SU NLP 29 at SemEval-2026 Task 5: DynaOrd - Hybrid Dynamic Ordinal Regression with LoRA-Fine-Tuned DeBERTa-v3

Musab Ahmed Khan
Sabancı University, İstanbul, Turkey
musab.khan@sabanciuniv.edu

Abstract

We describe our system submitted to SemEval-2026 Task 5 on rating the plausibility of word senses in ambiguous sentences within narrative contexts. The task requires predicting human-perceived plausibility scores on a 1–5 scale for candidate word meanings embedded in short stories, posing challenges such as limited training data and the ordinal nature of target labels. Our approach combines a DeBERTa-v3-large encoder with Low-Rank Adaptation (LoRA) and a dynamically weighted hybrid CORAL–MSE loss for ordinal regression. This formulation adapts the contribution of ranking and regression objectives during training, prioritizing ordinal consistency early and regression refinement in later epochs. We analyze the contributions of dynamic loss weighting to overall system performance.

1 Introduction

SemEval-2026 Task 5 (Gehring et al., 2026) focuses on rating the plausibility of word senses in ambiguous sentences within narrative contexts. Given a short story containing a homonymous word and a candidate word sense definition, systems must predict a plausibility score on a continuous 1–5 scale reflecting human judgment. This task lies at the intersection of word sense disambiguation (Navigli, 2009) and narrative understanding, requiring systems to integrate lexical knowledge with sentence-level and discourse-level contextual cues. Systems are evaluated using Spearman rank correlation and accuracy within one standard deviation of human ratings.

We approach this problem as ordinal regression, treating the plausibility scale as an ordered set of categories where label distances are meaningful. Our system leverages DeBERTa-v3-large (He et al., 2023) as the base encoder, adapted efficiently via Low-Rank Adaptation (LoRA; Hu et al., 2022) to accommodate limited GPU memory. Instead of relying on a single training objective, we combine CORAL ordinal loss (Cao et al., 2020) with MSE regression loss and dynamically adjust their relative contributions throughout training.

Participating in this shared task revealed that the balance between ordinal ranking and regression objectives can substantially affect model stability and generalization. In particular, we observed that dynamically shifting emphasis from ordinal ranking toward regression as training progresses consistently outperformed static loss combinations during development, and appeared more resilient to the precise choice of loss coefficients. Our final system ranked 14th among 27 teams on the final paper-submission leaderboard, demonstrating that parameter-efficient fine-tuning with a carefully designed loss schedule can achieve competitive performance even under constrained computational resources. Our code is publicly available.[1]

2 Background

Each instance in the dataset consists of a homonymous word, a candidate word sense definition, a narrative context (comprising a precontext, a target sentence containing the ambiguous word, and an optional ending), and a dictionary example sentence for the sense. Five human annotators independently rate the plausibility of the candidate sense on a 1–5 integer scale; the gold label is the average of these ratings, accompanied by their standard deviation. Figure 1 shows an example instance.

The dataset contains 2,280 training, 588 development, and 930 test instances in English. Systems are evaluated on two metrics: (1) Spearman rank correlation (ρ), which measures the monotonic relationship between predicted and gold scores, and (2) accuracy within standard deviation (ACC_{SD}), defined as the proportion of predictions falling within one standard deviation of the gold mean

[1] https://github.com/NLP-445-SemEval/SU-NLP-29

Proceedings of the The 20th International Workshop on Semantic Evaluation (2026), pages 514–521

Word: inflation
Sense: *the act of filling something with air*
Precontext: Jack loved maintaining his old car. He noticed things had been getting a lot more expensive lately. Last week, he found a good deal on a pump at the auto shop.
Target: The tires on the car were affected by **inflation**.
Ending: The front driver-side one had blown up to nearly 3 ft tall.
Rating: 3.4 **SD:** 1.82

Figure 1: Example training instance. The precontext mentions both car maintenance and rising prices, creating genuine ambiguity around "inflation." The ending resolves toward the physical sense, yet the high SD reflects substantial annotator disagreement.

rating. The official ranking uses the arithmetic mean of these two metrics.

Ordinal regression has been addressed using ranking-based losses such as CORAL (Cao et al., 2020), which decomposes the problem into a series of binary classification tasks with shared weights. Regression-based approaches typically rely on MSE loss but ignore the ordinal structure of labels. Our work combines these paradigms through a dynamically weighted hybrid objective.

3 System Overview

3.1 Model Architecture

We use DeBERTa-v3-large (He et al., 2023) as our base encoder, a 304-million parameter model pretrained with disentangled attention and an enhanced mask decoder. To reduce memory requirements, we employ LoRA (Hu et al., 2022) for parameter-efficient fine-tuning, applying low-rank decomposition to the query, key, value, and dense projection layers with rank r=32 and scaling factor α=128. This reduces the number of trainable parameters to about 6.3 million (approx. 2.1% of the 304M-parameter backbone).

Input formatting. Each instance is converted into a structured narrative input that provides the model with hierarchical context at multiple levels:

```
Ambiguous word: [word]. Candidate
sense: [meaning]. Sense tags: [tags].
Dictionary example: [example]. Story
context: [precontext]. Target sentence:
[sentence]. Ending: [ending].
```

The ambiguous word in the target sentence is highlighted with double asterisks (e.g., `**inflation**`) to provide an explicit signal of the homonym's position; implementation details are given in Appendix E.1. This format supplies word-level context (the sense definition and dictionary example), sentence-level context (the target sentence), and discourse-level context (the story precontext and ending) in a single input sequence.

Pooling. Rather than using only the `[CLS]` token representation, we employ weighted attention pooling over all token hidden states:

$$\alpha_i = \frac{\exp(\mathbf{w}^\top \mathbf{h}_i + b)}{\sum_{j=1}^{n} \exp(\mathbf{w}^\top \mathbf{h}_j + b)} \tag{1}$$

$$\mathbf{h}_{\text{pool}} = \sum_{i=1}^{n} \alpha_i \, \mathbf{h}_i \tag{2}$$

where $\mathbf{h}_i$ is the hidden state at position i and $\mathbf{w} \in \mathbb{R}^d$, $b \in \mathbb{R}$ are learned parameters. This allows the model to attend to multiple informative tokens, such as the homonym, the sense definition, and contextual cues, rather than relying solely on the `[CLS]` representation. The pooled representation is passed through a dropout layer before the prediction head.

3.2 Ordinal Regression Formulation

We model the task as ordinal regression, where label distances are meaningful but discrete. This formulation allows us to capture both absolute prediction error and relative ordering between instances.

For K=5 ordinal classes, the CORAL framework (Cao et al., 2020) decomposes the prediction into $K-1$=4 binary classification tasks, each predicting whether the rating exceeds threshold k. The model uses a shared weight vector $\mathbf{w}$ with class-specific bias terms b_k:

$$P(Y > k \mid \mathbf{x}) = \sigma(\mathbf{w}^\top \mathbf{h}_{\text{pool}} + b_k) \tag{3}$$

The final predicted rating is obtained by summing the threshold probabilities:

$$\hat{y} = 1 + \sum_{k=1}^{K-1} \sigma(\mathbf{w}^\top \mathbf{h}_{\text{pool}} + b_k) \tag{4}$$

This formulation inherently respects the ordinal structure: the shared weight vector ensures that threshold predictions are consistent across adjacent classes.

Since the gold labels are continuous averages of human ratings, we use soft binary targets rather than hard thresholds:

$$t_k = \text{clamp}(y - k,\ 0,\ 1) \tag{5}$$

For example, a gold rating of y=3.6 produces targets [1.0, 1.0, 0.6, 0.0], naturally encoding the fractional position between ordinal classes.

3.3 Dynamic Hybrid Loss

The CORAL loss alone optimizes ordinal consistency but does not directly minimize absolute prediction error. Conversely, MSE loss minimizes absolute error but ignores the ordinal structure of labels. We combine both objectives through a hybrid loss with epoch-dependent weights. This design is related to curriculum and multi-task loss weighting methods that adapt training signals over time (e.g., Bengio et al., 2009; Kendall et al., 2018; Chen et al., 2018), but here we use a simple linear schedule tailored to the CORAL–MSE combination for ordinal regression.

The CORAL loss is the binary cross-entropy over the $K-1$ ordinal thresholds:

$$\mathcal{L}_{\text{CORAL}} = \frac{1}{K-1} \sum_{k=1}^{K-1} \text{BCE}(t_k, \sigma(l_k)) \quad (6)$$

where l_k denotes the k-th logit and t_k is the soft target from Equation 5. The MSE component operates on the converted prediction: $\mathcal{L}_{\text{MSE}} = (y - \hat{y})^2$, where $\hat{y}$ is obtained via Equation 4.

The total loss is:

$$\mathcal{L} = \alpha(e) \cdot \mathcal{L}_{\text{CORAL}} + \beta(e) \cdot \mathcal{L}_{\text{MSE}} \quad (7)$$

where the weights vary linearly across epochs:

$$\alpha(e) = \alpha_0 + (\alpha_T - \alpha_0) \cdot \frac{e}{T} \quad (8)$$

$$\beta(e) = \beta_0 + (\beta_T - \beta_0) \cdot \frac{e}{T} \quad (9)$$

with α_0=1.0, α_T=0.7, β_0=0.1, β_T=0.4, and T being the total number of training epochs.

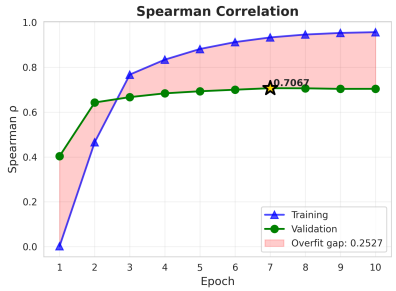
This schedule assigns high weight to the CORAL loss initially, encouraging the model to first establish correct ordinal rankings. As training progresses, the MSE weight increases to refine absolute prediction values. Unlike static weighting, this approach adapts the training signal to the model's learning stage: early epochs benefit from ordinal constraints that structure the output space, while later epochs benefit from regression pressure that fine-tunes predictions within that structure.

3.4 Training Procedure

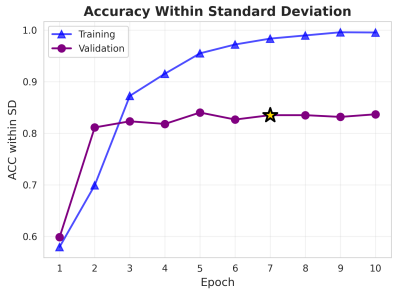
We train with AdamW (Loshchilov and Hutter, 2019) using a cosine learning rate schedule with a warmup phase covering the first 5% of training steps. Training employs mixed-precision (FP16) to reduce memory consumption on a single GPU. We use gradient accumulation over 8 steps with a micro-batch size of 2, yielding an effective batch size of 16. Gradient norms are clipped to 1.0.

Early stopping is applied based on a combined metric on the development set: the harmonic mean of Spearman correlation and ACC_{SD}, with a patience of 3 epochs. We use the harmonic mean rather than the arithmetic mean because, analogously to the F_1 score in information retrieval (Manning et al., 2008), it penalizes checkpoints that are strong on one metric at the expense of the other, thereby encouraging balanced optimization across both criteria.

Figure 2 shows the training dynamics for both metrics: validation Spearman and ACC_{SD} are plotted alongside their training-set counterparts, with the best checkpoint selected by the combined early-stopping criterion before the validation–training gap widens. Additional training curves for loss are provided in Appendix A.



(a) Spearman.



(b) ACC_{SD}.

Figure 2: Training and validation Spearman correlation and ACC_{SD} across epochs. The shaded region in the Spearman plot indicates the overfitting gap; stars mark the best checkpoint selected by the combined early-stopping criterion.

4 Experimental Setup

Hyperparameters were tuned on the development set. Table 1 lists the key settings for our final sys-

Hyperparameter	Value
Learning rate	8×10^{-5}
Batch size	2
Grad. accum. steps	8 (eff. batch 16)
Max epochs	10
Early stopping patience	3
LoRA rank (r)	32
LoRA scaling (α)	128
LoRA dropout	0.05
Classifier dropout	0.4
Weight decay	0.15
Warmup ratio	0.05
Scheduler	Cosine
Max sequence length	512
CORAL $\alpha_0 \rightarrow \alpha_T$	$1.0 \rightarrow 0.7$
MSE $\beta_0 \rightarrow \beta_T$	$0.1 \rightarrow 0.4$
Random seed	42

Table 1: Hyperparameters for our final system.

tem; all model selection and hyperparameter tuning were performed on the development set, with the test set used only for final evaluation. Development-set ablations are reported in Appendix B.

We implement our system using HuggingFace Transformers (Wolf et al., 2020) and PEFT libraries. For detailed library requirements, see Appendix F. All experiments are conducted on a single NVIDIA RTX 4060 GPU with 8 GB VRAM. The CORAL bias terms are initialized with linearly spaced values in $[-1, 1]$ to encourage predictions across the full rating range from the start of training.

5 Results and Analysis

5.1 Main Results

Table 2 presents the results of our official submission alongside post-submission ablation variants evaluated on the released test set; implementation details of the loss-weight initialization are given in Appendix E.3. All post-submission configurations were run under identical conditions to the official system to assess the contribution of individual design choices and to identify the closest alternatives to the submitted model.

We include static variants with CORAL weights of 0.9 and 0.8, which span the middle range of DynaOrd's dynamic schedule ($1.0 \rightarrow 0.7$), to directly compare fixed-weight configurations against our dynamic approach. Our official submission, DynaOrd, achieves a Spearman correlation of 0.68, accuracy within standard deviation of 0.78, and a combined score of 0.73, ranking 14th in the final leaderboard. Among all configurations we evaluated, including both the officially submitted system and post-submission variants, DynaOrd obtains the highest ρ and combined score, while outperforming its static variants in all metrics.

5.2 Ablation Study

All ablation variants share the same architecture, hyperparameters, and training procedure; only the loss configuration differs.

Both single-loss baselines underperform all hybrid variants, suggesting that combining ordinal and regression objectives is beneficial for this task. Replacing dynamic weighting (DynaOrd) with static configurations in our ablation study consistently reduces ρ by approximately 0.006–0.007 and ACC_{SD} by about 0.003–0.005, indicating that the scheduling itself matters beyond the choice of fixed weight values. Inverting the schedule direction (starting MSE-heavy and shifting toward CORAL) causes the largest drop among hybrid variants ($\Delta\rho = -0.037$), suggesting that establishing ordinal structure early may be important for downstream ranking performance in our experiments. The conservative and aggressive variants, which correspond to milder and more aggressive versions of the same dynamic schedule, bracket DynaOrd on both sides ($\Delta\rho = -0.010$ and -0.012), suggesting that our chosen end weights are effective within the explored range. Substituting Huber for MSE with identical weights reduces ρ by 0.032, indicating that MSE may provide a more suitable regression signal for this task. While carefully tuned static configurations (v1, v2) narrow the gap, a less balanced static weighting (v3) drops sharply to a combined score of 0.43, compared to DynaOrd's 0.73, while all dynamic variants remain above 0.70. This contrast suggests that the dynamic schedule provides meaningful resilience to the choice of loss coefficients, reducing the risk associated with a poorly calibrated fixed configuration.

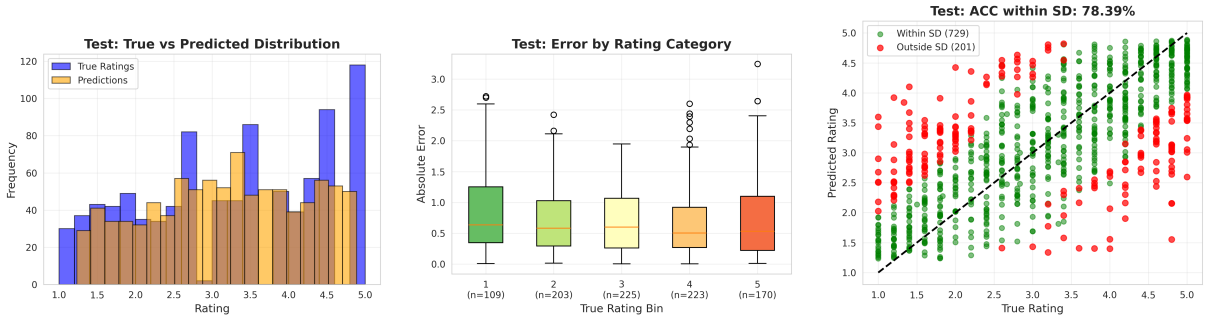
5.3 Error Analysis

Figure 3 presents test set prediction analysis across three complementary views.

The predicted-vs-true distribution (Figure 3a) reveals that our model produces a flatter, more uniform output distribution compared to the bimodal gold distribution, which peaks near ratings 2.5–3 and 4.5–5. The model underpredicts extreme ratings, particularly the strong peak at 5.0, and overpredicts the middle range, consistent with a regression-to-the-mean tendency commonly observed in ordinal regression systems.

Configuration	α (CORAL)	β (Regression)	ρ	ACC_{SD}	Combined
Single-loss baselines					
CORAL only	1.1	0.0	0.6458	0.7581	0.7020
MSE only	0.0	1.1	0.6692	0.7710	0.7201
Hybrid dynamic (CORAL + MSE)					
DynaOrd (Final)	$1.0 \rightarrow 0.7$	$0.1 \rightarrow 0.4$	**0.6825**	0.7839	**0.7332**
Conservative	$1.0 \rightarrow 0.8$	$0.1 \rightarrow 0.3$	0.6721	**0.7925**	0.7323
Aggressive	$1.0 \rightarrow 0.6$	$0.1 \rightarrow 0.5$	0.6705	0.7828	0.7267
Inverted	$0.1 \rightarrow 0.4$	$1.0 \rightarrow 0.7$	0.6454	0.7548	0.7001
DynaOrd (Huber)	$1.0 \rightarrow 0.7$	$0.1 \rightarrow 0.4$	0.6510	0.7796	0.7153
Hybrid static (CORAL + MSE)					
Static v1	0.9	0.2	0.6760	0.7785	0.7273
Static v2	0.8	0.3	0.6764	0.7806	0.7285
Static v3	0.6	0.5	0.3321	0.5312	0.4317

Table 2: Test set results and ablation study. α/β: CORAL and regression (MSE or Huber) weight schedules (start $\rightarrow$ end for dynamic, fixed values for static). ***DynaOrd (Final)*** is the official Codabench submission; all other configurations are post-submission ablations run on the released test set under identical conditions. DynaOrd (Huber) replaces MSE with Huber loss using identical weights. Bold values indicate best per metric column. Bold-italic rows highlight the key comparison: DynaOrd (dynamic scheduling) vs. static variants (fixed weights).



(a) True vs. predicted rating distributions. (b) Absolute error by rating category. (c) Predictions within/outside one SD.

Figure 3: Test set prediction analysis for DynaOrd. (a) The model's predictions are more uniform than the bimodal gold distribution. (b) Median errors are stable across rating bins; extreme ratings show more outliers. (c) 78.4% of predictions fall within one SD of the human mean; errors concentrate at scale extremes.

The error-by-rating boxplot (Figure 3b) shows that absolute errors are relatively stable across rating categories, with median errors between 0.5 and 0.65. Rating 1 exhibits the highest error variability (largest interquartile range), which may reflect the fact that low-plausibility instances often involve subtle semantic mismatches that are difficult to distinguish based on surface-level cues. Ratings 4 and 5 show more outliers with large errors, corresponding to cases where the model substantially underestimates highly plausible senses.

The ACC within SD scatter (Figure 3c) shows that 729 of 930 test predictions (78.4%) fall within one standard deviation of the human mean. The 201 outside-SD errors (red) are concentrated at the extremes of the rating scale and along class boundaries, suggesting that annotation ambiguity and inter-annotator disagreement may account for a portion of these errors. Manual inspection of a small sample of test predictions supports this: the model sometimes underestimates highly plausible senses (e.g., gold 4.8 for "crop" in an agricultural context predicted 1.6; gold 4.8 for "bars" in a Dublin pubs context predicted 2.2) and overestimates low-plausibility ones (e.g., gold 1.2 for (bowling) "pin" in a workshop context, predicted 3.9; gold 1.0 for (musical) "bars" in a pub context, predicted 3.6). Instances with high annotator variance (e.g., stdev 1.22) often show larger prediction errors, consistent with genuine ambiguity in the gold labels rather than pure model failure.

5.4 Limitations

All experiments reported in this paper use a single random seed (42), so we do not have estimates of variance across runs. The training curves (Figure 2) show an observable overfitting gap: validation

Spearman and ACC_{SD} plateau while training metrics continue to improve, and early stopping mitigates but does not fully close this gap. Additionally, the gains of the dynamic schedule over the two best-tuned static configurations (Static v1 and v2) are modest in terms of ρ (approximately 0.006–0.007) and should be interpreted as suggestive rather than conclusive without multi-seed replication; the more notable advantage of dynamic scheduling lies in its robustness to coefficient choices, as less carefully tuned static configurations degrade substantially (see §5.2). These constraints stem from limited computational resources, which also precluded ensembling and larger batch sizes.

6 Conclusion

We presented SU NLP 29's system for SemEval-2026 Task 5, based on DeBERTa-v3-large with LoRA adaptation and a dynamically weighted hybrid CORAL–MSE loss for ordinal regression. Despite its simplicity, our approach remains competitive with more complex systems, ranking 14th on the final official leaderboard and performing within 18% of the top-performing system. Our ablation study suggests that dynamic hybrid weighting outperforms both single-loss baselines and static combinations across all configurations tested, though robustness across seeds and splits remains an open question.

Future work includes exploring ensemble methods and multi-seed training when additional compute is available, implementing an LLM-generated dataset augmentation pipeline to mitigate overfitting and significantly boost performance, and investigating more sophisticated loss schedules (e.g., learned or validation-driven weighting) beyond fixed linear interpolation.

Acknowledgments

We thank Dr. Dilara Keküllüoğlu for insightful feedback and constructive comments that helped improve the clarity and quality of this work. We are also grateful to the anonymous reviewers for their constructive feedback and suggestions.

References

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum learning. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 41–48, Montreal, Canada.

Wenzhi Cao, Vahid Mirjalili, and Sebastian Raschka. 2020. Rank consistent ordinal regression for neural networks with application to age estimation. *Pattern Recognition Letters*, 140:325–331.

Zhao Chen, Vijay Badrinarayanan, Chen-Yu Lee, and Andrew Rabinovich. 2018. Gradnorm: Gradient normalization for adaptive loss balancing in deep multitask networks. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80, pages 793–802.

Janosch Gehring, Selina Meyer, and Michael Roth. 2026. SemEval-2026 task 5: Rating plausibility of word senses in ambiguous stories through narrative understanding. In *Proceedings of the 20th International Workshop on Semantic Evaluation*, San Diego, California. Association for Computational Linguistics.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2023. DeBERTaV3: Improving DeBERTa using ELECTRA-style pre-training with gradient-disentangled embedding sharing. In *International Conference on Learning Representations*.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Alex Kendall, Yarin Gal, and Roberto Cipolla. 2018. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7482–7491.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. 2008. *Introduction to Information Retrieval*. Cambridge University Press.

Roberto Navigli. 2009. Word sense disambiguation: A survey. *ACM Computing Surveys*, 41(2):1–69.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, and 1 others. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45.

A Additional Training Curves

Figure 4 shows the loss curves across training epochs. Validation loss plateaus around epoch 5 while training loss continues to decrease, consistent with the overfitting pattern visible in the Spearman and ACC_{SD} curves (Figure 2).

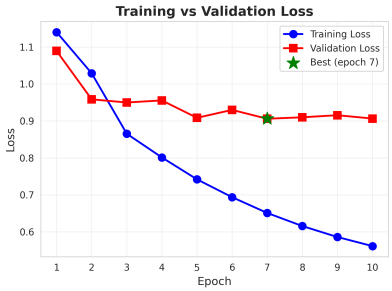


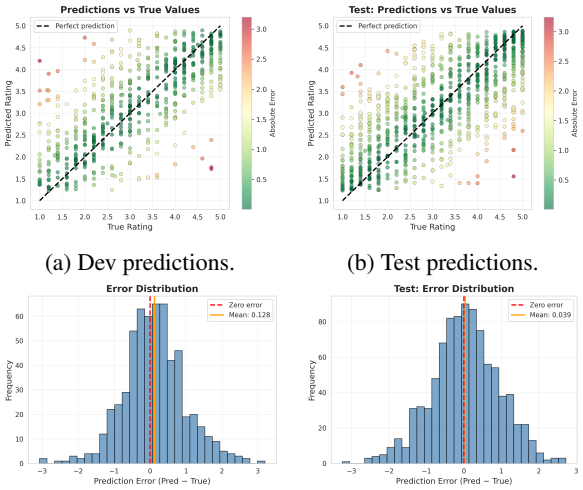
Figure 4: Training vs. validation loss. The star marks the best checkpoint (epoch 7) selected by the combined metric.

Config	ρ	ACC_{SD}	Comb.
Single-loss baselines			
CORAL only	0.71	0.83	0.77
MSE only	0.71	0.83	0.77
Hybrid Dynamic			
DynaOrd (Final)	0.71	0.84	0.77
Conservative	0.71	0.84	0.77
Aggressive	0.71	**0.86**	**0.78**
Inverted	0.68	0.83	0.76
DynaOrd (Huber)	0.68	0.81	0.74
Hybrid Static			
Static v1	0.71	0.84	0.78
Static v2	0.69	0.84	0.77
Static v3	0.30	0.57	0.43

Table 3: Development set ablation results (588 samples). Same configurations as Table 2. Bold: best per metric.

B Development Set Ablation

Table 3 reports the same ablation configurations as Table 2, evaluated on the development set (588 samples). The overall ranking of configurations is largely preserved, though Static v1 matches DynaOrd's combined score (0.78) on the dev set, unlike on the test set where DynaOrd clearly outperforms all static variants. The Aggressive schedule achieves the highest dev ACC_{SD} (0.86) and combined score (0.78), unlike on the test set where DynaOrd leads. This discrepancy may reflect the smaller dev set size or the early-stopping criterion selecting different checkpoints. Notably, DynaOrd's dev ρ (0.7067) is higher than its test ρ (0.6825), consistent with the checkpoint being selected on dev performance. The Inverted and Huber variants remain the weakest performers on both splits, reinforcing the finding that CORAL-dominant scheduling with MSE regression yields the most robust results.



(a) Dev predictions. (b) Test predictions.

(c) Dev error dist. (d) Test error dist.

Figure 5: Dev vs. test comparison. Top: prediction scatter plots colored by absolute error. Bottom: error distributions. Both splits show consistent patterns.

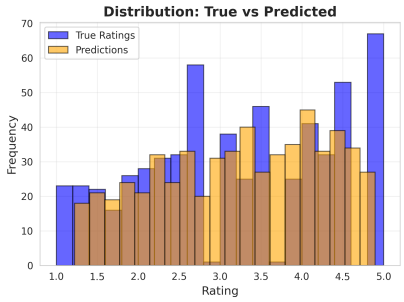


Figure 6: Development-set true vs. predicted rating distributions.

C Development vs. Test Set Comparison

Figure 5 compares prediction behavior across the development and test splits.

The scatter plots show similar dispersion patterns: both splits exhibit a positive trend along the diagonal with increased error at the extremes of the rating scale. The error distributions are both approximately centered at zero with a slight negative skew, indicating a mild tendency to overpredict. The consistency between splits suggests stable generalization rather than overfitting to development-specific characteristics.

D Development Set Analysis

Figures 6–8 present prediction analysis on the development set (588 samples), mirroring the test set analysis in Figure 3.

The patterns are consistent across both splits: the model produces a flatter distribution than the bimodal gold labels, with errors concentrated at

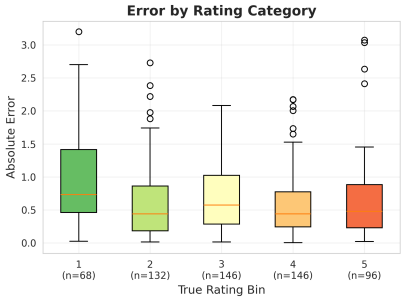


Figure 7: Development-set absolute error by rating category.

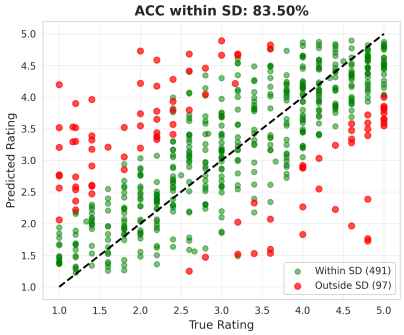


Figure 8: Development-set predictions within/outside one SD.

rating extremes. Rating 1 exhibits the highest error variability on both splits, while ratings 4–5 show the most outliers with large absolute errors.

E Additional Implementation Details

E.1 Target Word Marking

The ambiguous word occurrence in the target sentence is explicitly marked by surrounding it with double asterisks (for example, `**inflation**`) prior to tokenization. Only the occurrence in the designated target sentence is marked; occurrences in the precontext or ending are left unchanged. The asterisk symbols are included as literal characters in the input string and processed by the tokenizer without introducing additional special tokens.

E.2 Sequence Truncation

Inputs exceeding the maximum sequence length of 512 tokens are truncated from the end of the sequence, prioritizing retention of the candidate sense definition, dictionary example, and target sentence. Precontext and ending text are truncated last if necessary so that the model always receives the full sense definition and target sentence.

E.3 Avoiding Loss Deactivation at Initialization

The initial total weight of the hybrid objective is set to 1.1 so that both CORAL and regression components are active from the start of training. This avoids early-stage dominance or suppression of either objective and helps the model receive a balanced training signal while the dynamic schedule begins to take effect.

F Library Requirements

The following Python package versions (from `requirements.txt`) were used for training and evaluation.

`torch`	`2.9.0+cu128`
`transformers`	`4.57.1`
`peft`	`0.18.0`
`scikit-learn`	`1.7.2`
`numpy`	`2.2.6`
`pandas`	`2.3.3`
`scipy`	`1.16.3`
`matplotlib`	`3.10.7`
`seaborn`	`0.13.2`
`tqdm`	`4.67.1`
`jupyter`	`5.9.1`
`sentencepiece`	`0.2.1`