NUALIGN: A NUMERICAL DISTANCE PERCEPTION ALIGNMENT FOR LARGE LANGUAGE MODELS AS EVALUATORS

Anonymous authors
Paper under double-blind review

ABSTRACT

Large language models (LLMs) have shown huge potential as automatic evaluators, but the standard token-based likelihood objectives don't explicitly encode the numerical distance between scores. Previous research introduced numerical supervision through expected score regression or token-level numerical loss; however, the former can't distinguish distributions with the same average, and the latter doesn't differentiate the digit positions. To address this, we propose NuAlign, a numerical distance perception alignment for LLMs as evaluators that directly uses the full score deviations reconstructed during training to update the likelihood of the true score sequences. NuAlign consists of two stages. In Stage I, we learn to evaluate semantic and scoring boundaries through parameter-efficient supervised fine-tuning, thereby establishing a task-aware autoregressive scoring distribution. In Stage II, we utilize numerical scoring deviations to modulate token-level likelihood optimization, introducing numerical distance sensitivity while preserving the native autoregressive scoring distribution. We conducted experiments across multiple evaluation datasets and model families, achieving an MAE of 0.28 on the Mohler dataset (out of a maximum score of 5). Our code and data are available at: `https://anonymous.4open.science/r/I-LOVE_ICLR-7561`

1 INTRODUCTION

Large language models (LLMs) are increasingly being used as automated graders to assign numerical scores to students' answers. This role is becoming increasingly important in computer-assisted education. For this type of open-ended assessment, the model must map students' answers to discrete or continuous scores based on specific questions and evaluation criteria. At first glance, it is not immediately clear how a model based on text tokens can make numerical predictions. Existing research has successfully employed two main approaches. One is the autoregressive method, which directly predicts the numerical target as text (e.g., predicting 13.14 by directly predicting the tokens '1', '3', '.', '1', '4') or the corresponding discretized categories (e.g., predicting one of "good", "average", "bad"). These methods suit both context-aware learning and fine-tuning with standard cross-entropy (Fernandes et al., 2023; Liu & Low, 2023; Lukasik et al., 2024; Requeima et al., 2024; Chen et al., 2026). The other is a predictor-based approach that learns an independent head based on representations derived from the input. Common features include mean-pooled output embeddings (Zhuang et al., 2023) and token-specific log-likelihoods of the last position (Fernandes et al., 2023). Both approaches have demonstrated effectiveness in natural language regression. Standard autoregressive fine-tuning with cross-entropy learns the likelihood of target score sequences without explicitly accounting for numerical distances between scores. Predictor-based approaches enable direct numerical optimization, but produce scores through an additional prediction function rather than the native autoregressive output.

Recent research has introduced numerical supervision into language models' output space. This is categorized into expected score regression (Lukasik et al., 2025; Chiang et al., 2025) and numeric-token loss (NTL) (Zausinger et al., 2025). Although both introduce numerical structure, their numerical components have notable limitations. The expected score regression term cannot distinguish between distributions with the same mean, even if they assign different probabilities to the target

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

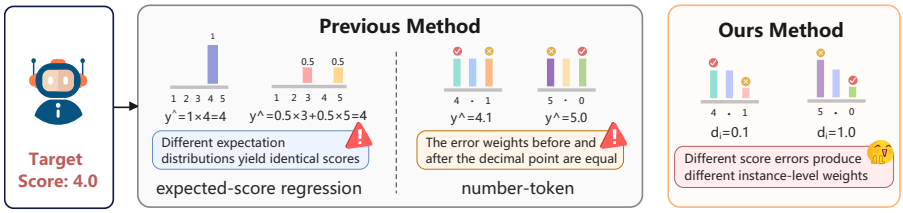


Figure 1: Comparison chart of methods for introducing numerical supervision. We use complete scores to introduce numerical supervision, addressing the shortcomings of both expected-distribution and bitwise numerical supervision methods.

score sequences. NTL does not explicitly account for bit values; therefore, equal token-level losses may correspond to different total score errors. Figure 1 illustrates these distinctions.

Based on this, we use the bias of the reconstructed total score as an instance-level signal to update the likelihood values of the gold tokens. Rather than introducing a separate regression gradient direction, this method leverages the severity of numerical errors to modulate likelihood supervision. We propose **NuAlign**, a numerically distance-aware alignment method for LLMs used as evaluators. The method consists of two stages. In **Stage I: Semantic Score Alignment (SSA)**, we first adapt a pre-trained large language model to the target scoring task through parameter-efficient supervised fine-tuning, thereby establishing a task-aware autoregressive score distribution. In **Stage II: Numerical Distance Perception (NDP)**, we use the full-score deviation in the scoring space to adjust the token-level likelihood gradient. This design utilizes full scores to measure the severity while preserving the local correction directions determined by the true target tokens. Table 1 compares optimization differences between our method and previous approaches.

The main contributions of this study are as follows: (1) We introduce reconstructed full-score deviations as a signal to modulate the likelihood updates of the gold scores, thereby accounting for the severity of numerical errors beyond individual numerical values. (2) We propose **NuAlign**, a numerical distance perception alignment for LLMs as evaluators. First, we establish a task-aware autoregressive scoring distribution, and then introduce the numerically distance-aware mechanism. (3) We conducted experiments across various scoring scenarios in three different languages and on multiple LLM architectures. Experimental results and ablation studies demonstrate NuAlign's effectiveness.

2 Related Work

2.1 Automatic Evaluation Based on Autoregression

Early work on LLM-based automatic grading primarily explored zero-shot and few-shot prompting. Prior studies have shown that carefully designed prompts, explicit grading criteria, and reasoning-based evaluation can substantially improve agreement with human judgments (Chang & Ginter, 2024; Jiang & Bosch, 2024; Duong & Meng, 2024; Mello et al., 2025; Zhao et al., 2025; Wei et al., 2025; Yoo et al., 2025). However, prompt-based methods remain sensitive to noisy or complex responses and may produce unstable or inaccurate scores. This limitation has motivated increasing interest in task-specific fine-tuning, which improves output consistency and scoring reliability through domain adaptation (Latif & Zhai, 2024). Recent work further combines fine-tuning with mechanisms such as error mining to identify informative training examples and improve grading performance (Chen et al., 2026; Xu & Corter, 2026; Fang et al., 2026; Chi et al., 2026; Wang et al., 2026). Beyond prompting and training-example selection, related work also introduces numerical structure into autoregressive training objectives. For numerical prediction and mathematical tasks, Number Token Loss (NTL) augments cross-entropy with a regression-like loss on number-token probabilities (Zausinger et al., 2025). For LLM-as-a-judge, TRACT combines cross-entropy supervision of CoT-and-score sequences with a squared-error loss on the expected score conditioned on chain-of-thought (CoT) reasoning (Chiang et al., 2025).

Table 1: Comparison table of methods from an optimization perspective. Unlike previous methods, we incorporate numerical supervision only as an optimization signal in the cross-entropy loss.

Approach	Autoregressive	Fine-tuning	References
Zero-shot learning	✓	None	Kocmi & Federmann (2023)
Fine-tuning and decoding	✓	Cross-entropy	Fernandes et al. (2023)
Predictive head	✗	Target metric	Fernandes et al. (2023)
RAFT	✓	Expected-score regression	Lukasik et al. (2025)
NTL	✓	Cross-entropy and number-token	Zausinger et al. (2025)
TRACT	✓	Cross-entropy and expected-score regression	Chiang et al. (2025)
NuAlign	✓	Cross-entropy and numerical-deviation	*This work*

2.2 Automatic Evaluation Based on Predictors

Automatic grading using LLM prediction heads primarily involves training LLMs to predict scores. For example, Hallac & Ogul (2026) proposed a two-stage method for grading essays using large models, transitioning from pairwise comparisons to absolute scores: in the first stage, a directed twin sorter is trained based on paired samples constructed from absolute attribute labels; in the second stage, an absolute score predictor is trained using a configurable transfer strategy. Similarly, Klus et al. (2025) fine-tuned LLaMA-3 for grading using continuous regression. Specifically, they designed adapters and predictors using quantization and Low-Rank Adaptation (LoRA), and adjusted the model to suit the grading task. Meanwhile, several studies, including (Chi et al., 2026; Katuka et al., 2024; Yang et al., 2020), have also adopted predictor-based grading methods.

3 Background

3.1 Notation and Task Definition

Let $\mathcal{V}$ denote the token vocabulary and $\mathcal{X} \subseteq \mathcal{V}^*$ the input space. For an evaluation instance $x \in \mathcal{X}$, let $y \in \mathcal{Y} \subseteq \mathbb{R}$ denote its human-assigned numerical score, and let $s(y) = \mathrm{str}(y)$ denote the corresponding score token sequence. We use $\mathbb{P}$ for the data distribution over $\mathcal{X} \times \mathcal{Y}$ and p_θ for the autoregressive distribution parameterized by an LLM.

Given x, an autoregressive scorer generates a score sequence $\hat{s}$ and converts it into a numerical prediction:

$$\hat{s} = \mathrm{Decode}(p_\theta(\cdot \mid x)), \qquad \hat{y}_{\mathrm{AR}}(x) = \mathrm{float}(\hat{s}), \tag{1}$$

where $\mathrm{float}(\cdot)$ parses the generated score string into a numerical value. Decoding may be greedy or constrained to valid score representations (Naseh et al., 2023; Geng et al., 2023; Lukasik et al., 2024).

3.2 Autoregressive Scoring

Standard autoregressive fine-tuning maximizes the likelihood of the human-labeled score sequence. For $s(y) = (s_1, \ldots, s_T)$, the objective is:

$$\mathcal{L}_{\mathrm{AR}} = \mathbb{E}_{(x,y)\sim\mathbb{P}}\left[-\sum_{t=1}^{T} \log p_\theta\left(s_t \mid x, s_{<t}\right)\right]. \tag{2}$$

This objective preserves the native autoregressive scoring, but its token-level gradient does not explicitly depend on the numerical distance between an incorrect score and the target score; see Appendix A.1.

3.3 Predictor-Based Scoring

An alternative is to predict the score from an internal LLM's representation. Let $\Phi_\theta(x) \in \mathbb{R}^q$ denote the representation produced by the backbone, where q is its dimensionality, and let s_ω denote a task-specific regression head. The prediction is:

$$\hat{y}(x) = s_\omega(\Phi_\theta(x)), \tag{3}$$

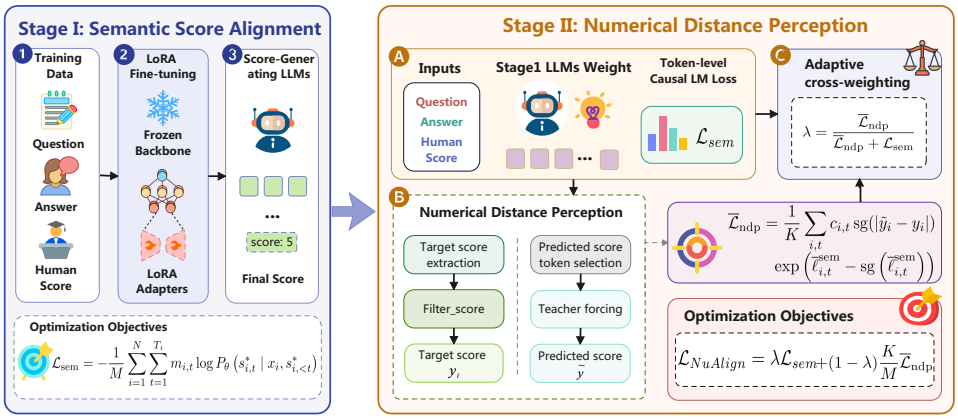


Figure 2: Overview of NuAlign. Stage I performs semantic score alignment, while Stage II introduces numerical-distance perception calibration.

where a linear head corresponds to $s_\omega(\Phi) = b + w^\top \Phi$. It is typically optimized with squared error,

$$\mathcal{L}_{\text{pred}} = \mathbb{E}_{(x,y)\sim\mathbb{P}}\left[(y - \hat{y}(x))^2\right]. \tag{4}$$

This formulation provides direct numerical supervision, but prediction is performed through an additional scalar head rather than the LLM's native autoregressive output. Moreover, the regression objective does not uniquely determine the underlying token distribution; see Appendix A.2.

3.4 Numerical Supervision for Autoregressive Scoring

Recent methods introduce numerical supervision while retaining autoregressive prediction. For a finite valid score set $\mathcal{Y}_{\text{grid}}$, let $q_\theta(y \mid x)$ denote the normalized probability that the LLM assigns to the textual representation $s(y)$. RAFT (Lukasik et al., 2025) optimizes the expected score:

$$\mu_\theta(x) = \sum_{y\in\mathcal{Y}_{\text{grid}}} q_\theta(y \mid x)\, y, \qquad \mathcal{L}_{\text{RAFT}} = \left(y^* - \mu_\theta(x)\right)^2. \tag{5}$$

Because this objective depends on the score distribution through its mean, different distributions with the same expectation receive the same regression loss; see Appendix A.3. The expected-score component of TRACT (Chiang et al., 2025) inherits this mean-only limitation, although its additional likelihood supervision can distinguish distributions that assign different probabilities to the gold sequence. NTL (Zausinger et al., 2025) instead introduces numerical supervision at the number-token level; however, for multi-token scores, token-level numerical distances need not match the error of the reconstructed complete score. We analyze this distinction in Appendix A.3.

4 NuAlign

Based on an analysis of existing research paradigms, NuAlign explores a different use of numerical distance. While retaining gold-score likelihood supervision, it uses the bias in score reconstruction during training to adjust its gradient. As shown in Figure 2, the method consists of two stages.

4.1 Stage I: Semantic Score Alignment

Let $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N$, where x_i contains the question and the answer to be evaluated, and y_i is the human-annotated numerical score. We serialize y_i into a fixed score-response template and denote the resulting token sequence by s_i^*. The token-level likelihood loss is $\ell_{i,t}^{\text{sem}} = -\log p_\theta(s_{i,t}^* \mid x_i, s_{i,<t}^*)$. Stage I minimizes:

$$\mathcal{L}_{\text{sem}} = \frac{1}{M}\sum_{i,t} m_{i,t}\ell_{i,t}^{\text{sem}}, \qquad M = \sum_{i,t} m_{i,t}, \tag{6}$$

where $m_{i,t}$ in $0, 1$ indicates the supervised response position; we apply fixed supervision to the fractional response area. Here and below, sums over i are taken over the current mini-batch. We optimize this standard supervised fine-tuning objective with LoRA and use the resulting parameters θ_1 as the task-adapted initialization for Stage II.

4.2 Stage II: Numerical Distance Perception

The likelihood objective supervises sequences of human-labeled scores but does not explicitly encode the numerical distance between incorrect and target scores. The second stage introduces this information through NDP while retaining likelihood supervision. NDP does not differentiate based on numerical score predictions; instead, it uses the deviation between these predictions and human scores to weight the likelihood gradient of the gold tokens.

Training-time score reconstruction. Starting with θ_1, we reuse the "teacher-forced" logits from standard likelihood training and reconstruct the full scores through position-wise discrete predictions, thereby avoiding the need for additional autoregressive generation at each training step. (Of course, we also compare reconstructing full scores using "teacher-forced" logits and using autoregressive generation in Appendix A.4.) Let u_i^* denote the full training sequence consisting of the scoring prompt followed by the gold score response, and $\mathbf{z}_{i,t} = f_\theta(\mathbf{u}^*_{i,<t})$ the logits predicting position t. The ordered readout positions $\mathcal{P}_i$ need not correspond to supervised targets. We take position-wise argmax tokens and parse the decoded sequence:

$$\tilde{y}_i = \phi\left(\text{TokenizerDecode}\left(\left(\arg\max_{v\in\mathcal{V}} z_{i,t,v}\right)_{t\in\mathcal{P}_i}\right)\right). \tag{7}$$

Here $\phi(\cdot)$ parses a numerical score; Appendix A.5 specifies parsing-failure handling. For a valid reconstruction, the numerical modulation signal is:

$$d_i = |\tilde{y}_i - y_i|\,. \tag{8}$$

NDP optimization of the likelihood gradient. Let $c_{i,t} \in \{0, 1\}$ select the batch-shared NDP computation slice. Define $r_{i,t} = c_{i,t}m_{i,t}$ and $\mathcal{R}_i = \{t : r_{i,t} = 1\}$, so $r_{i,t} \le m_{i,t}$. Treating d_i as a stop-gradient signal, NDP induces:

$$\nabla_\theta \ell^{\text{ndp}}_{i,t} = d_i \nabla_\theta \ell^{\text{sem}}_{i,t}, \qquad t \in \mathcal{R}_i. \tag{9}$$

The same d_i is shared across $\mathcal{R}_i$. For a fixed likelihood gradient, a larger deviation increases the magnitude of the NDP contribution without changing its direction.

Using the masked loss $\bar{\ell}^{\text{sem}}_{i,t} = m_{i,t}\ell^{\text{sem}}_{i,t}$, set to zero at ignored targets, we implement:

$$\ell^{\text{ndp}}_{i,t} = \text{sg}(d_i)\exp\left(\bar{\ell}^{\text{sem}}_{i,t} - \text{sg}\left(\bar{\ell}^{\text{sem}}_{i,t}\right)\right), \qquad c_{i,t} = 1, \tag{10}$$

with zero values outside the slice. Here $\text{sg}(\cdot)$ preserves forward values and blocks backward gradients. The exponential equals one: each selected slot has value d_i, while valid targets yield Eq. equation 9. Ignored slots retain this forward value but have zero local likelihood gradient. Neither token selection nor parsing is differentiated. Appendix A.6 details the masks, readout positions, and derivation.

Adaptive cross-weighting. We use adaptive cross-weighting to coordinate likelihood supervision and NDP optimization without a fixed mixing coefficient.

Let $M = \sum_{i,t} m_{i,t} > 0$ and $K = \sum_{i,t} c_{i,t} > 0$. We average over all selected computation slots, including ignored targets:

$$\overline{\mathcal{L}}_{\text{ndp}} = \frac{1}{K}\sum_{i,t} c_{i,t}\ell^{\text{ndp}}_{i,t}. \tag{11}$$

Since every example contributes the same number of slice positions, the forward value of $\overline{\mathcal{L}}_{\text{ndp}}$ is the batch mean of d_i. With $\mathcal{L}_{\text{sem}} = M^{-1}\sum_{i,t}\bar{\ell}^{\text{sem}}_{i,t}$, define:

$$\lambda = \frac{\overline{\mathcal{L}}_{\text{ndp}}}{\overline{\mathcal{L}}_{\text{ndp}} + \mathcal{L}_{\text{sem}}}, \tag{12}$$

Table 2: Comparison of NuAlign with existing methods across three datasets and different LLMs. For the five LLMs, the results are reported as mean ± standard deviation.

Models	Mohler			AES-ENEM			Chinese-QA		
	MAE ↓	RMSE ↓	Pearson ↑	MAE ↓	RMSE ↓	Pearson ↑	MAE ↓	RMSE ↓	Pearson ↑
MTS (Lee et al., 2024)	-	-	-	193.71	215.94	0.58	2.25	2.79	0.73
Praetor (Leng et al., 2025)	1.84	2.16	0.41	-	-	-	1.08	1.93	0.74
CEM-FT (Chen et al., 2026)	0.40	-	-	62.47	94.56	0.80	0.70	1.15	0.88
RAFT (Lukasik et al., 2025)	0.44	0.67	0.77	59.87	84.17	0.85	0.56	0.99	**0.93**
NTL (Zausinger et al., 2025)	0.40	0.69	0.78	64.03	91.09	0.82	0.57	1.09	0.92
TRACT (Chiang et al., 2025)	0.45	0.77	0.75	63.13	88.41	0.83	0.61	1.29	0.90
Qwen3-8B (ours)	0.32 ± 0.00	0.57 ± 0.01	0.84 ± 0.00	57.33 ± 0.06	85.24 ± 0.01	0.84 ± 0.00	0.54 ± 0.03	0.98 ± 0.04	**0.93** ± 0.02
Qwen3-14B (ours)	**0.28** ± 0.01	**0.50** ± 0.03	**0.88** ± 0.01	**57.21** ± 0.08	83.92 ± 0.15	0.84 ± 0.00	**0.53** ± 0.05	**0.97** ± 0.08	**0.93** ± 0.01
Llama-3.1-8B-Instruct (ours)	0.29 ± 0.00	0.53 ± 0.02	0.86 ± 0.01	57.21 ± 0.00	**83.45** ± 0.15	**0.85** ± 0.10	0.55 ± 0.02	1.00 ± 0.05	**0.93** ± 0.01
Mistral-7b-instruct-v0.3 (ours)	0.34 ± 0.01	0.61 ± 0.03	0.82 ± 0.00	57.58 ± 0.14	86.40 ± 0.25	**0.85** ± 0.05	0.62 ± 0.03	1.06 ± 0.05	0.92 ± 0.03
Gemma2-2B (ours)	0.42 ± 0.01	0.72 ± 0.05	0.77 ± 0.01	58.20 ± 0.05	85.15 ± 0.09	0.84 ± 0.02	0.56 ± 0.01	1.00 ± 0.03	**0.93** ± 0.02

for $\overline{\mathcal{L}}_{\text{ndp}} + \mathcal{L}_{\text{sem}} > 0$. A larger relative NDP value increases the semantic coefficient λ and decreases the NDP coefficient $1 - \lambda$, and conversely for a larger relative semantic loss.

The Stage II objective is:

$$\mathcal{L}_{\text{NuAlign}} = \frac{1}{M} \sum_{i,t} \left[\lambda \bar{\ell}_{i,t}^{\text{sem}} + (1 - \lambda) c_{i,t} \ell_{i,t}^{\text{ndp}} \right]. \tag{13}$$

Unlike d_i, λ is not detached, so its parameter dependence is included in backpropagation. Appendix A.7 derives the full gradient; Section 5.5 evaluates adaptive cross-weighting.

5 Experiment and Analysis

5.1 Experimental Settings

Datasets. We evaluate NuAlign on three automatic scoring benchmarks covering different tasks and languages: Mohler is an English short-answer question scoring dataset with a score range of 0–5 (Mohler et al., 2011), AES-ENEM is a Portuguese essay scoring dataset with a score range of 0–1,000 (Silveira et al., 2024), and Chinese-QA is an internal Chinese open-ended question-answering scoring dataset with a score range of 0–8. Appendix A.8 provides dataset statistics.

Models. We experiment with five open-source LLM backbones of different model families and scales: Qwen3-8B, Qwen3-14B (Yang et al., 2025), Llama-3.1-8B-Instruct (Team, 2024), Mistral-7B-Instruct-v0.3 (Jiang et al., 2023), and Gemma2-2B (Team et al., 2024). We used three different random seeds to report the experimental results. Training configurations and prompt templates are provided in Appendices A.9 and A.10. In addition, we provide the complete experimental data at the code link.

Methods. We compare NuAlign with representative recent automatic scoring methods, including MTS (Lee et al., 2024), Praetor (Leng et al., 2025), CEM-FT (Chen et al., 2026), RAFT (Lukasik et al., 2025), NTL (Zausinger et al., 2025) and TRACT (Chiang et al., 2025) . Appendix A.11 provides detailed baseline configurations.

Metrics. We report mean absolute error (MAE), root mean squared error (RMSE), and Pearson correlation. Lower MAE and RMSE indicate smaller scoring errors, while higher Pearson correlation indicates stronger agreement with human-assigned scores.

Table 3: Results of ablation experiments across three datasets.

Models	Method	Mohler			AES-ENEM			Chinese-QA		
		MAE ↓	RMSE ↓	Pearson ↑	MAE ↓	RMSE ↓	Pearson ↑	MAE ↓	RMSE ↓	Pearson ↑
Qwen3-8B	Base	0.34	0.62	0.82	60.92	91.06	0.82	0.71	1.13	0.90
	ONDP	0.50	0.83	0.73	78.51	108.16	0.75	0.65	1.08	0.91
	RH	0.48	0.82	0.75	67.74	91.35	0.81	0.64	1.19	0.92
	FLW	0.36	0.61	0.82	62.29	89.33	0.82	0.54	1.03	0.92
	NuAlign	**0.32**	**0.57**	**0.84**	**57.33**	**85.24**	**0.84**	**0.54**	**0.98**	**0.93**
Qwen3-14B	Base	0.32	0.56	0.86	63.52	88.04	0.82	0.66	1.06	0.91
	ONDP	0.54	0.85	0.74	73.68	99.95	0.79	0.68	1.14	0.91
	RH	0.39	0.51	**0.86**	65.15	88.89	0.82	0.59	0.99	**0.93**
	FLW	0.30	0.51	0.87	59.19	86.77	0.83	0.54	0.99	0.92
	NuAlign	**0.28**	**0.50**	**0.88**	**57.21**	**83.92**	**0.84**	**0.53**	**0.97**	**0.93**
Llama-3.1-8B	Base	0.39	0.70	0.81	59.46	95.88	0.79	0.78	1.21	0.89
	ONDP	0.44	0.72	0.74	66.62	94.37	0.80	0.68	1.14	0.90
	RH	0.35	0.53	**0.86**	58.49	96.15	0.81	0.64	1.01	0.92
	FLW	0.30	0.54	**0.86**	57.60	94.97	0.80	0.58	1.04	0.92
	NuAlign	**0.29**	**0.53**	**0.86**	**57.21**	**83.45**	**0.85**	**0.55**	**1.00**	**0.93**
Mistral-7b	Base	0.39	0.69	0.78	66.37	99.28	0.81	0.78	1.24	0.88
	ONDP	0.39	0.63	0.81	61.55	93.51	0.82	0.72	1.24	0.88
	RH	0.39	0.64	0.80	59.31	97.42	0.80	0.65	1.11	0.91
	FLW	0.36	0.62	0.81	57.90	94.92	0.80	0.62	1.08	0.91
	NuAlign	**0.34**	**0.61**	**0.82**	**57.58**	**86.40**	**0.85**	**0.62**	**1.06**	**0.92**
Gemma2-2B	Base	0.56	0.91	0.71	61.17	92.25	0.82	1.06	1.67	0.81
	ONDP	0.48	0.77	0.74	59.44	89.85	0.82	0.63	1.07	0.91
	RH	1.02	1.12	0.74	64.23	95.87	0.80	0.72	1.03	0.92
	FLW	0.43	0.74	0.75	58.45	88.30	0.83	0.59	1.06	0.92
	NuAlign	**0.42**	**0.72**	**0.77**	**58.20**	**85.15**	**0.84**	**0.56**	**1.00**	**0.93**

5.2 Overall performance

Table 2 compares NuAlign's performance with that of recent state-of-the-art methods. On the evaluation datasets, NuAlign's scoring error is significantly lower than that of MTS and Praetor. CEM-FT provides a stronger benchmark, but NuAlign's five backbone models still improve its MAE and RMSE on the AES-ENEM and Chinese-QA datasets.

RAFT, NTL, and TRACT integrate numerical supervision, providing a more targeted comparison. On the Mohler metric, Qwen3-8B, Qwen3-14B, Llama-3.1-8B, and Mistral-7B outperformed the three methods in MAE, RMSE, and Pearson correlation. Compared to NTL, Qwen3-14B reduced MAE by 30.0%; compared to RAFT and TRACT, it reduced RMSE by 25.4% and 35.06%, respectively. These results confirm the potential advantages of using full-score bias to guide likelihood updates, outperforming approaches that introduce numerical supervision solely through expected scores or individual numeric tokens.

NuAlign continues to perform consistently on the AES-ENEM and Chinese-QA tasks. On the AES-ENEM dataset, all NuAlign backbone models had lower MAE than all evaluated baseline models. Llama achieved the lowest RMSE while matching the Pearson correlation coefficient. NTL performed worse on the AES-ENEM dataset than on the other two datasets, which validates our claim that direct token-by-token optimization without considering bit values has limitations. On the Chinese-QA dataset, both Qwen backbone models reduced MAE and RMSE compared to RAFT, NTL, and TRACT, while achieving the state-of-the-art Pearson correlation coefficient of 0.93.

Overall, these results demonstrate that NuAlign is competitive across multiple model families, though this does not imply universal superiority. For example, Gemma2-2B outperforms CEM-FT on the AES-ENEM and Chinese-QA tasks but still lags behind the strongest baseline model on the Mohler task, which may be related to Gemma's parameter count. Subsequent ablation experiments further examined NuAlign's contribution.

5.3 Ablation experiments on different models

To further validate the roles of each module, we conducted four additional experiments: (1) Base: Removed the second stage, resulting in simple fine-tuning; (2) Only Numerical distance perception (ONDP): Removed the first stage, performing ONDP; (3) Regression head (RH): Added a regression

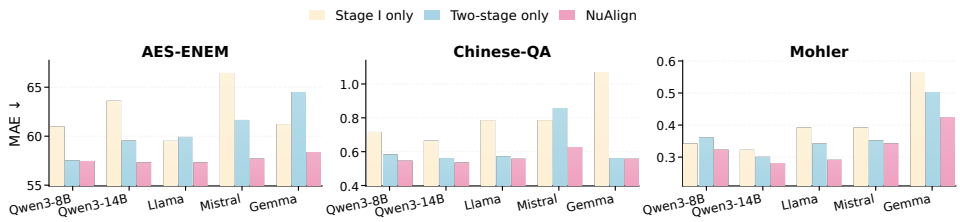


Figure 3: Effect of additional training on experimental results. Although additional parameters can improve model performance to some extent, they are not the primary reason for NuAlign's significant effectiveness.

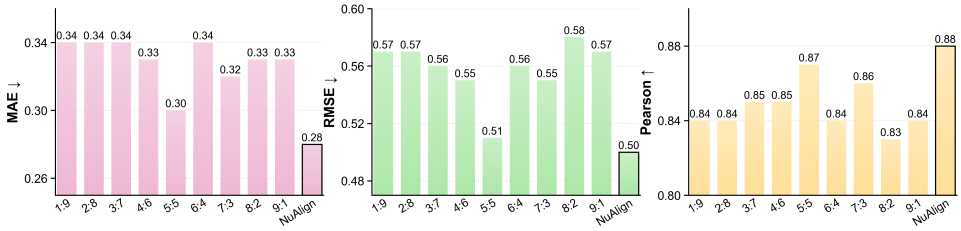


Figure 4: Effect of adaptive cross-weighting. Using λ helps, to some extent, avoid a grid search to determine the optimal parameters.

adapter after the large language model and used mean squared error loss for the experiment. (4) Fixed loss weighting (FLW): Set Stage II λ to 0.5. Table 3 shows the results.

Semantic Initialization NuAlign outperforms both Base and ONDP on all reported metrics, confirming the effectiveness of the full training scheme. However, on the three Gemma2-2B datasets, ONDP also outperformed Base, indicating that numerical distance-aware approaches without initialization still hold practical value. A plausible explanation for NuAlign's advantage is that the first stage establishes a mapping between answers and scores, making subsequent numerical deviations more informative for likelihood modulation. These results support task adaptation before numerical distance-aware approaches but do not prove its universal necessity.

Likelihood-based Supervision and Predictor-based Supervision. RH reveals a trade-off between error metrics: on the Qwen3-14B model on the Mohler dataset, it reduces the Base model's RMSE from 0.56 to 0.51 but increases the MAE from 0.32 to 0.39. NuAlign achieved an RMSE of 0.50 and an MAE of 0.28, thereby avoiding this trade-off. More broadly, NuAlign performed on par with or better than RH on all reported metrics, showing that likelihood-based numerical supervision is a valid alternative. Since RH modifies both the prediction interface and the objective function, this does not prove the inherent superiority of autoregressive prediction.

Adaptive and Fixed Weights. At the reported precision levels, NuAlign outperformed FLW in 42 of 45 comparisons and tied in 3. On AES-ENEM, the RMSE decreased from 94.97 to 83.45, while the MAE decreased only from 57.60 to 57.21, indicating that autoregressive prediction has a larger impact on larger residual errors in this setting. Nevertheless, FLW has achieved most of the improvements relative to Base. We further explore the impact of weights on the experimental results in Section 5.5.

5.4 Performance Gains from Two-Stage Training

To assess whether NuAlign's performance improvement stems solely from additional training, we used seed 3407. We compared the "Stage-I Only" and "Two-Stage Only" methods (Compared to Stage-I Only, we trained the model twice using the same configuration, with the second training run initialized from the first stage) and NuAlign across three datasets under the same configuration. Figure 3 shows the MAE results.

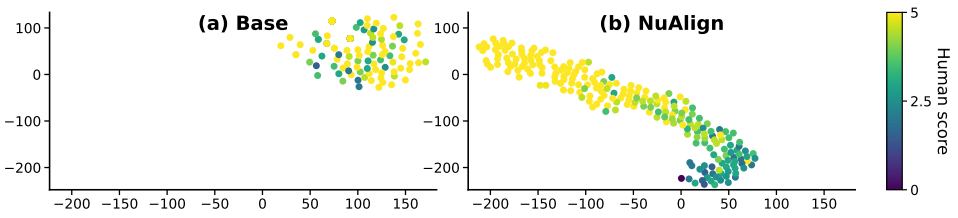


Figure 5: t-SNE visualization of Base and NuAlign representations, colored by human scores.

NuAlign consistently outperforms Stage-I Only and further improves over Two-Stage Only, indicating that the gains cannot be explained by additional training alone and supporting the contribution. On Mohler, NuAlign performs best across all model–metric combinations. Models with larger Stage I errors also benefit substantially: Gemma reduces MAE from 1.06 to 0.56 on the Chinese-QA dataset, a relative drop of about 47.2%; for Llama on the Mohler dataset, MAE decreases from 0.39 to 0.29, a relative reduction of about 25.6%. These results suggest that the complete Stage-II design remains effective across models with different initial scoring performance. Overall, the findings support our motivation that automatic scoring requires not only semantic score alignment but also modeling numerical deviations between predicted and human scores. NuAlign achieves this by using score-distance information to modulate likelihood optimization after Stage I alignment.

5.5 Stage II Weight Sensitivity Analysis

To investigate the effect of the Stage II weight λ, we conducted further experiments using the Qwen3-14B model (seed 3407) on the Mohler dataset, comparing NuAlign with nine fixed λ values: $(1-\lambda)$ ratios ranging from $1:9$ to $9:1$. As shown in Figure 4, performance under fixed-weight settings is non-monotonic. Among the fixed-weight ratios, the balanced $5:5$ setting performed best, achieving 0.30 MAE, 0.51 RMSE, and a Pearson correlation coefficient of 0.87, while more asymmetric settings performed worse. Adaptive cross-weighting in NuAlign further improves these results to 0.28 MAE, 0.50 RMSE, and a Pearson correlation coefficient of 0.88. This indicates that a single global coefficient cannot consistently capture the relative contributions of semantic likelihood supervision and NDP. Instead, adaptive weighting rules adjust their influence based on the current optimization margin, thereby providing a more effective balance than the fixed ratios tested.

5.6 Visual Analytics

To examine whether NuAlign induces score-aware representation structure, we visualize learned representations using t-SNE. We project final-layer evaluation representations into a two-dimensional space and color points by human-assigned score. As shown in Figure 5, the Base model (standard fine-tuning only) shows substantial overlap across different human scores, with low, medium, and high scores mixed. In contrast, NuAlign produces a structured manifold with gradual transitions from high to low scores. Similar-score samples form coherent neighborhoods, while larger score gaps are farther apart. This suggests that numerical-distance-aware optimization encourages representations to reflect the ordering and proximity of human scores, rather than merely fitting supervision labels. These observations align with the quantitative improvements above, indicating NuAlign learns a more score-aware, order-sensitive representation space.

6 Conclusion

We propose NuAlign, a numerical distance perception alignment method for autoregressive automatic scoring. In stage I, we establish a task-aware scoring distribution through parameter-efficient supervised fine-tuning. In stage II, we adjust the score-token likelihood gradient by reconstructing the deviation between the trained scores and human scores during training, thereby incorporating the severity of numerical errors without relying on an external regression head. Experiments on three scoring datasets and five LLM backbone models show that this method improves performance over standard fine-tuning. We expect this approach to contribute to further advances across a wider range of scoring tasks.

AI USE STATEMENT

In this work, we used generative AI tools to assist with translation and language polishing. In addition, as described in Appendix A.11.2, we used GPT-5.5 to generate dimension-specific scoring criteria for the MTS baseline on the AES-ENEM and Chinese-QA datasets. This use was part of the baseline implementation, separate from assistance with manuscript preparation. We reviewed the AI-assisted content and revised it as appropriate, including checking that the wording preserved our intended meaning. The authors take full responsibility for the research decisions, experimental results, mathematical claims, and all text, code, figures, and other artifacts included in the final work.

REFERENCES

Li-Hsin Chang and Filip Ginter. Automatic short answer grading for finnish with chatgpt. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 23173–23181, 2024.

Lei Chen, Tengteng Cheng, BoYu Gao, Zitao Liu, and Weiqi Luo. Learning from scoring disagreements: Contrastive error mining for efficient and robust llm-based assessment. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 30243–30251, 2026.

Jinwei Chi, Ke Wang, Yu Chen, Xuanye Lin, and Qiang Xu. Activations as features: Probing llms for generalizable essay scoring representations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 30395–30403, 2026.

Cheng-Han Chiang, Hung-yi Lee, and Michal Lukasik. TRACT: Regression-aware fine-tuning meets chain-of-thought reasoning for LLM-as-a-judge. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 2934–2952, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.147. URL https://aclanthology.org/2025.acl-long.147/.

Ta Nguyen Binh Duong and Chai Yi Meng. Automatic grading of short answers using large language models in software engineering courses. In *2024 IEEE Global Engineering Education Conference (EDUCON)*, pp. 1–10, 2024. doi: 10.1109/EDUCON60312.2024.10578839.

Luyang Fang, Tao Wang, Ping Ma, and Xiaoming Zhai. Generalizable and efficient automated scoring with a knowledge-distilled multi-task mixture-of-experts. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 40831–40839, 2026.

Patrick Fernandes, Daniel Deutsch, Mara Finkelstein, Parker Riley, André F. T. Martins, Graham Neubig, Ankush Garg, Jonathan H. Clark, Markus Freitag, and Orhan Firat. The devil is in the errors: Leveraging large language models for fine-grained machine translation evaluation. In Philipp Koehn, Barry Haddon, Tom Kocmi, and Christof Monz (eds.), *Proceedings of the Eighth Conference on Machine Translation, WMT 2023, Singapore, December 6-7, 2023*, pp. 1066–1083. Association for Computational Linguistics, 2023. doi: 10.18653/V1/2023.WMT-1.100. URL https://doi.org/10.18653/v1/2023.wmt-1.100.

Saibo Geng, Martin Josifoski, Maxime Peyrard, and Robert West. Grammar-constrained decoding for structured NLP tasks without finetuning. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pp. 10932–10952. Association for Computational Linguistics, 2023. doi: 10.18653/V1/2023.EMNLP-MAIN.674. URL https://doi.org/10.18653/v1/2023.emnlp-main.674.

Ibrahim Riza Hallac and Hasan Ogul. Pair2score: Pairwise-to-absolute transfer for llm-based essay scoring. *CoRR*, abs/2605.02069, 2026. doi: 10.48550/ARXIV.2605.02069. URL https://doi.org/10.48550/arXiv.2605.02069.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7b, 2023. URL https://arxiv.org/abs/2310.06825.

Lan Jiang and Nigel Bosch. Short answer scoring with gpt-4. In *Proceedings of the eleventh acm conference on learning@ scale*, pp. 438–442, 2024.

Gloria Ashiya Katuka, Alexander Gain, and Yen-Yun Yu. Investigating automatic scoring and feedback using large language models, 2024. URL https://arxiv.org/abs/2405.00602.

J. Klus, D. E. Cohen, A. N. Garcia, et al. Modeling memories, predicting prospections: Automated scoring of autobiographical detail narration using large language models. *Behavior Research Methods*, 57:245, 2025. doi: 10.3758/s13428-025-02767-3.

Tom Kocmi and Christian Federmann. Large language models are state-of-the-art evaluators of translation quality. In Mary Nurminen, Judith Brenner, Maarit Koponen, Sirkku Latomaa, Mikhail Mikhailov, Frederike Schierl, Tharindu Ranasinghe, Eva Vanmassenhove, Sergi Alvarez Vidal, Nora Aranberri, Mara Nunziatini, Carla Parra Escartín, Mikel L. Forcada, Maja Popovic, Carolina Scarton, and Helena Moniz (eds.), *Proceedings of the 24th Annual Conference of the European Association for Machine Translation, EAMT 2023, Tampere, Finland, 12-15 June 2023*, pp. 193–203. European Association for Machine Translation, 2023. URL https://aclanthology.org/2023.eamt-1.19.

Ehsan Latif and Xiaoming Zhai. Fine-tuning chatgpt for automatic scoring. *Computers and Education: Artificial Intelligence*, 6:100210, 2024. ISSN 2666-920X. doi: https://doi.org/10.1016/j.caeai.2024.100210. URL https://www.sciencedirect.com/science/article/pii/S2666920X24000110.

Sanwoo Lee, Yida Cai, Desong Meng, Ziyang Wang, and Yunfang Wu. Unleashing large language models' proficiency in zero-shot essay scoring. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2024, Miami, Florida, USA, November 12-16, 2024*, pp. 181–198. Association for Computational Linguistics, 2024. doi: 10.18653/V1/2024.FINDINGS-EMNLP.10. URL https://doi.org/10.18653/v1/2024.findings-emnlp.10.

Yongqi Leng, Renren Jin, Yue Chen, Zhuowen Han, Ling Shi, Jianxiang Peng, Lei Yang, Juesi Xiao, and Deyi Xiong. Praetor: A fine-grained generative LLM evaluator with instance-level customizable evaluation criteria. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2025, Vienna, Austria, July 27 - August 1, 2025*, pp. 10386–10418. Association for Computational Linguistics, 2025. URL https://aclanthology.org/2025.acl-long.513/.

Tiedong Liu and Bryan Kian Hsiang Low. Goat: Fine-tuned llama outperforms GPT-4 on arithmetic tasks. *CoRR*, abs/2305.14201, 2023. doi: 10.48550/ARXIV.2305.14201. URL https://doi.org/10.48550/arXiv.2305.14201.

Michal Lukasik, Harikrishna Narasimhan, Aditya Krishna Menon, Felix X. Yu, and Sanjiv Kumar. Regression aware inference with llms. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2024, Miami, Florida, USA, November 12-16, 2024*, volume EMNLP 2024 of *Findings of ACL*, pp. 13667–13678. Association for Computational Linguistics, 2024. doi: 10.18653/V1/2024.FINDINGS-EMNLP.799. URL https://doi.org/10.18653/v1/2024.findings-emnlp.799.

Michal Lukasik, Zhao Meng, Harikrishna Narasimhan, Yin-Wen Chang, Aditya Krishna Menon, Felix Yu, and Sanjiv Kumar. Better autoregressive regression with llms via regression-aware fine-tuning. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025. URL https://openreview.net/forum?id=xGs7Ch3Vyo.

Rafael Ferreira Mello, Cleon Pereira Junior, Luiz A. L. Rodrigues, Filipe Dwan Pereira, Luciano de Souza Cabral, Newarney T. Costa, Geber L. Ramalho, and Dragan Gasevic. Automatic short answer grading in the LLM era: Does GPT-4 with prompt engineering beat traditional models? In *Proceedings of the 15th International Learning Analytics and Knowledge Conference, LAK 2025, Dublin, Ireland, March 3-7, 2025*, pp. 93–103. ACM, 2025. doi: 10.1145/3706468.3706481. URL https://doi.org/10.1145/3706468.3706481.

Michael Mohler, Razvan Bunescu, and Rada Mihalcea. Learning to grade short answer questions using semantic similarity measures and dependency graph alignments. In Dekang Lin, Yuji Matsumoto, and Rada Mihalcea (eds.), *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pp. 752–762, Portland, Oregon, USA, June 2011. Association for Computational Linguistics. URL https://aclanthology.org/P11-1076/.

Ali Naseh, Kalpesh Krishna, Mohit Iyyer, and Amir Houmansadr. Stealing the decoding algorithms of language models. In Weizhi Meng, Christian Damsgaard Jensen, Cas Cremers, and Engin Kirda (eds.), *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security, CCS 2023, Copenhagen, Denmark, November 26-30, 2023*, pp. 1835–1849. ACM, 2023. doi: 10.1145/3576915.3616652. URL https://doi.org/10.1145/3576915.3616652.

James Requeima, John Bronskill, Dami Choi, Richard E. Turner, and David Kristjanson Duvenaud. LLM processes: Numerical predictive distributions conditioned on natural language. In Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang (eds.), *Advances in Neural Information Processing Systems 37: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024. URL http://papers.nips.cc/paper_files/paper/2024/hash/c5ec22711f3a4a2f4a0a8ffd92167190-Abstract-Conference.html.

Igor Cataneo Silveira, André Barbosa, and Denis Deratani Mauá. A new benchmark for automatic essay scoring in portuguese. In Pablo Gamallo, Daniela Barreiro Claro, António J. S. Teixeira, Livy Real, Marcos García, Hugo Gonçalo Oliveira, and Raquel Amaro (eds.), *Proceedings of the 16th International Conference on Computational Processing of Portuguese, PROPOR 2024, Santiago de Compostela, Galicia/Spain, March 12-15, 2024, Volume 1*, pp. 228–237. Association for Computational Lingustics, 2024. URL https://aclanthology.org/2024.propor-1.23.

Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, Johan Ferret, Peter Liu, Pouya Tafti, Abe Friesen, Michelle Casbon, Sabela Ramos, Ravin Kumar, Charline Le Lan, Sammy Jerome, Anton Tsitsulin, Nino Vieillard, Piotr Stanczyk, Sertan Girgin, Nikola Momchev, Matt Hoffman, Shantanu Thakoor, Jean-Bastien Grill, Behnam Neyshabur, Olivier Bachem, Alanna Walton, Aliaksei Severyn, Alicia Parrish, Aliya Ahmad, Allen Hutchison, Alvin Abdagic, Amanda Carl, Amy Shen, Andy Brock, Andy Coenen, Anthony Laforge, Antonia Paterson, Ben Bastian, Bilal Piot, Bo Wu, Brandon Royal, Charlie Chen, Chintu Kumar, Chris Perry, Chris Welty, Christopher A. Choquette-Choo, Danila Sinopalnikov, David Weinberger, Dimple Vijaykumar, Dominika Rogozińska, Dustin Herbison, Elisa Bandy, Emma Wang, Eric Noland, Erica Moreira, Evan Senter, Evgenii Eltyshev, Francesco Visin, Gabriel Rasskin, Gary Wei, Glenn Cameron, Gus Martins, Hadi Hashemi, Hanna Klimczak-Plucińska, Harleen Batra, Harsh Dhand, Ivan Nardini, Jacinda Mein, Jack Zhou, James Svensson, Jeff Stanway, Jetha Chan, Jin Peng Zhou, Joana Carrasqueira, Joana Iljazi, Jocelyn Becker, Joe Fernandez, Joost van Amersfoort, Josh Gordon, Josh Lipschultz, Josh Newlan, Ju yeong Ji, Kareem Mohamed, Kartikeya Badola, Kat Black, Katie Millican, Keelin McDonell, Kelvin Nguyen, Kiranbir Sodhia, Kish Greene, Lars Lowe Sjoesund, Lauren Usui, Laurent Sifre, Lena Heuermann, Leticia Lago, Lilly McNealus, Livio Baldini Soares, Logan Kilpatrick, Lucas Dixon, Luciano Martins, Machel Reid, Manvinder Singh, Mark Iverson, Martin Görner, Mat Velloso, Mateo Wirth, Matt Davidow, Matt Miller, Matthew Rahtz, Matthew Watson, Meg Risdal, Mehran Kazemi, Michael Moynihan, Ming Zhang, Minsuk Kahng, Minwoo Park, Mofi Rahman, Mohit Khatwani, Natalie Dao, Nenshad Bardoliwalla, Nesh Devanathan, Neta Dumai, Nilay Chauhan, Oscar Wahltinez, Pankil Botarda, Parker Barnes, Paul Barham, Paul Michel, Pengchong Jin, Petko Georgiev, Phil Culliton, Pradeep Kuppala, Ramona Comanescu, Ramona Merhej, Reena Jana, Reza Ardeshir Rokni, Rishabh Agarwal, Ryan Mullins, Samaneh Saadat, Sara Mc Carthy, Sarah Cogan, Sarah Perrin, Sébastien M. R. Arnold, Sebastian Krause, Shengyang Dai, Shruti Garg, Shruti Sheth, Sue Ronstrom, Susan Chan, Timothy Jordan, Ting Yu, Tom Eccles, Tom Hennigan, Tomas Kocisky, Tulsee Doshi, Vihan Jain, Vikas Yadav, Vilobh Meshram, Vishal Dharmadhikari, Warren Barkley, Wei Wei, Wenming Ye, Woohyun Han, Woosuk Kwon, Xiang Xu, Zhe Shen, Zhitao Gong, Zichuan

Wei, Victor Cotruta, Phoebe Kirk, Anand Rao, Minh Giang, Ludovic Peran, Tris Warkentin, Eli Collins, Joelle Barral, Zoubin Ghahramani, Raia Hadsell, D. Sculley, Jeanine Banks, Anca Dragan, Slav Petrov, Oriol Vinyals, Jeff Dean, Demis Hassabis, Koray Kavukcuoglu, Clement Farabet, Elena Buchatskaya, Sebastian Borgeaud, Noah Fiedel, Armand Joulin, Kathleen Kenealy, Robert Dadashi, and Alek Andreev. Gemma 2: Improving open language models at a practical size, 2024. URL https://arxiv.org/abs/2408.00118.

Llama Team. The llama 3 herd of models. *CoRR*, abs/2407.21783, 2024. doi: 10.48550/ARXIV.2407.21783. URL https://doi.org/10.48550/arXiv.2407.21783.

Yun Wang, Zhaojun Ding, Xuansheng Wu, Siyue Sun, Ninghao Liu, and Xiaoming Zhai. Autoscore: Enhancing automated scoring with multi-agent large language models via structured component recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 40898–40906, 2026.

Yuchen Wei, Dennis Pearl, Matthew Beckman, and Rebecca J Passonneau. Concept-based rubrics improve llm formative assessment and data synthesis. *arXiv preprint arXiv:2504.03877*, 2025.

Guoliang Xu and James Corter. Graph rag for automated short answer grading with feedback: bridging pedagogical needs and technical capabilities. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 40916–40924, 2026.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jian Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report. *CoRR*, abs/2505.09388, 2025. doi: 10.48550/ARXIV.2505.09388. URL https://doi.org/10.48550/arXiv.2505.09388.

Ruosong Yang, Jiannong Cao, Zhiyuan Wen, Youzheng Wu, and Xiaodong He. Enhancing automated essay scoring performance via cohesion measurement and combination of regression and ranking. In Trevor Cohn, Yulan He, and Yang Liu (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2020, Online Event, 16-20 November 2020*, volume EMNLP 2020 of *Findings of ACL*, pp. 1560–1569. Association for Computational Linguistics, 2020. doi: 10.18653/V1/2020.FINDINGS-EMNLP.141. URL https://doi.org/10.18653/v1/2020.findings-emnlp.141.

Haneul Yoo, Jieun Han, So-Yeon Ahn, and Alice Oh. DREsS: Dataset for rubric-based essay scoring on EFL writing. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 13439–13454, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.659. URL https://aclanthology.org/2025.acl-long.659/.

Jonas Zausinger, Lars Pennig, Anamarija Kozina, Sean Sdahl, Julian Sikora, Adrian Dendorfer, Timofey Kuznetsov, Mohamad Hagog, Nina Wiedemann, Kacper Chlodny, Vincent Limbach, Anna Ketteler, Thorben Prein, Vishwa Mohan Singh, Michael Danziger, and Jannis Born. Regress, don't guess: A regression-like loss on number tokens for language models. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 73995–74017. PMLR, 13–19 Jul 2025. URL https://proceedings.mlr.press/v267/zausinger25a.html.

Chenyan Zhao, Mariana Silva, and Seth Poulsen. Language models are few-shot graders. In *International Conference on Artificial Intelligence in Education*, pp. 3–16. Springer, 2025.

Honglei Zhuang, Zhen Qin, Rolf Jagerman, Kai Hui, Ji Ma, Jing Lu, Jianmo Ni, Xuanhui Wang, and Michael Bendersky. Rankt5: Fine-tuning T5 for text ranking with ranking losses. In Hsin-Hsi Chen, Wei-Jou (Edward) Duh, Hen-Hsen Huang, Makoto P. Kato, Josiane Mothe, and Barbara Poblete (eds.), *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2023, Taipei, Taiwan, July 23-27, 2023*, pp. 2308–2313. ACM, 2023. doi: 10.1145/3539618.3592047. URL https://doi.org/10.1145/3539618.3592047.

A Appendix

A.1 Numerical-Distance Blindness of Cross-Entropy

Proposition 1. Let v^* denote the target score token and let $p_v = \mathrm{softmax}(\mathbf{z})_v$. For the token-level cross-entropy loss

$$\ell_{\mathrm{sem}} = -\log p_{v^*}, \tag{14}$$

the gradient with respect to the logit z_v is

$$\frac{\partial \ell_{\mathrm{sem}}}{\partial z_v} = p_v - \mathbb{I}[v = v^*]. \tag{15}$$

Therefore, the gradient contains no explicit dependence on the numerical distance between the value represented by v and the target score.

Proof. From the derivative of the log-softmax function,

$$\frac{\partial(-\log p_{v^*})}{\partial z_v} = p_v - \mathbb{I}[v = v^*]. \tag{16}$$

Hence, two incorrect score tokens are differentiated according to their predicted probabilities, rather than their numerical distances from the target score. The same property holds token-wise for multi-token score representations.

A.2 Generative Non-Identifiability of Predictor-Based Regression

We analyze a basic property of predictor-based numerical fine-tuning. Consider an internal LLM's representation $\Phi_\theta(x) \in \mathbb{R}^q$ and a linear numerical predictor

$$\hat{y}(x) = b + w^\top \Phi_\theta(x), \tag{17}$$

with the squared-error objective

$$\ell_{\mathrm{pred}} = (y - \hat{y}(x))^2. \tag{18}$$

Proposition 1 (Representation non-identifiability). For any perturbation $\delta \in \mathbb{R}^q$ satisfying

$$w^\top \delta = 0, \tag{19}$$

the predictor output and regression loss remain unchanged:

$$\hat{y}(\Phi_\theta(x) + \delta) = \hat{y}(\Phi_\theta(x)), \tag{20}$$

and

$$\ell_{\text{pred}}\left(\Phi_\theta(x)+\delta\right)=\ell_{\text{pred}}\left(\Phi_\theta(x)\right). \tag{21}$$

Proof. By definition,

$$\hat{y}(\Phi_\theta(x)+\delta)=b+w^\top\left(\Phi_\theta(x)+\delta\right) \tag{22}$$

$$=b+w^\top\Phi_\theta(x)+w^\top\delta. \tag{23}$$

Since $w^\top\delta=0$,

$$\hat{y}(\Phi_\theta(x)+\delta)=b+w^\top\Phi_\theta(x)=\hat{y}(\Phi_\theta(x)). \tag{24}$$

Therefore, the corresponding squared-error loss is also unchanged.

Corollary 1 (Autoregressive distribution non-identifiability). Suppose that the native vocabulary logits are obtained from the same representation through

$$\mathbf{z}=E_{\text{out}}^\top\Phi_\theta(x), \tag{25}$$

and

$$p_\theta(\cdot\mid x)=\text{softmax}(\mathbf{z}). \tag{26}$$

For any perturbation δ satisfying $w^\top\delta=0$, the regression prediction is unchanged. However, if

$$E_{\text{out}}^\top\delta\notin\text{span}\{\mathbf{1}\}, \tag{27}$$

then

$$\text{softmax}\left(E_{\text{out}}^\top(\Phi_\theta(x)+\delta)\right)\neq\text{softmax}\left(E_{\text{out}}^\top\Phi_\theta(x)\right). \tag{28}$$

Therefore, identical predictor outputs and identical regression loss do not uniquely determine the LLM's native autoregressive token distribution.

A.3 Limitations of Existing Methods

A.3.1 Limitations of RAFT

We formalize a property of expected-score regression objectives such as RAFT. The key observation is that the regression loss depends on the predictive distribution only through its first moment. Consequently, distinct score distributions with the same expected value are indistinguishable under this objective.

For an input x, let $p(\cdot\mid x)$ be a probability distribution over a finite score grid $\mathcal{Y}_{\text{grid}}$, and define its expected score as

$$\mu_p(x)=\sum_{y\in\mathcal{Y}_{\text{grid}}}p(y\mid x)y. \tag{29}$$

The corresponding regression-aware objective for a target y^* is

$$\mathcal{L}_{\text{exp}}(p)=\left(y^*-\mu_p(x)\right)^2. \tag{30}$$

Proposition 1. Let p and q be two predictive distributions over $\mathcal{Y}_{\text{grid}}$. If

$$\sum_{y \in \mathcal{Y}_{\text{grid}}} p(y \mid x) y = \sum_{y \in \mathcal{Y}_{\text{grid}}} q(y \mid x) y, \tag{31}$$

then

$$\mathcal{L}_{\exp}(p) = \mathcal{L}_{\exp}(q). \tag{32}$$

In particular, whenever both distributions satisfy

$$\mu_p(x) = \mu_q(x) = y^*, \tag{33}$$

they are both global minimizers with zero regression loss, regardless of how their probability mass is distributed over the score space.

Proof. By definition, the objective depends on $p(\cdot \mid x)$ only through $\mu_p(x)$. Therefore, if $\mu_p(x) = \mu_q(x)$,

$$\begin{aligned} \mathcal{L}_{\exp}(p) &= \left(y^* - \mu_p(x)\right)^2 && (34)\\ &= \left(y^* - \mu_q(x)\right)^2 && (35)\\ &= \mathcal{L}_{\exp}(q). && (36) \end{aligned}$$

If their common expectation equals y^*, the squared error is zero, which is the minimum possible value of the objective.

A constructive example. Suppose the gold score y^* lies strictly between two candidate scores $y^- < y^* < y^+$ in $\mathcal{Y}_{\text{grid}}$. Define

$$\alpha = \frac{y^+ - y^*}{y^+ - y^-}, \qquad 1 - \alpha = \frac{y^* - y^-}{y^+ - y^-}. \tag{37}$$

Consider the distribution

$$p(y^- \mid x) = \alpha, \qquad p(y^+ \mid x) = 1 - \alpha, \tag{38}$$

with zero probability assigned to all other scores. Its expectation is

$$\begin{aligned} \mu_p(x) &= \alpha y^- + (1 - \alpha) y^+ && (39)\\ &= y^*. && (40) \end{aligned}$$

Hence,

$$\mathcal{L}_{\exp}(p) = 0, \tag{41}$$

even though the distribution may assign no probability mass to the gold score itself.

For example, if $y^* = 4$ and the score grid contains 3 and 5, then

$$p(3 \mid x) = 0.5, \qquad p(5 \mid x) = 0.5 \tag{42}$$

gives

$$\mathbb{E}_p[y] = 0.5 \times 3 + 0.5 \times 5 = 4. \tag{43}$$

Therefore, this distribution obtains the same expected-score regression loss as the concentrated distribution

$$q(4 \mid x) = 1, \tag{44}$$

although the two predictive distributions express substantially different scoring behavior.

A.3.2 A Place-Value Limitation of NTL

NTL incorporates numerical proximity into token-level training, but its standard digit-level formulation does not explicitly weight errors according to decimal place. We provide a constructive example showing that equal digit-level numerical losses can correspond to different full-score errors.

Setup. Consider an illustrative scoring setting in which 4.5, 5.5, and 4.6 are valid scores. Assume digit-level tokenization, so the target score $y^* = 4.5$ is represented as $(4, \texttt{.}, 5)$. Let p_1 and p_2 denote the predicted digit distributions at the units and tenths positions under the ground-truth prefixes, with target digit values $a_1 = 4$ and $a_2 = 5$. All punctuation and template-token contributions are identical between the configurations below and are omitted.

Using sums over the two digit positions, the numerical components of NTL-MSE and NTL-WAS are

$$\mathcal{L}_{\text{NTL-MSE}}(\mathbf{p}) = \sum_{t=1}^{2} \left(\sum_{k=0}^{9} k p_t(k) - a_t \right)^2, \tag{45}$$

$$\mathcal{L}_{\text{NTL-WAS}}(\mathbf{p}) = \sum_{t=1}^{2} \sum_{k=0}^{9} p_t(k) |k - a_t|, \tag{46}$$

where $\mathbf{p} = (p_1, p_2)$ and the Wasserstein ground cost is the absolute difference between digit values. Averaging over positions introduces a common normalization factor and does not affect the comparison.

Counterexample. Let δ_k denote a point mass on digit k, and choose $\alpha \in (1/2, 1)$. Consider the following two configurations:

$$\mathbf{p}^A = \big((1-\alpha)\delta_4 + \alpha\delta_5,\ \delta_5\big), \tag{47}$$

$$\mathbf{p}^B = \big(\delta_4,\ (1-\alpha)\delta_5 + \alpha\delta_6\big). \tag{48}$$

Configuration A places probability α on an incorrect units digit, whereas configuration B places the same probability on an incorrect tenths digit. In both cases, the incorrect digit differs from its target by one.

For NTL-MSE, the predicted digit means are $(4 + \alpha, 5)$ and $(4, 5 + \alpha)$, respectively. Therefore,

$$\mathcal{L}_{\text{NTL-MSE}}(\mathbf{p}^A) = \alpha^2 = \mathcal{L}_{\text{NTL-MSE}}(\mathbf{p}^B). \tag{49}$$

For NTL-WAS, each configuration assigns probability α to a digit at unit distance from its target, giving

$$\mathcal{L}_{\text{NTL-WAS}}(\mathbf{p}^A) = \alpha = \mathcal{L}_{\text{NTL-WAS}}(\mathbf{p}^B). \tag{50}$$

The digit-level cross-entropy losses are also equal:

$$\mathcal{L}_{\text{CE}}(\mathbf{p}^A) = -\log(1-\alpha) = \mathcal{L}_{\text{CE}}(\mathbf{p}^B). \tag{51}$$

Consequently, adding either numerical component to cross-entropy with the same fixed coefficient preserves this equality. The sparse distributions are used for clarity and can be approximated arbitrarily closely by softmax distributions with small positive probabilities on the remaining tokens.

Table 4: Comparison of deviation sources on Mohler using Qwen3-8B. The two variants achieve identical evaluation metrics at the reported precision, but differ substantially in elapsed time.

Deviation source	MAE ↓	RMSE ↓	Pearson ↑	Elapsed time
Teacher-forced reconstruction	0.32	0.57	0.84	1 h 38 min 52 s
Autoregressive reconstruction	0.32	0.57	0.84	5 h 45 min

Since $\alpha > 1/2$, taking the highest-probability token at each supervised position reconstructs $\tilde{y}^A = 5.5$ and $\tilde{y}^B = 4.6$. Their full-score absolute errors nevertheless differ by a factor of ten:

$$|\tilde{y}^A - y^*| = 1.0, \qquad |\tilde{y}^B - y^*| = 0.1. \tag{52}$$

Thus, equal digit-level training losses need not distinguish errors according to their severity in the full-score space.

Implication and Scope. The distinction arises because a unit change at the units position contributes 1 to the score, whereas the same digit change at the tenths position contributes only 0.1. This example isolates a limitation of uniformly aggregated digit-level losses without explicit place-value weighting. It does not imply that NTL lacks numerical-distance supervision or that a model trained with NTL cannot learn place values from data. The argument also does not directly apply when an entire score is represented by one token or when place-value-aware extensions are used.

A.4 Efficiency of Teacher-Forced Score Reconstruction

To examine the effect of the deviation source, we compare teacher-forced score reconstruction with free-running autoregressive decoding on Mohler using Qwen3-8B (seed 3407). The former reconstructs scores from the token-wise predictions already available in the gold-prefix-conditioned likelihood forward pass. The latter generates scores sequentially from the scoring prompt, without access to the gold score continuation. In both variants, the absolute deviation from the human score serves as a stop-gradient modulation signal for the gold-token likelihood objective.

As shown in Table 4, replacing teacher-forced reconstruction with autoregressive decoding yields no improvement in the reported MAE, RMSE, or Pearson correlation. However, autoregressive reconstruction requires approximately $3.49\times$ the elapsed time, corresponding to a 71.3% time reduction when using teacher-forced reconstruction. This efficiency difference is consistent with the additional sequential decoding required by the autoregressive variant, whereas teacher-forced reconstruction reuses the logits computed for likelihood training.

These results support teacher-forced reconstruction as an efficient training-time proxy in the evaluated setting: directly using free-running score predictions increases computational cost without improving the reported scoring metrics. We therefore adopt teacher-forced reconstruction to obtain the numerical modulation signal. This empirical comparison does not imply that the two reconstruction procedures produce identical instance-level predictions or that their deviations are generally equivalent across models and tasks.

A.5 Handling Score-Parsing Failures

During Stage II loss computation, we decode the token-wise argmax predictions from the selected prediction span and extract numerical substrings using the regular expression `-?\d+\.?\d*`. If multiple matches are found, the first match is converted to a floating-point score. Parsing failure is defined as the absence of any matching numerical substring; the parser does not perform additional range checking or clipping.

Let c_i denote the decoded prediction text, $m_{i,1}$ its first numerical match when available, and y_i the numerical score extracted from the gold response. The auxiliary score used for NDP is

$$\tilde{y}_i = \begin{cases} \mathrm{float}(m_{i,1}), & \text{if a numerical match exists in } c_i, \\ y_{max} - y_i, & \text{otherwise.} \end{cases} \tag{53}$$

In this implementation, the constant y_{max} is fixed. Accordingly, the numerical deviation for a parsing failure is

$$d_i = |\tilde{y}_i - y_i| = |y_{max} - 2y_i|. \tag{54}$$

Table 5: Supervised targets and direct NDP application in the illustrative aligned example. Target position t corresponds to the raw logit and loss column $t-1$.

Target position t	Gold token	$m_{i,t}$	Direct NDP term
0–99	Padding and preceding context	0	No
100	`assistant`	0	No
101	Newline	0	No
102	`Final`	0	No
103	`score` with leading space	0	No
104	`:`	0	No
105	Space	1	Yes
106	`4`	1	Yes
107	`.`	1	Yes
108	`5`	1	Yes
109	`<|im_end|>`	1	No

The sample is neither discarded nor regenerated, and its gold-score token labels remain unchanged. The resulting deviation is treated as a non-differentiable coefficient in the NDP likelihood surrogate and subsequently enters the existing adaptive loss-weighting procedure. The same parsing routine is used for validation-loss computation. Because the fallback depends on the gold score, it is a rule for supervised loss computation rather than a label-free inference rule.

A.6 Definitions of $m_{i,t}$, $\mathcal{P}_i$, $c_{i,t}$, and $\mathcal{R}_i$ and Gradient Realization of NDP

A.6.1 Definitions of $m_{i,t}$, $\mathcal{P}_i$, $c_{i,t}$, and $\mathcal{R}_i$

The supervision mask $m_{i,t}$, score-extraction positions $\mathcal{P}_i$, NDP computation mask $c_{i,t}$, and effective NDP region $\mathcal{R}_i$ serve different operations rather than partitioning the response into disjoint spans. Specifically, $m_{i,t}$ determines which gold tokens contribute to likelihood supervision, $\mathcal{P}_i$ specifies the ordered predictions read for numerical reconstruction, and $c_{i,t}$ selects the shared surrogate-computation slice. With $r_{i,t} = c_{i,t}m_{i,t}$, $\mathcal{R}_i = \{t : r_{i,t} = 1\}$ is the valid-target support of the local NDP gradient.

Indexing convention and illustrative example. We distinguish target-token positions from raw logit and loss columns. With causal-label shifting, logit column j predicts the target token at position $j+1$, and the corresponding token-level loss is stored at column j. Consider a single, untruncated sequence of padded length $L = 110$ with the gold response `assistant\nFinal score: 4.5<|im_end|>`. For illustration, assume the tokenization in Table 5, with the `assistant` marker at position $a_i = 100$. These are illustrative zero-based tensor indices, not tokenizer-independent boundaries.

Supervision mask. The preprocessing code masks labels before $a_i + 5$. In this example, the retained targets are therefore

$$\{t : m_{i,t} = 1\} = \{105, 106, 107, 108, 109\},$$

corresponding to the leading space, the serialized score `4.5`, and the end-of-response token. Their likelihood losses occupy raw columns 104–108. Consequently, the offsets $a_i + 5$ in label preprocessing and $a_i + 4$ in loss indexing refer to the same first supervised prediction after causal shifting. Masking the preceding labels removes their direct likelihood supervision; it does not remove the question, answer, or template from the model's conditioning context.

Score-extraction positions. For this example, the loss boundary is $b = a_i + 4 = 104$, while score reconstruction reads `logits[:, b-4:]`. Thus, the extraction window comprises raw logit columns 100–109, rather than only the columns predicting numerical digits. We take the highest-logit token at each selected position, decode the resulting sequence, and parse its first numerical substring. For example, if the decoded window begins with `\nFinal score:  4.6`, the reconstructed score is $\widetilde{y}_i = 4.6$, giving

$$d_i = |\widetilde{y}_i - y_i| = |4.6 - 4.5| = 0.1.$$

The final logit column is included in this readout even though it has no observed next-token target for likelihood supervision. This reconstruction uses teacher-forced predictions, not a freely generated autoregressive response. Accordingly, $\mathcal{P}_i$ is a prediction-readout window, not a supervision mask or a digit-only span.

NDP application region. The NDP surrogate is computed over `en_loss[:, b:-2]`, which selects raw loss columns 104–107 in this example. Every selected slot has a valid target in this aligned example. Thus, in target-token coordinates,

$$\{t : c_{i,t} = 1\} = \mathcal{R}_i = \{105, 106, 107, 108\},$$

covering the leading space and the serialized score `4.5`. The same detached deviation modulates each local likelihood gradient:

$$\nabla_\theta \ell_{i,t}^{\mathrm{ndp}} = 0.1\, \nabla_\theta \ell_{i,t}^{\mathrm{sem}}, \qquad t \in \mathcal{R}_i.$$

The supervision target remains the gold score `4.5`; the reconstructed score `4.6` supplies only the modulation signal. The final two raw loss columns are excluded because they correspond to predicting the end-of-response token and the additional ignored target introduced by causal-label shifting, respectively. The end-of-response token therefore receives no direct NDP term, although its likelihood loss remains subject to the global adaptive weighting.

Why the ranges differ. The separation follows the different roles of conditioning, numerical readout, and optimization. Likelihood supervision trains the retained score continuation and response termination without requiring the model to reproduce the preceding context. Numerical reconstruction reads a wider prediction suffix to obtain a scalar deviation through text parsing; reading a prediction does not itself assign a training target to it. NDP then applies this full-score deviation to the likelihood of the serialized score continuation, including its formatting tokens, rather than introducing separate numerical regression targets for individual digits. The particular offsets are implementation choices associated with the response template and causal indexing, not theoretical requirements of NDP.

Batched implementation. The current implementation uses a shared boundary $b = \min_i(a_i + 4)$ within each batch of padded length L. It reads predictions from `logits[:, b-4:]` and computes the surrogate over `en_loss[:, b:-2]` for all examples. In target-token coordinates,

$$c_{i,t} = \mathbf{1}\{b \le t - 1 < L - 2\}, \qquad r_{i,t} = c_{i,t} m_{i,t}.$$

When score token lengths differ, selected slots may have ignored targets ($c_{i,t} = 1$, $m_{i,t} = 0$). Their masked likelihood loss is zero, so the surrogate retains forward value d_i through $\exp(0 - \mathrm{sg}(0)) = 1$, but has zero local likelihood gradient. These slots remain in the forward aggregation and its normalization. Because every example contributes the same slice width, the forward value of $\overline{\mathcal{L}}_{\mathrm{ndp}}$ is the batch mean of d_i. This is a forward-value identity; the surrogate determines the backward signal on $\mathcal{R}_i$. Thus, aggregation uses $c_{i,t}$, while valid-target gradient support uses $r_{i,t}$.

A.6.2 Gradient realization of NDP

For an example i, the numerical deviation is

$$d_i = |\tilde{y}_i - y_i|, \tag{55}$$

where $\tilde{y}_i$ is obtained through discrete token selection and numerical parsing. Since this path contains non-differentiable operations, d_i is treated as a constant with respect to the model parameters during backpropagation.

Let $\mathrm{sg}(\cdot)$ denote the stop-gradient operator, defined by

$$\mathrm{sg}(u) = u \quad \text{in the forward pass}, \qquad \nabla_u \,\mathrm{sg}(u) = 0 \quad \text{in the backward pass}. \tag{56}$$

Let $\bar{\ell}_{i,t}^{\mathrm{sem}} = m_{i,t} \ell_{i,t}^{\mathrm{sem}}$, set to zero at ignored targets. For a selected computation slot, NuAlign defines

$$\ell_{i,t}^{\mathrm{ndp}} = \mathrm{sg}(d_i) \exp\!\left(\bar{\ell}_{i,t}^{\mathrm{sem}} - \mathrm{sg}\left(\bar{\ell}_{i,t}^{\mathrm{sem}}\right)\right), \qquad c_{i,t} = 1, \tag{57}$$

with zero values outside the computation slice.

Proposition 1. For $t \in \mathcal{R}_i$, the surrogate in Eq. equation 57 has forward value d_i and induces the gradient

$$\nabla_\theta \ell_{i,t}^{\mathrm{ndp}} = d_i \nabla_\theta \ell_{i,t}^{\mathrm{sem}}. \tag{58}$$

Proof. For $t \in \mathcal{R}_i$, $m_{i,t} = 1$, so $\bar{\ell}_{i,t}^{\mathrm{sem}} = \ell_{i,t}^{\mathrm{sem}}$. In the forward pass, the stop-gradient operator preserves its numerical value. Therefore,

$$\ell_{i,t}^{\mathrm{sem}} - \mathrm{sg}\left(\ell_{i,t}^{\mathrm{sem}}\right) = 0, \tag{59}$$

and hence

$$\ell_{i,t}^{\mathrm{ndp}} = d_i \exp(0) = d_i. \tag{60}$$

For the backward pass, define

$$g_{i,t} = \ell_{i,t}^{\mathrm{sem}} - \mathrm{sg}\left(\ell_{i,t}^{\mathrm{sem}}\right). \tag{61}$$

Because d_i is detached and $\nabla_\theta \,\mathrm{sg}(\ell_{i,t}^{\mathrm{sem}}) = 0$, we obtain

$$\begin{aligned}
\nabla_\theta \ell_{i,t}^{\mathrm{ndp}} &= \mathrm{sg}(d_i) \exp(g_{i,t}) \nabla_\theta g_{i,t} && (62)\\
&= d_i \exp(g_{i,t}) \left[\nabla_\theta \ell_{i,t}^{\mathrm{sem}} - \nabla_\theta \,\mathrm{sg}\left(\ell_{i,t}^{\mathrm{sem}}\right)\right] && (63)\\
&= d_i \exp(g_{i,t}) \nabla_\theta \ell_{i,t}^{\mathrm{sem}}. && (64)
\end{aligned}$$

Since $g_{i,t} = 0$ in the forward pass, $\exp(g_{i,t}) = 1$, yielding

$$\boxed{\nabla_\theta \ell_{i,t}^{\mathrm{ndp}} = d_i \nabla_\theta \ell_{i,t}^{\mathrm{sem}}}. \tag{65}$$

Local interpretation. Eq. equation 57 characterizes the gradient contributed by the NDP component before it is combined with semantic likelihood supervision through the adaptive cross-weighting rule. Therefore, for a fixed gold-token likelihood gradient, a larger numerical deviation produces a proportionally larger rescaling *within the NDP component*. Importantly, NDP does not introduce an independent regression-gradient direction: the update remains anchored to the likelihood gradient of the human-annotated score sequence.

For a mini-batch, define

$$K = \sum_{i,t} c_{i,t} > 0, \qquad \overline{\mathcal{L}}_{\mathrm{ndp}} = \frac{1}{K} \sum_{i,t} c_{i,t} \ell_{i,t}^{\mathrm{ndp}}.$$

Here K counts all selected computation slots, not only valid targets. Since selected ignored slots have zero local likelihood gradient and $r_{i,t} = c_{i,t} m_{i,t}$, this gives

$$\nabla_\theta \overline{\mathcal{L}}_{\mathrm{ndp}} = \frac{1}{K} \sum_{i,t} r_{i,t} d_i \nabla_\theta \ell_{i,t}^{\mathrm{sem}}. \tag{66}$$

Thus, within the NDP component, the same instance-level deviation d_i modulates all valid targets in $\mathcal{R}_i$. When $d_i = 0$, the example's local NDP contribution vanishes; the behavior of the complete objective additionally depends on the batch-level adaptive coefficient.

Table 6: Results of ablation experiments across three datasets. Bold values indicate the best performance for each metric column.

Method	Mohler			AES-ENEM			Chinese-QA		
	MAE ↓	RMSE ↓	Pearson ↑	MAE ↓	RMSE ↓	Pearson ↑	MAE ↓	RMSE ↓	Pearson ↑
Normalized	0.33	0.57	0.84	59.44	**84.86**	0.84	0.55	1.04	0.92
Unnormalized	**0.32**	**0.57**	**0.84**	**57.33**	85.24	**0.84**	**0.54**	**0.93**	**0.93**

Scope with respect to score scale. The deviation d_i is calculated in each task's native scoring units. Therefore, by construction, the absolute value of the NDP component depends on the task scale. The statement in Proposition 1 concerns how numerical deviations modulate the likelihood gradient *within a fixed scoring task*; it does not imply that the complete second-stage objective is invariant to arbitrary rescaling of numerical labels, nor does it imply that the final parameter updates will grow linearly with d_i after applying adaptive cross-weighting. To further investigate the specific performance of d_i on tasks with significantly different rating ranges, we conducted additional experiments on our three datasets using Qwen3-8B with random seed 3407. We normalized d_i for each dataset, and Table 6 shows the results. As shown in Table 6, our method performs consistently across datasets with different rating ranges.

In our experiments, we trained and evaluated each dataset separately under its predefined scoring scale, and used numerical deviations to distinguish the relative severity of scoring errors within each task. Therefore, the adaptive coefficient should be understood as a batch-dependent optimization heuristic, rather than a scale-invariant measure that can be directly compared across datasets.

A.7 Gradient Analysis of Adaptive Cross-Weighting

For compactness, define the mini-batch semantic loss as

$$S = \mathcal{L}_{\text{sem}} = \frac{1}{M} \sum_{i,t} m_{i,t} \ell_{i,t}^{\text{sem}}, \tag{67}$$

and the mean numerical-distance surrogate as

$$N = \overline{\mathcal{L}}_{\text{ndp}} = \frac{1}{K} \sum_{i,t} c_{i,t} \ell_{i,t}^{\text{ndp}}. \tag{68}$$

Here $M = \sum_{i,t} m_{i,t} > 0$ counts supervised targets, whereas $K = \sum_{i,t} c_{i,t} > 0$ counts all selected NDP computation slots, including ignored targets. Since the final objective is normalized by M, define

$$\rho = \frac{K}{M} > 0. \tag{69}$$

Although $r_{i,t} = c_{i,t} m_{i,t} \le m_{i,t}$, K counts computation slots rather than valid targets; hence, ρ may exceed one.

The adaptive coefficient used by NuAlign is

$$\lambda = \frac{N}{N+S}, \qquad 1-\lambda = \frac{S}{N+S}, \tag{70}$$

for $N + S > 0$. Because λ is not detached in our implementation, its dependence on θ must be taken into account during back-propagation.

Using the definitions above, the Stage II objective can be written as

$$\mathcal{L}_{\text{NuAlign}} = \lambda S + (1-\lambda)\frac{1}{M}\sum_{i,t} c_{i,t}\ell_{i,t}^{\text{ndp}} \tag{71}$$

$$= \lambda S + (1-\lambda)\rho N. \tag{72}$$

Proposition 2. Let $S + N > 0$ and assume that M and K are fixed for a given mini-batch. The gradient of the objective in Eq. equation 72 is

$$\boxed{\nabla_\theta \mathcal{L}_{\text{NuAlign}} = \frac{1+\rho}{(S+N)^2}\left[N^2\nabla_\theta S + S^2\nabla_\theta N\right].} \tag{73}$$

Proof. Substituting Eq. equation 70 into Eq. equation 72 gives

$$\mathcal{L}_{\text{NuAlign}} = \frac{N}{N+S}S + \frac{S}{N+S}\rho N \tag{74}$$

$$= (1+\rho)\frac{SN}{S+N}. \tag{75}$$

Differentiating Eq. equation 75 with respect to θ yields

$$\nabla_\theta \mathcal{L}_{\text{NuAlign}} = (1+\rho)\nabla_\theta\left(\frac{SN}{S+N}\right) \tag{76}$$

$$= (1+\rho)\frac{(S+N)\nabla_\theta(SN) - SN\nabla_\theta(S+N)}{(S+N)^2}. \tag{77}$$

Using

$$\nabla_\theta(SN) = N\nabla_\theta S + S\nabla_\theta N, \tag{78}$$

we obtain

$$\nabla_\theta \mathcal{L}_{\text{NuAlign}} = \frac{1+\rho}{(S+N)^2}\Big[(S+N)\left(N\nabla_\theta S + S\nabla_\theta N\right) \tag{79}$$

$$- SN\left(\nabla_\theta S + \nabla_\theta N\right)\Big] \tag{80}$$

$$= \frac{1+\rho}{(S+N)^2}\left[N^2\nabla_\theta S + S^2\nabla_\theta N\right]. \tag{81}$$

Gradient amplification. The following analysis concerns the backward gradients defined by the stop-gradient surrogate, rather than differentiation through discrete score reconstruction. For finite $S, N \geq 0$ with $S+N > 0$, both $N^2/(S+N)^2$ and $S^2/(S+N)^2$ lie in $[0,1]$. Thus, the denominator in Eq. equation 73 does not, by itself, produce an unbounded adaptive coefficient as $S+N$ becomes small. Moreover, for finite token losses, the exponential factor in the NDP surrogate equals one, rather than introducing exponential gradient amplification.

To make the bound explicit, let $G = \max_{i,t:m_{i,t}=1} \|\nabla_\theta \ell_{i,t}^{\text{sem}}\|_2$ be finite for the current mini-batch. Since $r_{i,t} = c_{i,t}m_{i,t}$ and $d_i \geq 0$, we have

$$\|\nabla_\theta S\|_2 \leq G, \qquad \|\nabla_\theta N\|_2 \leq \frac{G}{K}\sum_{i,t} r_{i,t}d_i \leq GN,$$

where the last inequality uses the forward identity $N = K^{-1} \sum_{i,t} c_{i,t} d_i$. Consequently,

$$\begin{aligned} \|\nabla_\theta \mathcal{L}_{\text{NuAlign}}\|_2 &\le (1+\rho) G \frac{N^2 + S^2 N}{(S+N)^2} \\ &\le (1+\rho) G \left(1 + \frac{S}{4}\right), \end{aligned} \tag{82}$$

using $SN/(S+N)^2 \le 1/4$. For fixed finite S, G, and ρ, increasing the deviation magnitudes alone therefore cannot make the gradient unbounded. This is a conditional bound, not a global guarantee against gradient explosion: the semantic loss, token-gradient norms, and normalization ratio can vary during training.

Zero-deviation and vanishing-gradient cases. When $N > 0$, the coefficient of $\nabla_\theta S$ in Eq. equation 73 is strictly positive. An example with $d_i = 0$ therefore contributes no local NDP gradient but remains included in likelihood supervision. However, if all deviations in the mini-batch vanish, then $N = 0$ and

$$\nabla_\theta N = \frac{1}{K} \sum_{i,t} r_{i,t} d_i \nabla_\theta \ell_{i,t}^{\text{sem}} = 0.$$

For $S > 0$, this gives

$$\lambda = 0, \qquad \mathcal{L}_{\text{NuAlign}} = 0, \qquad \nabla_\theta \mathcal{L}_{\text{NuAlign}} = 0.$$

Hence, the current objective supplies no additional gradient from such a mini-batch, even when the gold-token likelihood loss remains nonzero. This is a consequence of the weighting rule, not evidence of likelihood convergence or a division-by-zero error. More generally, for fixed $S > 0$ and bounded G and ρ, the first bound in Eq. equation 82 tends to zero as $N \to 0$. We therefore do not claim that likelihood training remains active when all numerical deviations vanish.

Numerical safeguards and Scope. The training loop invokes gradient-norm clipping with threshold 1.0 after each backward pass. For finite computed gradients, this limits the accumulated gradient norm, but it does not restore a zero gradient or repair non-finite loss values. The case $S = N = 0$ is outside the stated domain of the weighting rule and requires explicit numerical handling. Adding a positive constant to the denominator alone would avoid an undefined ratio but would not prevent the zero-gradient case at $N = 0$. Accordingly, the bounded adaptive coefficients and gradient clipping should not be interpreted as an unconditional guarantee against vanishing or exploding gradients.

Although N is expressed in the native score units of each task, cross-weighting equalizes the weighted means before applying the normalization factor ρ:

$$\lambda S = (1-\lambda) N = \frac{SN}{S+N}.$$

In the full objective, however, the numerical contribution is $(1-\lambda)\rho N = \rho \lambda S$. Thus, the forward contributions retain the count factor ρ, rather than being generally equal. This scalar relation does not imply balanced gradient magnitudes or invariance to score rescaling. Because λ remains differentiable, its parameter dependence is included in Eq. equation 73, with $\nabla_\theta N$ determined by the stop-gradient surrogate in Appendix A.6. We therefore interpret the current weighting rule as a task-local optimization heuristic rather than a scale-invariant objective.

A.8 Experiment dataset overview

A.8.1 Mohler dataset overview

The Mohler Dataset is an English-language automated short-answer grading dataset comprising 80 questions sourced from assignments in a data structures course. These questions span ten assignments and two exams and include 2,273 student responses. Two graduate students in related fields independently graded these answers using a scale from 0 (completely incorrect) to 5.



Figure 6: Question content distribution figure (showing only the top 10 most frequent questions).

Table 7: Correlation coefficients among the three annotators.

Item	Value
Expert 1 vs. Expert 2 (Pearson)	0.74
Expert 1 vs. Expert 3 (Pearson)	0.92
Expert 2 vs. Expert 3 (Pearson)	0.83

A.8.2 AES-ENEM DATASET OVERVIEW

The AES-ENEM dataset is a Portuguese-language automated essay grading dataset comprising 3,586 essays sourced from the websites Educação UOL and Brasil Escola. Two experienced annotators independently scored these essays across five dimensions, each worth up to 200 points, for a total possible score of 1,000 points.

A.8.3 CHINESE-QA DATASET OVERVIEW

We selected 5,000 data entries from our research group's projects. These entries primarily assess the language proficiency of second-language learners through question-and-answer tasks and cover a variety of scenarios: "single question, single answer," "multiple questions, multiple answers," and "contextual questions and answers." Each question is scored out of 8 points. The dataset contains 181 distinct and fully open-ended questions covering daily life, sociocultural topics, and other domains. Figure 6 shows the 10 most frequently occurring questions.

We constructed the dataset through manual annotation. First, the research team selected 5,000 candidate samples. We then randomly and independently assigned these samples to two groups of domain experts, who performed double-masked annotation based on a unified scoring standard. The experts' evaluations showed a high positive correlation. For samples with a scoring difference of ≥ 2 points, a third-party expert was invited to review and refine the annotations, ultimately resulting in the final dataset. Table 7 shows the correlation among experts. This dataset covers diverse scenarios and a broad range of questions, enabling a comprehensive and detailed assessment of candidates' overall abilities.

Additionally, our dataset complies with relevant ethics review committee regulations, does not involve any personal information, and the data annotation experts were compensated $2,800.

We cleaned three datasets for validity; the details of the final datasets are shown in Table 8.

A.9 EXPERIMENTAL DETAILS

We deploy NuAlign on 2 NVIDIA A40 GPUs (48GB VRAM) for high-performance real-time processing. To balance compute resources and efficiency, we run it on these GPUs using Python 3.12+,

Table 8: Statistical summary of the datasets.

Dataset	Language	Train size	validation size	Test size
Mohler	English	1811	226	227
AES-ENEM	Portuguese	2575	321	323
Chinese-QA	Chinese	4000	500	500

Table 9: Experimental details across all datasets.

Parameters	**Details**
Batch_size	24
Epoch_num (Each stage)	8
Learning_rate	3e-04
seed1	3406
seed2	3407
seed3	3408
Lora_rank	64
Lora_alpha	16
Lora_dropout	0.05
Temperature	0
Top_k	1
Top_p	1

PyTorch 2.6, CUDA 12.4, and random seeds 3406, 3407, and 3408. We implement asynchronous inference via vLLM to boost throughput. We use Transformers to integrate pre-trained models efficiently and simplify model management. Table 9 provides more experimental details.

A.10 PROMPT TEMPLATE

We designed a highly versatile prompt template. To ensure the method's reusability and transferability, we applied the same template to different tasks and datasets without additional task-specific rewrites or customizations; see Table 10 for details.

Table 10: Prompt template (If this dataset does not have a scoring criterion, do not include this section.)

Prompt Template

`###Task Description###`
You are a professional {`subject`}. You will assign a reasonable score to answers regarding {`subject`}-related questions/essay based on the scoring criteria, with the score not exceeding {`score_range`} points.
`###Evaluation Materials###`
Scoring criteria: {`scoring criteria`}
Question: {`question/essay`}
Pending rating response: {`answer`}
`###Output format###`
Directly provide the final score. The output format is: `Final score:` {`score`}
`###End marker and Target output###`
`Final score:`

A.11 Introduction to the Praetor, MTS, CEM-FT, RAFT, NTL and TRACT Methods and Experimental Setup

A.11.1 Introduction to the Praetor Method and Experimental Setup

Praetor is a fine-grained generative LLM evaluator trained with multi-task learning and supporting instance-level, customizable evaluation criteria. It can evaluate LLMs through single-sentence scoring or pairwise comparisons, supporting both English and Chinese languages, while offering high flexibility in setting evaluation standards.

We applied Praetor's prompt design methodology to create prompts for the Mohler and Chinese-QA datasets; see Tables 11 and 12 for specific design details. Based on the code and relevant hyperparameter settings provided by Praetor, we conducted comparative experiments. We ran experiments only on the Mohler and Chinese-QA datasets because this method currently supports only English and Chinese.

Table 11: Using the Praetor method on the prompt template for the Mohler dataset.

Prompt Template
Task Description ### You are a lecturer for a data structures course (computer science major). The system will present you with questions related to the data structures course and students' responses. Please evaluate the accuracy of the responses and grade them based on their correctness. 1. Write a detailed commentary on the answer. This commentary will be used to assess the answer's quality. 2. After completing the review, provide a decimal score between score_range representing your assessment of the answer. A higher score indicates better quality. 3. Output format: Your review of the response Overall score: Your rating for the answer 4. Generate content strictly adhering to this format. Do not add any additional opening, closing, or explanatory text. ### User question ### question ### Answer to be evaluated ### answer

Table 12: Using the Praetor method on the prompt template for the Chinese-QA dataset.

Prompt Template
Task Description ### You are a Chinese linguistics instructor. The system will present you with an open-ended question and a second-language learner's response. Evaluate the response according to the grading criteria and assign a score based on its accuracy. 1. Write a detailed commentary on the answer. Use this commentary to assess the answer's quality. 2. After completing the review, provide a decimal score between score_range representing your assessment of the answer. A higher score indicates better quality. 3. Output format: Your review of the response Overall score: Your rating for the answer 4. Generate content strictly adhering to this format. Do not add any additional opening, closing, or explanatory text. ### User question ### question ### Answer to be evaluated ### answer ### Scoring Criteria ### scoring criteria

A.11.2 Introduction to the MTS Method and Experimental Setup

MTS is a zero-shot prompting framework that automatically decomposes questions/essays into distinct dimensions and generates scoring criteria for each dimension. It then uses multi-round dialogues to guide LLMs to extract scores for each dimension. Each dialogue round completes the

scoring for its corresponding dimension based on its specific criteria, ultimately yielding a total score.

To adapt this method to our datasets, we used the prompts provided by MTS to invoke GPT-5.5, which automatically generated scoring criteria for each dimension of the AES-ENEM and Chinese-QA datasets (since no scoring dimensions were identified for the Mohler dataset, we did not conduct experiments on it). Then we ran experiments based on Qwen3-8B using random seed 3407 and the same setup as our method. The scoring criteria for each dimension of the AES-ENEM dataset are shown in Tables 13, 14, 15, 16, and 17 (these dimensions are taken from the AES-ENEM paper), while the scoring criteria for each dimension of the Chinese-QA dataset are shown in Tables 18 and 19 (these dimensions were extracted from the overall scoring criteria). We then conducted experiments using the hyperparameter configurations provided by MTS.

Table 13: Scoring criteria for the formal language dimension using the MTS method on the AES-ENEM dataset.

Formal Language
0: The language does not meet written language standards at all; it contains serious grammatical errors and frequent spelling mistakes, is extremely casual and informal, and does not meet academic or formal writing requirements.
1-40: The language contains several grammatical or spelling errors; some word choices are inaccurate or unsuitable for a formal style; the expression is somewhat chaotic or irregular, giving an overall unprofessional impression and lacking clear logic.
41-80: The language meets basic written language standards, with occasional grammatical or spelling errors that have little impact on overall understanding; word choices are relatively formal but not precise, and some expressions may appear unclear or lack professionalism.
81-120: The language is fairly standard, with occasional minor grammatical or spelling errors; overall expression is clear and basically meets the requirements of formal written language, with accurate, concise, and relatively professional word choices; the language style is consistent, though some parts may lack greater depth or refinement.
121-160: The language is very standard, with almost no grammatical, spelling, or logical errors; word choices are appropriate and accurate, fully meeting the requirements of written and academic language; expression is concise and clear, with a consistent and professional style, demonstrating a strong level of formality.
161-200: The language perfectly meets written language standards, with no grammatical or spelling errors; every word choice and sentence structure is extremely precise, showing a high degree of formality and academic quality; logic is tight, expression is clear and fluent, fully complying with written cultural standards, demonstrating exceptional language ability.

A.11.3 Introduction to the CEM-FT Method and Experimental Setup

CEM-FT framework automatically identifies high-value hard samples by analyzing the score differences between fully fine-tuned models and few-shot models, and then trains LoRA adapters on these samples to improve performance with minimal computational cost. Based on the public code provided by the authors, we conducted experiments on AES-ENEM and Chinese-QA datasets using Qwen3-8B with random seed 3407 and the same experimental settings as in our method. For the Mohler data, we directly refer to the results reported in the CEM-FT paper.

A.11.4 Introduction to the RAFT Method and Experimental Setup

RAFT resolves the mismatch between traditional autoregressive fine-tuning and numerical regression objectives: standard cross-entropy only optimizes the generation probability of tokens corresponding to correct numerical values and fails to capture the numerical distance between different incorrect predictions, while simply adding a regression prediction head would deviate from the decoder LLM's original next-token prediction paradigm. By using the full probability distribution over numerical tokens for regression-aware optimization, RAFT explicitly introduces numerical structure while preserving the LLM's autoregressive prediction framework. We replicated this method on our three datasets using Gemma2-9B, following the paper's parameter configurations.

Table 14: Scoring criteria for understanding the task dimension using the MTS method on the AES-ENEM dataset.

Understanding the Task
0: Completely does not understand the task requirements, and the response content is entirely off-topic. No relevant knowledge is applied, the content is incoherent, and it lacks a reasonable logical structure.
1-40: Has a shallow understanding of the task; part of the response is relevant, but it lacks a deep understanding of key concepts. Knowledge application is insufficient, reasoning is confused, and the structure is loose or unclear, failing to support the main argument effectively.
41-80: Has some understanding of the task, but still not deep enough. Able to apply some related knowledge and develop reasonable arguments, but reasoning is inadequate, the structure is somewhat messy or loose, and the organization is not optimal within the constraints of an essay.
81-120: Has a good understanding of the task and can effectively analyze and argue using relevant knowledge. The theme is fairly well developed, reasoning is strong, the structure is reasonable, and the content is well organized within the given framework. However, there is room for improvement, such as depth or comprehensiveness.
121-160: Has a very deep understanding of the task and can accurately apply knowledge from various disciplines for analysis, fully developing arguments. Reasoning is clear and strong; the structure is tight, showing strong logic and coherence. Completes the task within the essay structure while also deepening the discussion of the topic.
161-200: Has an extremely thorough understanding of the task and can integrate concepts from multiple disciplines for profound analysis, developing arguments comprehensively from multiple angles. Reasoning is comprehensive and precise, logic is rigorous and deep, and the theme is perfectly presented within structural constraints. The text structure is flawless, the content is rich and highly persuasive, and it fully demonstrates a high level of academic understanding and critical thinking skills.

Table 15: Scoring criteria for organizing the information dimension using the MTS method on the AES-ENEM dataset.

Organization of Information
0: The selected information is completely irrelevant or misleading, fails to connect different pieces of information effectively, and lacks structure. There is no clear argument or viewpoint, and the reasoning cannot support any perspective.
1-40: The selection of information is relatively random, lacks focus, is poorly connected, and includes irrelevant or repetitive information. The organization is loose, the content lacks clear hierarchy or logic, and some arguments are not strongly supported.
41-80: The selection of information is basically reasonable, but there are still some irrelevant or redundant contents. The connection between information is preliminary; the structure is somewhat loose; some arguments are basically supported, but the overall reasoning is insufficient, and there is room for improvement.
81-120: The selection of information is clear and mostly relevant, connections are reasonable, and key information can be effectively organized and presented. The overall structure is clear, logic is coherent, and the arguments are fairly well supported, though some reasoning may still be insufficiently in-depth or have minor gaps.
121-160: The selection of information is precise and relevant, connections are reasonable and tight, effectively supporting the arguments. The structure is clear, content is well-organized, reasoning is strong, fully supporting the position with information, facts, and viewpoints, and the expression is very clear.
161-200: The selection of information is perfect, with all information closely connected and precisely supporting the arguments. The structure is rigorous and layered, with in-depth and persuasive reasoning. The integration and organization of information demonstrate strong logic and critical thinking. The arguments are strongly supported, showing high-level writing skills.

Table 16: Scoring criteria for the knowing argumentation dimension using the MTS method on the AES-ENEM dataset.

Knowing Argumentation
0: Shows no understanding of the language mechanisms of argumentation; the argument lacks clear structure, logic is chaotic, reasoning is unsupported or contains obvious logical errors. Language mechanisms are used improperly, failing to effectively convey the argument.
1-40: The argument structure is confusing or incomplete, reasoning lacks logic, and evidence is insufficient. The use of language mechanisms is somewhat stiff, reasoning or refutation is ineffective, and the argument lacks persuasiveness.
41-80: The argument structure is basically clear, but there are still some logical gaps or imperfect reasoning. Able to use basic language mechanisms, such as causality or analogy, but not fully developed; the refutation part is still insufficient.
81-120: The argument structure is relatively clear, and reasoning is fairly rigorous. Language mechanisms are used appropriately, effectively employing causality, comparison, etc., to support points. Able to respond to counterarguments, enhancing the depth and persuasiveness of the argument, though some parts may still lack depth.
121-160: The argument structure is clear and rigorous, logical reasoning is solid, and evidence is strong. Able to skillfully use language mechanisms such as causality, analogy, comparison, and refutation, enhancing the persuasiveness and depth of the argument. Handles counterarguments very reasonably, demonstrating strong argumentation ability.
161-200: The argument structure is extremely clear and rational, reasoning is flawless, points are fully supported by evidence, and logic is rigorous and perfect. Language mechanisms are used with ease, flexibly employing multiple mechanisms (such as refutation, causality, analogy, etc.) to effectively support the argument. Responses to opposing views are very strong, showing an exceptionally high level of argumentation.

Table 17: Scoring criteria for the solution proposal dimension using the MTS method on the AES-ENEM dataset.

Solution Proposal
0: The solution lacks innovation or is completely infeasible, ignores human values, has negative impacts on individuals or groups, completely disregards social and cultural diversity, and has a negligible effect on the problem.
1-40: The solution lacks effectiveness or practical feasibility, has certain negative impacts or biases, fails to adequately respect human values or the diversity of social cultures, provides insufficient details, and is difficult to implement.
41-80: The solution has some innovation and feasibility but may face challenges during implementation. It respects basic human values but does not fully consider social and cultural diversity. Implementation details are somewhat lacking, and it may have adverse effects on certain groups.
81-120: The solution is fairly innovative and considers practical feasibility while providing some implementation details. It adequately respects basic human values and makes efforts to consider social and cultural diversity. It can be effectively implemented in diverse cultural contexts, but may still require further optimization or improvement.
121-160: The solution is innovative and highly feasible, with thorough details and consideration of potential implementation obstacles. It respects and embodies basic human values and has good adaptability to cultural diversity, allowing effective application across different cultural contexts.
161-200: The solution is extremely innovative and practically feasible, with detailed and comprehensive consideration of all possible obstacles, and proposes concrete, workable countermeasures. It fully respects core human values and has high adaptability in global or multicultural contexts, effectively reflecting social and cultural diversity, and has a very significant impact on solving the problem.

Table 18: Scoring criteria for the completeness-of-response dimension using the MTS method on the Chinese-QA dataset.

Completeness of Response
0: Unable to answers or provides completely irrelevant answers. **1-3**: Barely answers some questions; answers are incomplete, sentences are disjointed or lack key information; frequent repetitions prevent clear communication of all key points. **4-6**: Answers all questions but with simple structure; answers lack detail or fail to fully cover some points. Expression is relatively concise, with occasional repetitions, or grammatical errors. **7-8**: Effectively answers all questions using one or more complete sentences, with substantial and detailed content.

Table 19: Scoring criteria for the answer completeness dimension using the MTS method on the Chinese-QA dataset.

Answer Completeness
0: The answer is completely off-topic or does not address the question at all. **1-3**: The answer has a weak connection to the question and contains some irrelevant information. **4-6**: The answer is generally on topic, conveying the main information, but may be not fully aligned with the question. **7-8**: The answer is entirely on topic, clearly and accurately expressed, and closely aligned with the core of the question.

A.11.5 Introduction to the NTL Method and Experimental Setup

NTL is a regression-like loss function that operates entirely at the token level. NTL offers two variants: one that minimizes the L_p norm between the actual token values and the predicted token values, and another that minimizes the Wasserstein distance. By introducing numerical proximity at the token level, NTL enhances the language model's inherent ability to process numerical data. Since this loss function is available as a library, we conducted experiments on three datasets using Qwen3-8B, with experimental settings consistent with our own.

A.11.6 Introduction to the TRACT Method and Experimental Setup

TRACT combines thought-chain reasoning with regression-aware training. The TRACT training objective has two parts: a cross-entropy loss for learning thought-chain reasoning and a regression-aware loss for score prediction. TRACT has two training phases: the first fine-tunes a base large model to generate thought chains; the second retrains the base large model using the thought chains generated in the first phase. We used Mistral-7B-Instruct and reproduced the experimental results on three datasets using its specified parameter settings.