. Among the essays in the latter group, we saw that their scores were predicted more accurately by the INDEPENDENT model than the JOINT model.

[18] Additional experimental results are shown in Appendix G

	Basic	LLM	Rationale	**Overall**	**Content**	**Org**	**WC**	**SF**	**Conv**	**PA**	**Lang**	**Nar**	**AVG**
1	20	G3	✓	**.645**	.613	.535	**.582**	.606	.449	**.686**	**.633**	**.680**	**.603±.005**
2	20	G3	–	.629	.597	.497	.571	**.609**	.439	.651	.619	.665	.586±.005
3	20	L2	✓	.641	.618	.539	.537	.583	**.473**	.667	.591	.656	.589±.003
4	20	L2	–	.620	.587	.516	.524	.546	.454	.622	.549	.628	.561±.003
5	20	G2	✓	.625	.579	.484	.563	.594	.421	.641	.613	.658	.575±.008
6	20	G2	–	.630	.579	.518	.513	.548	.443	.608	.524	.614	.553±.003
7	20	–	–	.620	.564	.481	.476	.502	.426	.573	.491	.583	.524±.004
8	–	G3	✓	.506	.517	.323	.506	.499	.324	.643	.596	.629	.505±.008
9	–	G3	–	.457	.484	.357	.389	.447	.334	.630	.583	.624	.478±.011
10	100	G3	✓	.636	**.620**	**.542**	.557	.594	.462	.676	.621	.679	.599±.002
11	500	G3	✓	.616	.601	.525	.550	.564	.453	.653	.596	.650	.579±.003
12	1000	G3	✓	.583	.576	.482	.483	.510	.436	.617	.560	.627	.542±.004

Table 4: Trait-wise QWK scores of JOINT^T with different feature sets, z-score normalization and QWK^T.

4 Impact of Features

In our second set of experiments, we take a closer look at how different feature sets impact the performance of our best-performing scorer from the previous section, the JOINT^T model that uses the ADVANCED^G feature set in combination with z-score normalization and QWK^T. Below we refer to this model as the Baseline model.

Results of this set of experiments are shown in Table 4. Note that the feature set used in each experiment can be inferred from the first three columns of the table. Specifically, "Basic" shows the number of top statistical features that are included in the Basic feature set, "LLM" shows the LLM used to score the trait features, and "Rationales" shows whether rationales are included in the feature set. For comparison purposes, we show the results of the Baseline model in row 1.

Usefulness of rationales. To determine the usefulness of rationales, we retrain the Baseline model without rationales. Results are shown in row 2. Comparing rows 1 and 2, we see that removing the rationales causes a significant drop in AVG QWK score, showing that the rationales generated by Gemma3 are useful for cross-prompt scoring. To examine whether rationales are similarly useful when they are generated by the older LLMs, we repeat the experiments in rows 1 and 2 by retraining the scorer using LLM features and rationales that are computed by Llama2 and Gemma2. Results are shown in rows 3–6. As can be seen, removing rationales still results in a significant drop in AVG score, confirming the usefulness of rationales for cross-prompt scoring when used as features.

Usefulness of LLM-scored trait features. To determine whether the Gemma3-scored trait features are useful in the *absence* of rationales, we retrain the model that produced the results in row 2 *without* using any LLM-scored trait features. Results are shown in row 7. Comparing rows 2 and 7, we see that AVG results drop significantly when the Gemma3-scored trait features are removed from the feature set, thus confirming their usefulness.

Usefulness of statistical features. Next, to determine the usefulness of the statistical features (those in the Basic feature set), we retrain the Baseline model without the statistical features. Results are shown in row 8. Comparing rows 1 and 8, we see that AVG QWK scores drop significantly, suggesting the usefulness of the statistical features.

Usefulness of rationales in the absence of statistical features. To determine whether rationales are still useful in the *absence* of the statistical features, we retrain the model that produced the results in row 8 *without* using rationales. Results are shown in row 9. Comparing rows 8 and 9, we see that AVG QWK score drops significantly, suggesting that rationales remain useful for cross-prompt scoring even when statistical features are not used.

Impact of the number of statistical features. Finally, we examine how scoring performance changes by *increasing* the number of statistical features. Specifically, we retrain the Baseline model by using the top 100, 500, and 1000 statistical features. Results of these experiments are shown in rows 10–12. Comparing them with the results in row 1, we see that AVG QWK drops significantly as we increase the number of top statistical features, implying that the size of the statistical feature set has a significant impact on AVG performance.

5 Exploiting Target Essays

In this section, we describe our third set of experiments, which examine whether cross-prompt scorers can be improved if we use the information extracted from the target essays to re-weigh the training instances for model training. All the experiments in this section are conducted using

	Setting	Overall	Content	Org	WC	SF	Conv	PA	Lang	Nar	AVG
						INDEPENDENT					
1	Baseline	.645	.624	.501	.541	.597	**.496**	.697	.601	.693	.599±.002
2	Setting 1	.644	.622	.504	.552	.601	.483	.692	.602	.692	.599±.003
3	Setting 2	.635	.626	.505	.551	**.619**	.485	**.712**	.604	.716	**.606±.003**
4	Setting 3	**.646**	**.630**	.493	.550	.615	.486	.709	.605	**.718**	**.606±.003**
						JOINT^T					
5	Baseline	.645	.613	**.535**	.582	.606	.449	.686	.633	.680	.603±.005
6	Setting 1	.634	.610	.498	.552	.587	.434	.689	.630	.687	.591±.006
7	Setting 2	.636	.609	.526	**.586**	.587	.468	.692	**.635**	.691	.603±.005
8	Setting 3	.632	.612	.526	**.586**	.596	.460	.694	.634	.692	.604±.006

Table 5: Trait-wise results of exploiting the target-prompt essays for re-weighting training instances (Baseline: the scorer that has achieved the highest Average QWK score in Table 2; Setting 1: same essay type; Setting 2: same essay type, kNN; Setting 3: all training set, kNN).

the best-performing INDEPENDENT model and the best-performing JOINT model from the first set of experiments. These two scorers will also serve as our baselines in this set of experiments.

Setting 1: same essay type. In this setting, all essays in the training set that have the same essay type as the target prompt are assigned a higher weight t during the training loss calculation; all other training instances have a weight of 1. The intuition is that these essays are more similar to the target prompt than essays of other types, so the model should place greater emphasis on predicting them accurately. The value of $t \in \{2, 5, 10, 50, 100, 1000\}$ is selected via a grid search based on development set performance.

Setting 2: same essay type, kNN. For each essay in the target prompt, we select the top-k most similar training instances that have the same essay type as the target prompt based on the cosine similarity of the ADVANCED features.[19] If a training instance appears among the top-k neighbors for T test instances, its weight in the training loss calculation is set to $(1+x)^T$, where x is a hyperparameter. The value of $x \in \{0.05, 0.075, 0.1, 0.125, 0.15\}$ and $k \in \{5, 10, 20\}$ is selected via a grid search based on development set performance.

Setting 3: all training set, kNN. This setting is the same as Setting 2, except that the top-k most similar training instances are selected from the entire training set, rather than only from those having the same essay type as the target prompt.

Note that in Setting 1, we only make use of a piece of shallow information, the *type* of the target essays, and do not exploit information from the target essays for model training at all. Hence, this setting can be viewed as non-transductive.

Results of these experiments are shown in Table 5.[20] For comparison purposes, we show in rows 1 and 5 the results of our baselines, the best-performing INDEPENDENT model and JOINT model from the first set of experiments.

A few points deserve mention. First, comparing row 1 with row 2 and row 5 with row 6, we see that simply increasing the weight of all training instances that have the same essay type as the target essays does not yield statistically significant improvements over the baseline.

Second, comparing rows 3 and 4 (the two transductive settings) with rows 1 and 2 (the non-transductive settings), we can see that exploiting information from the target essays yields significant improvements in the AVG score for the INDEPENDENT model. A closer inspection of the trait results, however, reveals that transduction does not always yield better results. For example, with the INDEPENDENT model, the QWK scores for ORGANIZATION and CONVENTIONS are significantly worse after transduction.

Third, for both models, the two transductive settings are statistically indistinguishable in AVG.

Fourth, the best AVG QWK score in this set of experiments is achieved when the INDEPENDENT model is used in combination with the transductive settings. This is perhaps one of the most interesting observations, as conventional wisdom suggests that JOINT models in general would outperform their INDEPENDENT counterparts. In this case, we speculate that *simplicity* wins: when the source essays are re-weighted using the target essays, the score distribution of the training set changes, and the simplicity of the INDEPENDENT model allows it to more easily optimize for the new score distribution. For the JOINT model, even though the source essays are re-weighted in the same way as

[19] Results of using only rationale embeddings for selecting the nearest neighbors can be found in Appendix H.

[20] Prompt-wise results can be found in Appendix I.

those in the INDEPENDENT model, the complexity of the learning task that the JOINT model faces may have made it more difficult to optimize for the new score distribution.

Fifth, while the best AVG score is achieved by the INDEPENDENT model, this best-performing model does not perform the best w.r.t. all the trait scores. For instance, the QWK scores for LANGUAGE and WORD CHOICE are significantly better for the JOINT model than the INDEPENDENT model, and the two models perform statistically indistinguishably w.r.t. ORGANIZATION. These results suggest that when transductive learning is used for cross-prompt scoring, it may be beneficial to use different models for scoring different traits.

Finally, we note that these are only some of the possible ways to exploit information extracted from the target essays. Our investigation only represents a starting point for using transduction for cross-prompt scoring, but the results we have obtained so far suggest that transduction is a promising avenue for future research.

6 Related Work

Cross-prompt scoring. Research on cross-prompt scoring is still in its infancy. Ridley et al. (2021) propose the first model that explores cross-prompt multi-trait AES, laying the foundation for subsequent research in this area by scoring an essay using two types of information: (1) a representation learned from the input essay when it is represented as a sequence of part-of-speech embeddings and (2) a set of prompt-independent hand-crafted features.

While existing cross-prompt scorers rely on the two sources of information identified by Ridley et al. (2021) for scoring, recent work has focused on the first source, aiming to develop methods to learn better essay representations (e.g., Chen and Li (2023), Dong et al. (2017), Cao et al. (2020)). While early cross-prompt scorers largely ignore the essay prompt, recent work has begun to exploit prompt information (e.g., Jiang et al. (2023), Do et al. (2025), Xu et al. (2025), Chen et al. (2025)).[21]

In contrast, only a handful of researchers have focused on feature development. Li and Ng (2024) show that state-of-the-art cross-prompt AES results can be achieved by using a feature-selected set of prompt-independent features in combination with a simple neural network, demonstrating the values that these features can deliver for cross-prompt AES. Eltanbouly et al. (2025) use LLMs to induce new features from scoring rubrics. Our work focuses largely on the latter category: the cross-prompt scorers we study in this paper are largely feature-based, and are augmented with a representation defined by the LLM-generated rationales.

Trait scoring. The development of the ASAP++ corpus has facilitated the training and evaluation of multi-trait scoring models that capture the interaction among different essay traits during the scoring process (Kumar et al., 2022; Shibata and Uto, 2022; He et al., 2022; Do et al., 2023; Chen and Li, 2024; Wang and Liu, 2025). Historically, however, research has focused on *single-trait* scoring, where the traits are scored independently of each other using either rule-based or learning-based methods. In particular, systems have been developed for scoring COHERENCE (Higgins et al., 2004; Somasundaran et al., 2014; Wu et al., 2023), ORGANIZATION (Persing et al., 2010), THESIS CLARITY (Persing and Ng, 2013), PROMPT ADHERENCE (Persing and Ng, 2014; Zhuang et al., 2024), ARGUMENT PERSUASIVENESS (Persing and Ng, 2015; Carlile et al., 2018), STYLE (Mathias and Bhattacharyya, 2018b), and THESIS STRENGTH (Ke et al., 2019).

7 Conclusion

Given that little analysis has been performed for cross-prompt essay scorers, we conducted the first empirical study on how the four key components of a cross-prompt scorer (feature set, model architecture, feature normalization, development set metric) interact with each other and impact performance. We also examined for the first time the applicability of transduction to cross-prompt scoring, showing that the use of information extracted from target-prompt essays is a promising way to improve a state-of-the-art cross-prompt scorer for use in the rarely-studied classroom setting.

In future work, we plan to explore two directions. First, we plan to determine the generalizability of our component interaction findings by validating them on additional essay scoring datasets, such as ELLIPSE (Crossley et al., 2023) and PERSUADE 2.0 (Crossley et al., 2024). Second, while our initial results on using transductive learning for cross-prompt AES hold promise, we plan to explore other transductive techniques beyond instance re-weighting, such as consistency regularization and graph-based methods, which could potentially yield larger gains.

[21] See Appendix J for a detailed description of these scorers.

Limitations

We discuss two limitations of our work. First, while we conducted a rare empirical analysis of how four key components of a cross-prompt scorer interact with each other and impact performance, this is by no means an exhaustive investigation. A cross-prompt essay scorer is complex, containing many fine-grained details. There could be various less-investigated components that can have a big impact on scoring performance but which we have not investigated in this study. Second, our investigation of the application of transductive learning to cross-prompt scoring is by no means exhaustive. In fact, we consider our investigation preliminary, as we have only examined one of numerous ways in which the information extracted from the target essays can be used to improve a cross-prompt scorer.

References

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*, pages 1011–1020.

Winston Carlile, Nishant Gurrapadi, Zixuan Ke, and Vincent Ng. 2018. Give me more feedback: Annotating argument persuasiveness and related attributes in student essays. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 621–631, Melbourne, Australia. Association for Computational Linguistics.

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.

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: NAACL 2025*, pages 5811–5829, Albuquerque, New Mexico. Association for Computational Linguistics.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, Luke Marris, Sam Petulla, Colin Gaffney, Asaf Aharoni, Nathan Lintz, Tiago Cardal Pais, Henrik Jacobsson, Idan Szpektor, Nan-Jiang Jiang, and 3416 others. 2025. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *Preprint*, arXiv:2507.06261.

Scott Crossley, Perpetual Baffour, Tian Yu, Alex Franklin, Meg Benner, and Ulrich Boser. 2024. A large-scale corpus for assessing written argumentation: PERSUADE 2.0. *Assessing Writing*, 61:100865.

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.

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.

Heejin Do, Taehee Park, Sangwon Ryu, and Gary Lee. 2025. Towards prompt generalization: Grammar-aware cross-prompt automated essay scoring. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 2818–2824, Albuquerque, New Mexico. Association for Computational Linguistics.

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 (CoNLL 2017)*, pages 153–162, Vancouver, 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.

Yaqiong He, Feng Jiang, Xiaomin Chu, and Peifeng Li. 2022. Automated Chinese essay scoring from multiple traits. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 3007–3016, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Derrick Higgins, Jill Burstein, Daniel Marcu, and Claudia Gentile. 2004. Evaluating multiple aspects of coherence in student essays. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics: HLT-NAACL 2004*, pages 185–192, Boston, Massachusetts, USA. Association for Computational Linguistics.

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).

Zhiwei Jiang, Tianyi Gao, Yafeng Yin, Meng Liu, Hua Yu, Zifeng Cheng, and Qing Gu. 2023. Improving domain generalization for prompt-aware essay scoring via disentangled representation learning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12456–12470, Toronto, Canada. Association for Computational Linguistics.

Zixuan Ke, Hrishikesh Inamdar, Hui Lin, and Vincent Ng. 2019. Give me more feedback II: Annotating thesis strength and related attributes in student essays. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3994–4004, Florence, Italy. Association for Computational Linguistics.

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.

Sanwoo Lee, Yida Cai, Desong Meng, Ziyang Wang, and Yunfang Wu. 2024. Unleashing large language models' proficiency in zero-shot essay scoring. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 181–198, Miami, Florida, USA. Association for Computational Linguistics.

Shengjie Li and Vincent Ng. 2024. 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.

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 (LREC-COLING 2024)*, pages 2777–2786, Torino, Italia. ELRA and ICCL.

Sandeep Mathias and Pushpak Bhattacharyya. 2018a. 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)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Sandeep Mathias and Pushpak Bhattacharyya. 2018b. Thank "goodness"! a way to measure style in student essays. In *Proceedings of the 5th Workshop on Natural Language Processing Techniques for Educational Applications*, pages 35–41, Melbourne, Australia. Association for Computational Linguistics.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Isaac Persing, Alan Davis, and Vincent Ng. 2010. Modeling organization in student essays. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 229–239, Cambridge, MA. Association for Computational Linguistics.

Isaac Persing and Vincent Ng. 2013. Modeling thesis clarity in student essays. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 260–269, Sofia, Bulgaria. Association for Computational Linguistics.

Isaac Persing and Vincent Ng. 2014. Modeling prompt adherence in student essays. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1534–1543, Baltimore, Maryland. Association for Computational Linguistics.

Isaac Persing and Vincent Ng. 2015. Modeling argument strength in student essays. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 543–552, Beijing, China. Association for Computational Linguistics.

Robert Ridley, Liang He, Xin-yu Dai, Shujian Huang, and Jiajun Chen. 2021. Automated cross-prompt scoring of essay traits. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(15):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*, abs/2008.01441.

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, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Swapna Somasundaran, Jill Burstein, and Martin Chodorow. 2014. Lexical chaining for measuring discourse coherence quality in test-taker essays. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 950–961, Dublin, Ireland. Dublin City University and Association for Computational Linguistics.

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, Austin, Texas. Association for Computational Linguistics.

Masaki Uto, Yikuan Xie, and Maomi Ueno. 2020. Neural automated essay scoring incorporating handcrafted features. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6077–6088, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Vladimir N. Vapnik. 1998. *Statistical Learning Theory*. Wiley, New York.

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, Abu Dhabi, UAE. Association for Computational Linguistics.

Hongyi Wu, Xinshu Shen, Man Lan, Shaoguang Mao, Xiaopeng Bai, and Yuanbin Wu. 2023. A multi-task dataset for assessing discourse coherence in Chinese essays: Structure, theme, and logic analysis. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 6673–6688, Singapore. Association for Computational Linguistics.

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.

Xinlin Zhuang, Hongyi Wu, Xinshu Shen, Peimin Yu, Gaowei Yi, Xinhao Chen, Tu Hu, Yang Chen, Yupei Ren, Yadong Zhang, Youqi Song, Binxuan Liu, and Man Lan. 2024. TOREE: Evaluating topic relevance of student essays for Chinese primary and middle school education. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 5749–5765, Bangkok, Thailand. Association for Computational Linguistics.

A Details of ASAP/ASAP++

A.1 Writing Prompts

Table 6 shows the eight essay prompts in ASAP along with the corresponding number of essays and the average word count for each prompt. The applicable set of traits for each prompt as well as their corresponding score range(s) are shown in Table 7.

A.2 Trait Definitions

Below is the definition of each trait.

- OVERALL: how good the overall quality is.
- CONTENT: how clear and focused the writing is and how well-developed the main ideas are.
- PROMPT ADHERENCE: how adherent the essay is to the prompt.
- LANGUAGE: how good grammar and spelling are.
- NARRATIVITY: how coherent and cohesive the response is.
- ORGANIZATION: how well-organized the essay is.
- WORD CHOICE: how well the words convey the intended message.
- SENTENCE FLUENCY: whether the sentences in the essay are of high quality.
- CONVENTIONS: how well the essay demonstrates standard writing conventions.

B Do et al.'s (2023) Data Partitions

Table 8 shows the sizes of the training set, the development set, and the test set for each essay prompt in ASAP.

C Li and Ng's (2024) Features

In this section, we list the 1535 features employed by Li and Ng (2024), many of which are borrowed from those used by Ridley et al. (2020) and Uto et al. (2020), as discussed below.

Table 9 enumerates all features alongside their detailed descriptions, categorizing them into distinct groups for enhanced clarity. Feature names are appended with superscripts for source identification: [1] denotes features derived using the textstat

	Prompt	Avg. # Words	# Essays
1	Write a letter to the editor of a newspaper about how computers affect society today.	365.4	1783
2	Write a letter to the editor of a newspaper about censorship in libraries	380.7	1800
3	Write a review about an article called Rough Road Ahead by Joe Kurmaskie. The article will be provided.	108.5	1726
4	Explain why the author concludes the story the way the author did. The short story will be provided.	94.3	1772
5	Describe the mood created by the author in the memoir. Support your answer with relevant and specific information from the memoir	122.1	1805
6	Describe the difficulties that builders of the Empire State Building faced because of allowing dirigibles to dock there.	153.2	1800
7	Write a story about a time when you were patient OR write a story about a time when someone you know was patient OR write a story in your own way about patience.	167.6	1569
8	We all understand the benefits of laughter. For example, someone once said, "Laughter is the shortest distance between two people." Many other people believe that laughter is an important part of any relationship. Tell a true story in which laughter was one element or part.	604.7	723
Overall		222.5	12978

Table 6: The eight writing prompts in ASAP.

Prompt	Traits	Score Range(s)
1	Overall, Cont, WC, Org, SF, Conv	OVERALL: [2, 12]; Other: [1, 6]
2	Overall, Cont, WC, Org, SF, Conv	OVERALL: [0, 6]; Other: [1, 6]
3	Overall, Cont, PA, Nar, Lang	[0, 3]
4	Overall, Cont, PA, Nar, Lang	[0, 3]
5	Overall, Cont, PA, Nar, Lang	[0, 3]
6	Overall, Cont, PA, Nar, Lang	[0, 3]
7	Overall, Cont, Org, Conv	OVERALL: [0, 30]; Other: [0, 6]
8	Overall, Cont, WC, Org, SF, Conv	OVERALL: [0, 60]; Other: [1, 12]

Table 7: Statistics for the combined ASAP and ASAP++ dataset. Trait names are abbreviated as follows: Cont: CONTENT, Org: ORGANIZATION, WC: WORD CHOICE, SF: SENTENCE FLUENCY, Conv: CONVENTIONS, PA: PROMPT ADHERENCE, Lang: LANGUAGE, Nar: NARRATIVITY.

Prompt	Traits	Score Range(s)	
Prompt	Training set size	Dev set size	Test set size
1	9513	1680	1783
2	9499	1679	1798
3	9561	1689	1726
4	9522	1682	1772
5	9494	1677	1805
6	9498	1678	1800
7	9695	1712	1569
8	10414	1839	723

Table 8: Statistics on Do et al.'s (2023) data partitions for the ASAP essay prompts.

package[22], [2] for those from the readability package[23], [3] for NLTK package-derived features[24], and [4] for features obtained via the spaCy package[25].

[22] https://github.com/textstat/textstat
[23] https://github.com/andreasvc/readability
[24] https://www.nltk.org/
[25] https://spacy.io/

Feature Group	Feature Name	Description
	Ridley et al.'s (2020) Features (86 features)	
LB^R	word_count	The total number of words in the essay.
	mean_word	The average number of characters in each word.
	ess_char_len	The number of characters in the essay.
	mean_sent[3]	The average number of words in each sentence.
	characters_per_word[2]	The average number of characters in each word.
	avg_word_len	The average number of characters in each word.
	avg_words_per_sentence	The average number of words in each sentence.
	characters[2]	The number of characters in the essay.
	syllables[2]	The number of syllables in the essay.
	words[2]	The number of words in the essay.
	words_per_sentence[2]	The average number of words in each sentence.
	sentences_per_paragraph[2]	The average number of sentences in each paragraph.
	.[3]	The number of periods in the essay.
	,[3]	The number of commas in the essay.
	syll_per_word[2]	The average number of syllables in each word.
RB^R	automated_readability[1]	A readability metric that measures the readability of a text based on characters per word and words per sentence.
	linsear_write[1]	A readability metric developed for the U.S. Air Force to help them calculate the understandability of technical manuals, factoring in sentence length and words that are considered difficult.
	Kincaid[2]	A readability metric which estimate the readability of English texts based on sentence length and word length.
	ARI[2]	A readability metric that measures the readability of a text based on characters per word and words per sentence.
	Coleman-Liau[2]	A readability assessment that estimates the U.S. grade level required to understand a piece of text based on characters, words, and sentences.
	FleschReadingEase[2]	A readability metric that measures the readability of text based on syllables, words, and sentences. The scores are on a scale from 0 to 100, with higher scores indicating easier-to-read text.
	GunningFogIndex[2]	A readability metric that estimates the years of formal education a person needs to understand the text on the first reading.
	LIX[2]	A readability metric that considers sentence length and the percentage of long words (words with more than six characters) in a text.
	SMOGIndex[2]	A readability formula that estimates the education level needed to understand a piece of text by analyzing the number of polysyllabic words (words with three or more syllables) within the text.
	RIX[2]	A variant of the LIX readability index that only takes into account the average number of long words per sentence.
	DaleChallIndex[2]	A readability formula that uses word difficulty based on a list of familiar words, along with sentence length, to estimate the grade level required to understand a text.
	sentences[2]	The total number of sentences present in the essay.
	paragraphs[2]	The total number of paragraphs present in the essay.
	long_words[2]	The number of words that have 7 or more characters.
	complex_words[2]	The number of words that have 3 or more syllables.
	complex_words_dc[2]	The total number of words that are not in the Dale-Chall word list of 3000 words recognized by 80% of fifth graders.
TC^R	clause_per_s[4]	The average number of clauses per sentence.
	sent_ave_depth[4]	The average parse tree depth per sentence in each essay,
	ave_leaf_depth[4]	The average parse depth of each leaf node in the parse tree.
	max_clause_in_s[4]	The maximum number of clauses in the sentences of the essay.
	mean_clause_l[4]	The average number of words in each clause.
SB^R	overall_positivity_score[3]	Overall, how positive the essay is.
	overall_negativity_score[3]	Overall, how negative the essay is.

Continued on next page

Feature Group	Feature Name	Description
	positive_sentence_prop[3]	The percentage of positive sentences in the essay.
	neutral_sentence_prop[3]	The percentage of neutral sentences in the essay.
	negative_sentence_prop[3]	The percentage of negative sentences in the essay.
TV^R	sent_var[3]	The variance of the length of sentences in the essay.
	word_var[3]	The variance of the length of words in the essay.
	stop_prop	The percentage of stopwords in the essay.
	unique_word	The total number of unique words in the essay.
	type_token_ratio[2]	The number of unique words divided by the number of words.
	wordtypes[2]	The total number of unique words present in the essay.
	tobeverb[2]	The number of "to be" verbs in the essay.
	auxverb[2]	The number of auxilllary verbs in the essay.
	conjunction[2]	The number of conjunctions in the essay.
	pronoun[2]	The number of pronouns in the essay
	preposition[2]	The number of prepositions in the essay
	nominalization[2]	The number of nominalizations in the essay
	begin_w_pronoun[2]	The number of sentences in the essay that begin with a pronoun.
	begin_w_interrogative[2]	The number of sentences in the essay that begin with an interrogative.
	begin_w_article[2]	The number of sentences in the essay that begin with an article.
	begin_w_subordination[2]	The number of sentences in the essay that begin with a subordination.
	begin_w_conjunction[2]	The number of sentences in the essay that begin with a conjunction.
	begin_w_preposition[2]	The number of sentences in the essay that begin with a preposition.
	spelling_err[3]	The number of words that are not in The Brown corpus of the NLTK package.
	prep_comma[3]	The number of preprositions and commas in the essay.
	MD[3]	The number of tokens having a POS tag of MD in the text.
	DT[3]	The number of tokens having a POS tag of DT in the text.
	TO[3]	The number of tokens having a POS tag of TO in the text.
	PRP$[3]	The number of tokens having a POS tag of PRP$ in the text.
	JJR[3]	The number of tokens having a POS tag of JJR in the text.
	WDT[3]	The number of tokens having a POS tag of WDT in the text.
	VBD[3]	The number of tokens having a POS tag of VBD in the text.
	WP[3]	The number of tokens having a POS tag of WP in the text.
	VBG[3]	The number of tokens having a POS tag of VBG in the text.
	RBR[3]	The number of tokens having a POS tag of RBR in the text.
	CC[3]	The number of tokens having a POS tag of CC in the text.
	VBP[3]	The number of tokens having a POS tag of VBP in the text.
	JJS[3]	The number of tokens having a POS tag of JJS in the text.
	VBN[3]	The number of tokens having a POS tag of VBN in the text.
	POS[3]	The number of tokens having a POS tag of POS in the text.
	NNS[3]	The number of tokens having a POS tag of NNS in the text.
	WRB[3]	The number of tokens having a POS tag of WRB in the text.
	JJ[3]	The number of tokens having a POS tag of JJ in the text.

Continued on next page

Feature Group	Feature Name	Description
	CD[3]	The number of tokens having a POS tag of CD in the text.
	NNP[3]	The number of tokens having a POS tag of NNP in the text.
	RP[3]	The number of tokens having a POS tag of RP in the text.
	RB[3]	The number of tokens having a POS tag of RB in the text.
	IN[3]	The number of tokens having a POS tag of IN in the text.
	VB[3]	The number of tokens having a POS tag of VB in the text.
	VBZ[3]	The number of tokens having a POS tag of VBZ in the text.
	NN[3]	The number of tokens having a POS tag of NN in the text.
	PRP[3]	The number of tokens having a POS tag of PRP in the text.
	Uto et al.'s (2020) Features (25 features)	
LB^U	syllable_count	The number of syllables in the essay.
	num_words	The number of words in the essay.
	num_sentences	The number of sentences in the essay.
	lemma_count	The number of lemmas in the essay.
	,	The number of commas in the essay.
	!	The number of exclamation marks in the essay.
	?	The number of question marks in the essay.
TV^U	noun_count	The number of nouns in the essay.
	verb_count	The number of verbs in the essay.
	adverb_count	The number of adverbs in the essay.
	adjective_count	The number of adjectives in the essay.
	conjunction_count	The number of conjunctions in the essay.
	spelling_error_count	The number of spelling errors in the essay.
	stopwords_count	The number of stop words in the essay.
RB^U	ARI	A readability metric that measures the readability of a text based on characters per word and words per sentence.
	coleman_liau	A readability assessment that estimates the U.S. grade level required to understand a piece of text based on characters, words, and sentences.
	dale_chall	A readability formula that uses word difficulty based on a list of familiar words, along with sentence length, to estimate the grade level required to understand a text.
	difficult_words	The total number of words that are not in the Dale-Chall word list of 3000 words recognized by 80% of fifth graders.
	flesch_reading_ease	A readability metric that measures the readability of text based on syllables, words, and sentences. The scores are on a scale from 0 to 100, with higher scores indicating easier-to-read text.
	flesch_kincaid_grade	A readability metric which estimate the readability of English texts based on sentence length and word length.
	gunning_fog	A readability metric that estimates the years of formal education a person needs to understand the text on the first reading.
	linsear_write	A readability metric developed for the U.S. Air Force to help them calculate the understandability of technical manuals, factoring in sentence length and words that are considered difficult.
	smog_index	A readability formula that estimates the education level needed to understand a piece of text by analyzing the number of polysyllabic words (words with three or more syllables) within the text.
	Part-of-speech Bigram Features (902 features)	
POSB	(DT, NN)	The number of appearance of the bigram (DT, NN)
	...	
	Pronoun Features (218 features)	
PRO-Pronoun Count	pronoun_cnt_I	The number of pronoun "I" in the essay.
	...	

Continued on next page

Feature Group	Feature Name	Description
PRO-Pronoun Group Count	first_person_pronoun_cnt ...	The number of first person pronouns in the essay.
PRO-Sent Pronoun	sent_cnt_I ...	The number of sentences that contain "I"
PRO-Sent Pronoun Group	sent_first_person_pronoun ...	The number of sentences that contain first person pronouns.
PRO-Sent Pronoun Portion	percentage_sent_I ...	The percentage of sentences that contain pronoun "I".
PRO-Sent Pronoun Group Portion	percentage_sent_first_person ...	The percentage of sentences that contain first person pronouns.
Prompt Adherence Features (4 features)		
PA	max_sentence_dot_score	Dot score between the embeddings of an essay and its prompt.
	mean_sentence_dot_score	The maximum dot score between the embeddings of sentences of an essay and its prompt.
	min_sentence_dot_score	The average dot score between the embeddings of sentences of an essay and its prompt.
	dot_score	The minimum dot score between the embeddings of sentences of an essay and its prompt.
Top-N Words Features (300 features)		
TNW-Word Count	top_n_word_count_the ...	The count of "the" in the essay.
TNW-Sent Count	top_n_num_sent_have_the ...	The number of sentences in an essay that contains "the".
TNW-Sent Portion	top_n_percentage_sent_have_the ...	The percentage of sentences in an essay that contains "the".

Table 9: Description of the features along with their group information. Features marked with the superscript R are Ridley et al.'s (2020) features. Features marked with the superscript U are Uto et al.'s (2020) features. Group LB is composed of length-based features. Group RB is composed of readability-based features. Group TC is composed of text complexity features. Group TV is composed of text variation features. Group SB is composed of sentiment-based features. Group POSB is composed of the part-of-speech bigram features. Group PRO is composed of the pronoun-related features. Group PA is composed of the prompt adherence features. Group TNW is composed of the top-N words features.

As an expert essay grader, your task is to grade a student's essay written in response to a specific prompt. The essay and prompt are provided within "<essay></essay>" and "<essay_prompt></essay_prompt>" tags, respectively. Your grading must be strictly based on the rubric provided below. Do not apply personal interpretations or expectations beyond what is outlined in the rubric.

Rubrics for Evaluation:
<rubric>
$rubric
</rubric>

This is the prompt that the student's essay responds to:
<essay_prompt>
$essay_prompt
</essay_prompt>

Here is the student's essay that you need to assign scores to:
<essay>
$essay
</essay>

Provide your response in the following JSON format:
{ "rationale": "<rationale>", "score": "<score>" }
Replace <rationale> with a brief justification of how the essay aligns with the rubric(s), and <score> with an integer score between $lowest_score and $highest_score inclusive.

Do NOT write anything before or after the JSON object.

Figure 1: Prompt template.

D Prompt Template

Figure 1 shows the prompt template we use to instruct an LLM to score a trait and generate a rationale for it. As can be seen, the prompt template includes task instructions, the scoring rubric, the essay prompt, the essay, and finally the instructions on how the output should be formatted.

Response is generated using the recommended hyperparameters from the LLM authors. For Qwen3, we use temperature=0.7, top_p=0.8, top_k=20. For Gemma3, we use temperature=1.0, top_p=0.95, top_k=64, repetition_penalty=1.0, min_p=0.01. For both Llama2 and Gemma2, we use temperature=1.0, top_k=50, top_p=0.9.

E Best Found Hyperparameters

The best hyperparameters are:

- learning rate = 0.001.
- best values of t are shown in Table 10.
- $k = 10$.
- $x = 0.1$.

F Baseline Systems

In this section, we briefly describe each of the baseline systems used in our study.

ProTACT (Do et al., 2023) is a cross-prompt trait scoring model. The model extracts essay representations by employing CNNs and LSTMs on POS embeddings of input essays. It also extracts prompt representations by applying the same network architecture to the sum of POS embeddings and GloVe embeddings (Pennington et al., 2014) for each word in the prompt. Prompt-aware essay representations are then derived using multi-head attention, with the prompt representations acting as the query and the essay representations as the key and value. These representations are subsequently concatenated with hand-crafted features and processed through a linear layer for predicting both overall and trait-specific scores.

GAPS (Do et al., 2025) is the current state-of-the-art trait scoring model. For an input essay, the model employs an LLM to correct grammatical errors in the input essay, and then uses the same text encoder as ProTACT to obtain an embedding for the original essay and an embedding for the corrected essay. Then, the model uses a multi-head attention mechanism to obtain the final embedding, allowing for knowledge sharing between the original essay and the corrected essay. Lastly, GAPS uses the same regression head as the ProTACT model.

G Additional Experimental Results

In this section, we report the experimental results that are not shown in the main text due to space limitations.

We report the complete trait-wise results for Table 2 in Table 11.

As the counterpart to the results reported in Table 4, we report the results for INDEPENDENT with different feature sets when z-score normalization and QWK^T are used in Table 12.

As the counterpart to the results reported in Table 3, we report the results for:

- the BASIC, z-score normalization setting in Table 13;
- the ADVANCED^Q, z-score normalization setting in Table 14; and
- the min-max normalization setting for BASIC, ADVANCED^Q, ADVANCED^G feature sets in Tables 15 to 17 respectively.

Prompt	**Overall**	**Content**	**Org**	**WC**	**SF**	**Conv**	**PA**	**Lang**	**Nar**
1	2	2	10	10	2	5	–	–	–
2	2	2	2	5	10	10	–	–	–
3	2	5	–	–	–	–	5	2	5
4	2	50	–	–	–	–	10	5	5
5	5	5	–	–	–	–	5	2	2
6	2	2	–	–	–	–	100	2	50
7	2	2	2	–	–	5	–	–	–
8	2	2	5	5	5	100	–	–	–

Table 10: Best values of t for each prompt and each trait.

H Exploiting Target Essays: Using only Rationales as Features

Recall that in our transductive learning experiments, for each essay in the target prompt, we select the top-k most similar training instances that have the same essay type as the target prompt based on the cosine similarity of the ADVANCED^G feature set, which comprises the rationale embedding. A natural question is: will the trait scores be completely drowned out by the high-dimensional rationale embedding?

To answer this question, we repeat the Setting 2 and Setting 3 experiments in Table 19 by using only the rationale embedding as features. As can be seen from this table, there is a consistent precipitation in the QWK scores across the board when only the rationale embedding is used as features. These results provide suggestive evidence that the trait scores are contributing positively in transductive learning when used in combination with the high-dimensional rationale embedding.

I Exploiting Target Essays: Prompt-wise Scoring Results

In Table 18, we report the prompt-wise results for our third set of experiments, as well as the prompt-wise results of ProTACT and GAPS.

J Other Cross-Prompt Scorers

Several cross-prompt scorers other than ProTACT and GAPS have been developed in the past few years, including:

Hi att (Dong et al., 2017) is a holistic scoring model that first employs a CNN on the input characters with both max pooling and average pooling to obtain the word embeddings. Then, another CNN layer with attention pooling is applied on the word embeddings for extracting sentence representations. After that, a LSTM network with attention pooling is applied to the resulting sentence representations to obtain the document representation. Finally, a linear layer with a sigmoid output neuron is used to predict the holistic score.

AES aug (Hussein et al., 2020) builds on Taghipour and Ng's (2016) AES model by increasing the number of output neurons, with the goal of jointly predicting the trait scores and the holistic score. More specifically, each output neuron is used to predict either the holistic score or one of the trait scores. AES aug utilizes a CNN to extract n-gram level features, passes them through an LSTM network, performs mean pooling, and then uses a linear layer for joint holistic and trait scoring.

PAES (Ridley et al., 2020) is a holistic scoring model that is structurally similar to Hi att. A notable distinction of PAES is that its CNN layer is applied on top of POS tags instead of words or characters. Additionally, PAES incorporates hand-crafted features as the input of the final linear layer.

CTS (Ridley et al., 2021) is the first model that explores cross-prompt multi-trait scoring. Similar to PAES, CTS utilizes a CNN with attention pooling on the POS tags of the input essays to obtain n-gram level features. For each trait, it applies a LSTM network with attention pooling to the n-gram representations to obtain trait-specific essay representations. These representations are then combined with hand-crafted features from Ridley et al. (2020), followed by a cross-trait attention mechanism so that information can be shared by different traits. Finally, the trait scores and the holistic score are predicted using a linear layer with sigmoid activation.

PMAES (Chen and Li, 2023) performs holistic scoring both with traits and without traits. It enhances the representations extracted by Hi att by applying a contrastive learning objective to learn consistent essay representations across different prompts. This approach captures features shared by essays from different prompts, thereby helping the model generalize well across prompts. Additionally, it incorporates the hand-crafted features from Ridley et al. (2020) before feeding the representations into the final linear layer.

Features	Normalization	Setting	**Overall**	**Content**	**Org**	**WC**	**SF**	**Conv**	**PA**	**Lang**	**Nar**	**AVG**
						INDEPENDENT						
BASIC	z-score	Loss	.601	.542	.482	.484	.517	.433	.549	.487	.583	.520±.001
		QWK^S	.565	.537	.463	.470	.511	.433	.594	.503	.598	.520±.006
		QWK^T	.612	.557	.494	.490	.518	.435	.594	.506	.600	.534±.003
	min-max	Loss	.584	.463	.322	.526	.526	.201	.462	.437	.503	.447±.005
		QWK^S	.485	.459	.318	.518	.522	.196	.458	.434	.504	.433±.005
		QWK^T	.583	.462	.315	.521	.525	.195	.444	.431	.509	.443±.005
ADV.^{Q}	z-score	Loss	.643	.612	.482	.546	.593	.496	.689	.598	.699	.596±.002
		QWK^S	.641	.617	.482	.546	.581	.494	.698	.600	.698	.595±.002
		QWK^T	.645	.624	.501	.541	.597	.496	.697	.601	.693	.599±.002
	min-max	Loss	.631	.526	.397	.573	.588	.385	.581	.538	.623	.538±.002
		QWK^S	.619	.528	.391	.555	.574	.384	.585	.540	.619	.533±.001
		QWK^T	.624	.529	.393	.574	.582	.391	.601	.534	.623	.539±.002
ADV.^{G}	z-score	Loss	.639	.599	.475	.564	.592	.487	.675	.605	.693	.592±.001
		QWK^S	.643	.606	.472	.549	.581	.484	.690	.605	.694	.592±.001
		QWK^T	.644	.613	.489	.563	.596	.495	.686	.601	.683	.597±.004
	min-max	Loss	.623	.527	.388	.579	.612	.380	.605	.583	.643	.549±.003
		QWK^S	.619	.532	.386	.571	.576	.366	.607	.576	.638	.541±.003
		QWK^T	.619	.532	.392	.572	.579	.383	.611	.578	.639	.545±.003
						JOINT^S						
BASIC	z-score	Loss	.615	.550	.479	.467	.487	.407	.557	.479	.575	.513±.004
		QWK^S	.554	.548	.471	.463	.489	.405	.568	.494	.580	.508±.004
		QWK^T	.620	.554	.485	.474	.489	.409	.567	.491	.581	.519±.004
	min-max	Loss	.589	.504	.337	.489	.443	.246	.505	.474	.537	.458±.006
		QWK^S	.411	.499	.326	.482	.429	.240	.504	.467	.535	.433±.010
		QWK^T	.589	.500	.337	.486	.434	.246	.503	.463	.527	.454±.005
ADV.^{Q}	z-score	Loss	.637	.613	.512	.562	.609	.458	.684	.612	.672	.596±.003
		QWK^S	.642	.616	.517	.569	.614	.464	.682	.607	.682	.599±.006
		QWK^T	.643	.616	.514	.567	.607	.467	.687	.603	.672	.597±.004
	min-max	Loss	.614	.534	.408	.518	.583	.365	.579	.521	.567	.521±.007
		QWK^S	.600	.534	.408	.525	.578	.364	.577	.519	.564	.519±.007
		QWK^T	.611	.534	.407	.523	.581	.368	.573	.513	.559	.519±.009
ADV.^{G}	z-score	Loss	.635	.602	.487	.565	.606	.449	.671	.617	.670	.589±.005
		QWK^S	.641	.606	.503	.568	.609	.448	.675	.619	.678	.594±.004
		QWK^T	.638	.605	.499	.570	.613	.450	.677	.613	.670	.593±.005
	min-max	Loss	.600	.513	.392	.560	.587	.340	.589	.540	.573	.522±.005
		QWK^S	.590	.512	.392	.550	.568	.340	.582	.541	.570	.516±.007
		QWK^T	.597	.513	.391	.545	.575	.332	.583	.538	.568	.516±.005
						JOINT^T						
BASIC	z-score	Loss	.614	.549	.474	.454	.493	.418	.554	.481	.573	.512±.002
		QWK^S	.597	.544	.461	.454	.490	.418	.572	.489	.582	.512±.002
		QWK^T	.620	.564	.481	.476	.502	.426	.573	.491	.583	.524±.004
	min-max	Loss	.583	.478	.334	.501	.446	.234	.432	.418	.490	.435±.007
		QWK^S	.498	.479	.330	.491	.434	.231	.426	.412	.488	.421±.011
		QWK^T	.581	.477	.334	.496	.433	.234	.421	.421	.486	.431±.008
ADV.^{Q}	z-score	Loss	.635	.622	.515	.565	.595	.456	.686	.627	.682	.598±.007
		QWK^S	.638	.624	.522	.562	.597	.458	.691	.617	.684	.599±.007
		QWK^T	.640	.625	.523	.573	.596	.458	.693	.620	.675	.600±.006
	min-max	Loss	.615	.529	.397	.541	.578	.350	.584	.531	.564	.521±.007
		QWK^S	.608	.527	.396	.521	.566	.348	.581	.527	.560	.515±.008
		QWK^T	.610	.529	.398	.530	.567	.353	.585	.528	.556	.517±.011
ADV.^{G}	z-score	Loss	.639	.611	.504	.574	.606	.445	.680	.627	.686	.597±.004
		QWK^S	.640	.612	.511	.580	.605	.446	.684	.632	.686	.599±.004
		QWK^T	.645	.613	.535	.582	.606	.449	.686	.633	.680	.603±.005
	min-max	Loss	.573	.515	.352	.535	.555	.343	.605	.550	.572	.511±.007
		QWK^S	.564	.513	.352	.530	.522	.344	.601	.536	.561	.503±.004
		QWK^T	.567	.516	.350	.530	.523	.344	.596	.539	.566	.503±.006

Table 11: Complete trait-wise results for the first set of experiments along all four components. ADV.^{Q} and ADV.^{G} denote the ADVANCED^Q and ADVANCED^G feature sets respectively.

	Basic	LLM	Rationale	**Overall**	**Content**	**Org**	**WC**	**SF**	**Conv**	**PA**	**Lang**	**Nar**	**AVG**
1	20	Q3	✓	.645	.624	.501	.541	.597	.496	**.697**	**.601**	**.693**	.599±.002
2	20	Q3	–	.640	.617	**.523**	.524	.595	.489	.671	.593	.658	.590±.005
3	20	L2	✓	.639	.592	.478	.553	.578	.484	.650	.590	.661	.581±.003
4	20	L2	–	.609	.582	.502	.514	.561	.467	.615	.554	.631	.559±.002
5	20	G2	✓	.637	.575	.473	.542	.567	.475	.634	.574	.668	.572±.001
6	20	G2	–	.621	.586	.513	.509	.570	.460	.616	.535	.614	.558±.003
7	20	–	–	.612	.557	.494	.490	.518	.435	.594	.506	.600	.534±.003
8	–	Q3	✓	.530	.548	.324	.485	.522	.403	.675	.561	.633	.520±.002
9	–	Q3	–	.460	.495	.325	.351	.477	.314	.630	.564	.607	.469±.005
10	100	Q3	✓	**.651**	**.627**	.516	**.557**	**.608**	**.501**	.692	.588	.671	.601±.002
11	500	Q3	✓	.619	.610	.516	.517	.575	.490	.654	.565	.653	.578±.003
12	1000	Q3	✓	.599	.577	.489	.492	.531	.478	.621	.551	.620	.551±.003

Table 12: Results of INDEPENDENT with different feature sets when z-score normalization and QWK^T are used.

	Setting	**Overall**	**Content**	**Org**	**WC**	**SF**	**Conv**	**PA**	**Lang**	**Nar**	**AVG**
					INDEPENDENT						
1	Loss	.601	.542	.482	.484	.517	.433	.549	.487	.583	.520±.001
2	QWK^S	.565	.537	.463	.470	.511	.433	**.594**	.503	.598	.520±.006
3	QWK^T	.612	.557	**.494**	**.490**	**.518**	**.435**	**.594**	**.506**	**.600**	**.534±.003**
					JOINT^S						
4	Loss	.615	.550	.479	.467	.487	.407	.557	.479	.575	.513±.004
5	QWK^S	.554	.548	.471	.463	.489	.405	.568	.494	.580	.508±.004
6	QWK^T	**.620**	.554	.485	.474	.489	.409	.567	.491	.581	.519±.004
					JOINT^T						
7	Loss	.614	.549	.474	.454	.493	.418	.554	.481	.573	.512±.002
8	QWK^S	.597	.544	.461	.454	.490	.418	.572	.489	.582	.512±.002
9	QWK^T	**.620**	**.564**	.481	.476	.502	.426	.573	.491	.583	.524±.004

Table 13: Trait-wise QWK scores for the BASIC, z-score normalization setting.

PLAES (Chen and Li, 2024) seeks to capture general knowledge across prompts. To acquire knowledge across prompts, meta-learning techniques are used to train a model on the training prompt essays. Contrastive learning is then used to make the representation of an essay closer to its level and far away from the other levels.

These scorers were developed prior to ProTACT and GAPS and were all outperformed by them. Consequently, we did not include them in Table 1 as baseline systems. For reference, their results can be found in Table 20.

	Setting	Overall	Content	Org	WC	SF	Conv	PA	Lang	Nar	AVG
						INDEPENDENT					
1	Loss	.643	.612	.482	.546	.593	**.496**	.689	.598	**.699**	.596±.002
2	QWK^S	.641	.617	.482	.546	.581	.494	**.698**	.600	.698	.595±.002
3	QWK^T	**.645**	.624	.501	.541	.597	**.496**	.697	.601	.693	.599±.002
						JOINT^S					
4	Loss	.637	.613	.512	.562	.609	.458	.684	.612	.672	.596±.003
5	QWK^S	.642	.616	.517	.569	**.614**	.464	.682	.607	.682	.599±.006
6	QWK^T	.643	.616	.514	.567	.607	.467	.687	.603	.672	.597±.004
						JOINT^T					
7	Loss	.635	.622	.515	.565	.595	.456	.686	**.627**	.682	.598±.007
8	QWK^S	.638	.624	.522	.562	.597	.458	.691	.617	.684	.599±.007
9	QWK^T	.640	**.625**	**.523**	**.573**	.596	.458	.693	.620	.675	**.600±.006**

Table 14: Trait-wise QWK scores for the ADVANCED^Q, z-score normalization setting.

	Setting	Overall	Content	Org	WC	SF	Conv	PA	Lang	Nar	AVG
						INDEPENDENT					
1	Loss	.584	.463	.322	**.526**	**.526**	.201	.462	.437	.503	.447±.005
2	QWK^S	.485	.459	.318	.518	.522	.196	.458	.434	.504	.433±.005
3	QWK^T	.583	.462	.315	.521	.525	.195	.444	.431	.509	.443±.005
						JOINT^S					
4	Loss	**.589**	**.504**	**.337**	.489	.443	**.246**	**.505**	**.474**	**.537**	**.458**±.006
5	QWK^S	.411	.499	.326	.482	.429	.240	.504	.467	.535	.433±.010
6	QWK^T	**.589**	.500	**.337**	.486	.434	**.246**	.503	.463	.527	.454±.005
						JOINT^T					
7	Loss	.583	.478	.334	.501	.446	.234	.432	.418	.490	.435±.007
8	QWK^S	.498	.479	.330	.491	.434	.231	.426	.412	.488	.421±.011
9	QWK^T	.581	.477	.334	.496	.433	.234	.421	.421	.486	.431±.008

Table 15: Trait-wise QWK scores for the BASIC, min-max normalization setting.

	Setting	Overall	Content	Org	WC	SF	Conv	PA	Lang	Nar	AVG
						INDEPENDENT					
1	Loss	**.631**	.526	.397	.573	**.588**	.385	.581	.538	**.623**	.538±.002
2	QWK^S	.619	.528	.391	.555	.574	.384	.585	**.540**	.619	.533±.001
3	QWK^T	.624	.529	.393	**.574**	.582	**.391**	**.601**	.534	**.623**	**.539**±.002
						JOINT^S					
4	Loss	.614	**.534**	**.408**	.518	.583	.365	.579	.521	.567	.521±.007
5	QWK^S	.600	**.534**	**.408**	.525	.578	.364	.577	.519	.564	.519±.007
6	QWK^T	.611	**.534**	.407	.523	.581	.368	.573	.513	.559	.519±.009
						JOINT^T					
7	Loss	.615	.529	.397	.541	.578	.350	.584	.531	.564	.521±.007
8	QWK^S	.608	.527	.396	.521	.566	.348	.581	.527	.560	.515±.008
9	QWK^T	.610	.529	.398	.530	.567	.353	.585	.528	.556	.517±.011

Table 16: Trait-wise QWK scores for the ADVANCED^Q, min-max normalization setting.

	Setting	Overall	Content	Org	WC	SF	Conv	PA	Lang	Nar	AVG
						INDEPENDENT					
1	Loss	**.623**	.527	.388	**.579**	**.612**	.380	.605	**.583**	**.643**	**.549**±.003
2	QWK^S	.619	**.532**	.386	.571	.576	.366	.607	.576	.638	.541±.003
3	QWK^T	.619	**.532**	**.392**	.572	.579	**.383**	**.611**	.578	.639	.545±.003
						JOINT^S					
4	Loss	.600	.513	**.392**	.560	.587	.340	.589	.540	.573	.522±.005
5	QWK^S	.590	.512	**.392**	.550	.568	.340	.582	.541	.570	.516±.007
6	QWK^T	.597	.513	.391	.545	.575	.332	.583	.538	.568	.516±.005
						JOINT^T					
7	Loss	.573	.515	.352	.535	.555	.343	.605	.550	.572	.511±.007
8	QWK^S	.564	.513	.352	.530	.522	.344	.601	.536	.561	.503±.004
9	QWK^T	.567	.516	.350	.530	.523	.344	.596	.539	.566	.503±.006

Table 17: Trait-wise QWK scores for the ADVANCED^G, min-max normalization setting.

	Setting	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**AVG**
1	ProTACT	.647	.587	.623	.632	.674	.584	.446	.541	.592±.016
2	GAPS	**.654**	**.614**	.636	.646	.665	.590	**.469**	.498	.597±.019
	INDEPENDENT									
3	Baseline	.573	.546	.644	.651	.692	.709	.436	.595	.606±.003
4	Setting 1	.563	.552	.643	.653	.692	.710	.437	.594	.605±.003
5	Setting 2	.566	.559	.665	.670	.702	**.712**	.405	.595	.609±.003
6	Setting 3	.577	.554	.662	**.672**	**.703**	.709	.414	**.598**	**.611**±.003
	JOINT^T									
7	Baseline	.606	.546	.660	.664	.693	.695	.387	.589	.605±.003
8	Setting 1	.572	.544	.661	.670	.691	.692	.369	.563	.595±.004
9	Setting 2	.588	.576	**.673**	.651	.699	.697	.381	.571	.604±.003
10	Setting 3	.578	.570	.670	.652	.699	.696	.383	.586	.604±.003

Table 18: Prompt-wise results of exploiting the target-prompt essays for re-weighting training instances (Baseline: the scorer that has achieved the highest Average QWK score in Table 2; Setting 1: same essay type; Setting 2: same essay type, kNN; Setting 3: all training set, kNN).

	Setting	Overall	Content	Org	WC	SF	Conv	PA	Lang	Nar	AVG
	INDEPENDENT										
1	Setting 2	.635	.626	.505	.551	.619	.485	.712	.604	.716	.606±.003
2	Setting 2 Rationale Only	.604	.593	.486	.559	.624	.456	.646	.609	.662	.582±.005
3	Setting 3	.646	.630	.493	.550	.615	.486	.709	.605	.718	.606±.003
4	Setting 3 Rationale Only	.618	.611	.507	.500	.607	.503	.645	.609	.657	.584±.008
	JOINT^T										
5	Setting 2	.636	.609	.526	.586	.587	.468	.692	.635	.691	.603±.005
6	Setting 2 Rationale Only	.570	.577	.511	.498	.557	.422	.638	.568	.633	.553±.003
7	Setting 3	.632	.612	.526	.586	.596	.460	.694	.634	.692	.604±.006
8	Setting 3 Rationale Only	.585	.579	.506	.514	.546	.412	.653	.561	.645	.555±.007

Table 19: Trait-wise results of exploiting the target-prompt essays for re-weighting training instances: a comparison between using the full ADVANCED^G feature set and using only the rationale embedding as features (Baseline: the scorer that has achieved the highest Average QWK score in Table 2; Setting 2: same essay type, kNN; Setting 3: all training set, kNN).

	System	**Overall**	**Content**	**Org**	**WC**	**SF**	**Conv**	**PA**	**Lang**	**Nar**	**AVG**
	SUPERVISED										
1	Hi att	.453	.348	.243	.416	.428	.244	.309	.293	.379	.346
2	AES aug	.402	.342	.256	.402	.432	.239	.331	.313	.377	.344
3	PAES	.657	.539	.414	.531	.536	.357	.570	.531	.605	.527
4	CTS	.670	.555	.458	.557	.545	.412	.565	.536	.608	.545
5	PMAES	.671	.567	.481	.584	.582	.421	.584	.545	.614	.561
6	PLAES	.673	.574	.491	.579	.580	.447	.601	.554	.631	.570

Table 20: Trait-wise QWK scores of other cross-prompt scorers.