ClaHF: A Human Feedback-inspired Reinforcement Learning Framework for Improving Classification Tasks

Anonymous ACL submission

Abstract

Text classification models are typically trained via supervised fine-tuning (SFT). However, SFT essentially performs behavior cloning from instance-wise labels and thus fails to adequately capture relative preference relations among samples, which limits the model's ability to shape decision boundaries and calibrate predictive confidence. In this paper, we propose ClaHF, a human feedback-inspired reinforcement learning (RL) framework for text classification that integrates preference modeling and RL optimization into the classification pipeline without requiring additional human annotations. Unlike prior work that relies solely on instance-wise supervision, ClaHF constructs multiple candidate predictions together with their relative ranking relations, and jointly models the Top-1 preference and the ordering among non-optimal candidates within a reward model (RM). This design converts conventional label supervision into preference signals that are directly applicable to policy optimization. We conduct systematic evaluations on eight classification tasks spanning three categories of scenarios. Results demonstrate that ClaHF consistently improves both classification performance and confidence calibration across diverse language models (LMs). The data and code are available at https://anonymous.4open.science/r/ClaHF.

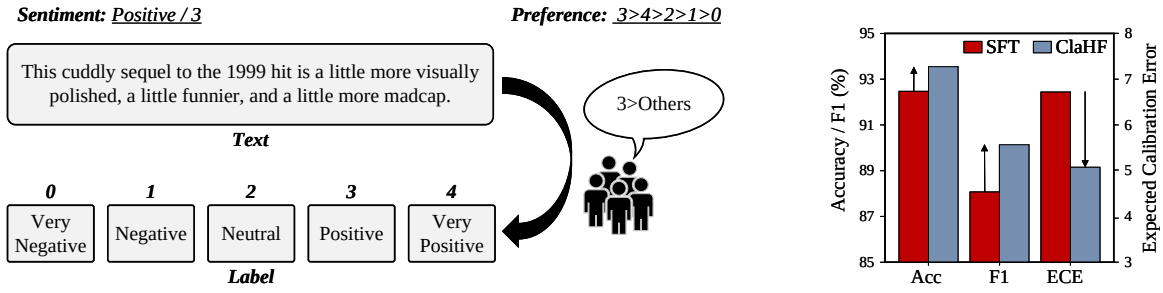
1 Introduction

Text classification is one of the most fundamental and important tasks in natural language processing (NLP), forming the basis of a wide range of downstream applications such as sentiment analysis, news categorization and emotion recognition (Li et al., 2022). In recent years, the rapid development of pre-trained LMs has led to remarkable improvements in these domains, substantially boosting downstream performance (Sun et al., 2023). Nevertheless, most existing text classification models are still trained via SFT, where model predictions are aligned with gold labels on an instance-by-instance basis (Fonseca et al., 2025; Wei and Zhu, 2025). From a learning paradigm perspective, SFT essentially performs behavior cloning: the model is trained to imitate annotators' decisions on the training data in order to learn classification boundaries (Chu et al., 2025).

Although SFT effectively activates the semantic representation capacity of pre-trained models, it lacks further exploration of the decision space (Fu et al., 2023). When data quality is limited, when class boundaries differ only in subtle semantics, or when annotations themselves contain bias, SFT tends to amplify these issues. As a result, the model may overfit the label distribution rather than the underlying semantic patterns, leading to misclassification and unreliable predictions (Ye et al., 2025). This phenomenon shares a similar root cause with the amplification of hallucinations in generative tasks, but in the classification setting it manifests as unstable decision boundaries and degraded generalization (Almeida et al., 2021).

As a complementary paradigm, Reinforcement learning from human feedback (RLHF) (Stiennon et al., 2020; Ouyang et al., 2022) has demonstrated great potential in aligning LMs with human preferences, particularly in generative tasks such as dialogue and instruction following (Dong et al., 2024; Ouyang et al., 2025; Bai et al., 2022). Instead of predicting a single target, RLHF learns from preference signals and optimizes outputs through relative comparisons, thereby better reflecting subjective human judgments (Jiang et al., 2025). This naturally raises the question: can the principles of RLHF be transferred to classification tasks to overcome the inherent limitations of SFT?

Unfortunately, directly extending RLHF to classification faces fundamental challenges. First, unlike the rich sequential outputs of generative models, the output space of classification is discrete



(a) Implicit preference signals in human decisions.

(b) Improved text classification.

Figure 1: (a) An example illustrating the implicit preference signal in the construction of the SST-5 sentiment classification dataset. (b) The performance improvements of ClaHF on text classification tasks (*e.g.*, higher accuracy and F1, lower error). More results are reported in Section 4.2.

and lacks expressive structure (Casper et al., 2023). Second, conventional RLHF relies heavily on manually annotated preference datasets, whose construction is costly and difficult to reuse across tasks and label spaces (Casper et al., 2023; Lindström et al., 2024). These issues make existing RLHF techniques hard to apply to classification, and relevant studies remain scarce.

To address these challenges, we propose ClaHF, a novel human feedback-inspired RL framework for text classification that introduces RLHF into classification training without requiring any additional human annotation. ClaHF is motivated by a key insight: although standard classification datasets do not contain explicit preference annotations, the ground-truth label of each instance implicitly encodes the preference signal underlying human decisions (Figure 1(a)). Based on this insight, ClaHF automatically constructs multiple candidate predictions and their ranking relations from raw classification data, yielding preference data at no extra cost. These preference signals are then used to train a RM that evaluates the relative quality of different predictions. Finally, the classification model is optimized with a preference-aligned proximal policy optimization (PPO) (Schulman et al., 2017) algorithm. Notably, ClaHF is applicable to both binary and multi-class classification.

Unlike conventional instance-wise supervision, ClaHF leverages a multi-stage collaborative process that enables the model not only to "predict correctly" but also to "align with preference signals" embedded in the data, thereby overcoming the limitations of SFT in classification. We conduct systematic experiments on eight benchmark datasets covering binary, multi-class, and code classification tasks. The results show that ClaHF significantly improves both performance and decision reliability over strong SFT baselines (Figure 1(b)). These findings demonstrate that ClaHF bridges the gap between RLHF and classification, and introduce a new perspective: classification models can benefit from preference alignment in addition to label supervision.

2 Related Work

Supervised Learning for Classification. The dominant paradigm is to perform SFT on labeled data using pre-trained models such as RoBERTa (Liu et al., 2019) and Qwen3 (Yang et al., 2025), which yields strong discriminative capability. Despite its empirical success, this training paradigm inherits the limitations of instance-wise supervision, which often leads to overconfidence, unreliable decisions, and misclassification of ambiguous samples near class boundaries. To alleviate these issues, a number of studies have proposed improved objectives. Lin et al. (2017) introduced focal loss to emphasize hard and minority samples, while Müller et al. (2019) proposed label smoothing to improve generalization by injecting soft targets and noise regularization. However, these methods still treat labels as independent supervision signals and fail to exploit the latent relative preferences and ranking relations among samples. Consequently, the training process remains essentially behavior cloning, which is insufficient to provide deeper guidance at the decision level.

Applications of RLHF. RLHF was originally developed to optimize agent behavior in simulated environments (Christiano et al., 2017), and was later applied to domains such as autonomous driving

(Wu et al., 2022) and robot learning (Wang et al., 2022). Prior work (Xu et al., 2023; Zhang et al., 2024; Fang et al., 2025) has also demonstrated the effectiveness of human feedback in image generation. More recently, the successful application of RLHF to large language models (LLMs) has substantially advanced NLP (Stiennon et al., 2020; Ouyang et al., 2022; Ziegler et al., 2019). By modeling human preferences and performing policy updates with PPO, RLHF integrates reward-driven mechanisms into language modeling, fundamentally reshaping dialogue reasoning, safety alignment, and long-form generation (Ji et al., 2025; Liu et al., 2023). However, within downstream NLP tasks, RLHF has so far been largely confined to generative scenarios, such as high-quality code test case generation (Steenhoek et al., 2025).

Preference Learning. Preference learning is a core component of RLHF and models relations among samples via pairwise or listwise comparisons. It has been widely adopted in ranking and recommendation systems (Fürnkranz and Hüllermeier, 2010). In NLP, preference learning is mainly used for generative tasks, although recent studies (Atapour-Abarghouei et al., 2021; Kim et al., 2023) have attempted to introduce preference modeling into classification to overcome the limitations of instance-wise supervision. For example, Kim et al. (2023) proposed the P2C method, which incorporates preference relations during training to enhance the model's discrimination near class boundaries. Nevertheless, existing approaches typically treat preference ranking as an auxiliary loss jointly optimized with SFT; they neither construct an independent RM nor perform policy optimization with RL, and thus remain within the supervised learning paradigm.

3 Method

In this section, we present ClaHF, a novel human feedback-inspired framework for classification. Unlike conventional training paradigms that rely solely on instance-wise supervision, ClaHF borrows the core ideas of RLHF and introduces preference signals into text classification through reward modeling and RL, enabling finer calibration of the predictive distribution and improving both classification performance and decision reliability.

3.1 Problem Definition

We focus on text classification tasks, including both binary and multi-class settings. Given a dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$, where x_i denotes an input sequence (typically a sentence or a paragraph) and y_i is the corresponding class label, conventional approaches learn a classifier f_θ by minimizing instance-wise losses to maximize the agreement between predictions and ground-truth labels. However, such instance-wise supervision provides only a single discriminative signal and cannot characterize the relative quality among different predictions, which limits further optimization.

To this end, we propose ClaHF, which augments standard label supervision with automatically constructed preference signals. Specifically, ClaHF introduces ranking constraints among multiple candidate predictions for the same instance to capture fine-grained differences in model outputs. The goal of ClaHF is therefore to learn a classifier f_θ that not only predicts the correct label but also optimizes its predictive distribution under preference constraints, thereby aligning the model with implicit "human feedback" embedded in the data. The overall framework of ClaHF is illustrated in Figure 2.

3.2 Supervised Fine-tuning

The SFT stage provides a stable and performant initial policy model f_θ for subsequent RL. In ClaHF, the model f_θ consists of a pre-trained LM g_ϕ and a randomly initialized classification head W_ψ. Given an input text x_i, we encode it into token sequences and feed them into f_θ to obtain hidden representations, followed by a softmax layer to produce the class probability distribution:

$$p_\theta(y_i \mid x_i) = \mathrm{Softmax}(f_\theta(x_i)). \tag{1}$$

We optimize the model using the standard cross-entropy loss:

$$\mathcal{L}_{\text{sft}} = -\mathbb{E}_{(x_i,y_i)\sim\mathcal{D}} \log p_\theta(y_i \mid x_i). \tag{2}$$

This stage ensures that the policy model has strong baseline classification capability. However, the supervision signal is purely instance-wise and only enforces consistency with the ground-truth label, without reflecting the relative quality among alternative predictions.

3.3 Preference Data Construction

Training the RM requires data annotated with preference signals. In conventional RLHF, such data

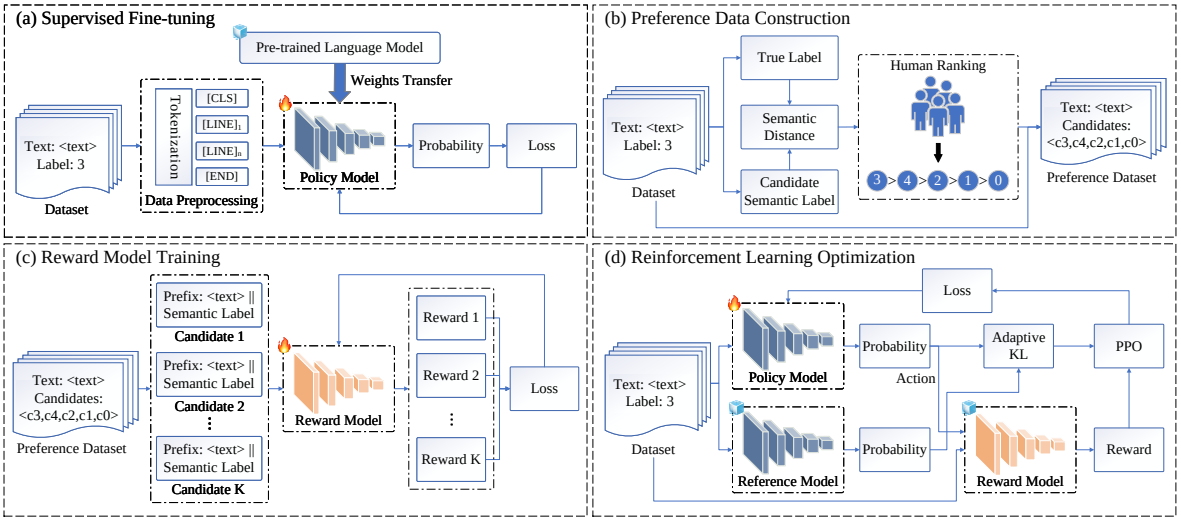


Figure 2: Overall framework of ClaHF. (a) SFT to provide high-quality initialization. (b) Automatic construction of preference data from the original classification dataset. (c) Training the RM with preference data. (d) RL optimization of the policy model using the trained RM.

Dataset	Label	Candidate Semantic Label
SST-5	0	Prediction: the sentence is very negative
	1	Prediction: the sentence is negative
	2	Prediction: the sentence is neutral
	3	Prediction: the sentence is positive
	4	Prediction: the sentence is very positive

Table 1: Ground-truth labels and corresponding candidate semantic labels for SST-5.

are manually labeled, which is expensive and task-specific. In contrast, classification datasets naturally contain ground-truth labels that implicitly encode human preferences. Based on this observation, ClaHF introduces a fully automated method for constructing preference data without any additional annotation.

Specifically, given a standard K-class dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$, we first define a set of semantic label templates:

$$\mathcal{Y}_{\text{text}} = \{0 : c_0, 1 : c_1, \ldots, K-1 : c_{K-1}\}, \quad (3)$$

where each c_j is a natural language description of class j, referred to as a candidate semantic label. Examples for SST-5 are shown in Table 1, and those for other datasets are provided in Appendix B.

For each instance (x_i, y_i), we rank all candidate semantic labels c_j in ascending order of their semantic distance to the true label y_i, yielding a candidate list:

$$C_i = \{c_{i1}, c_{i2}, \ldots, c_{iK}\}, \quad (4)$$

where c_{i1} always corresponds to the ground-truth class, and the remaining candidates are ordered by increasing semantic distance. For example, in SST-5, if the true label is 3, the resulting list is $\{c_3, c_4, c_2, c_1, c_0\}$.

Aggregating all instances produces the preference dataset $\mathcal{D}_{\text{pre}} = \{(x_i, C_i)\}_{i=1}^{N}$. Unlike the original dataset that only contains binary correctness information, $\mathcal{D}_{\text{pre}}$ encodes relative plausibility among classes and naturally introduces a "Top-1 is better than others" preference structure.

3.4 Reward Model Training

The third stage of ClaHF trains a RM to evaluate the relative quality of text–candidate pairs and provide reward signals for subsequent policy optimization. The RM f_{rm} consists of the same pre-trained encoder g_ϕ as in SFT and a linear scoring head that outputs a scalar reward.

Given $\mathcal{D}_{\text{pre}} = \{(x_i, C_i)\}_{i=1}^{N}$, for each (x_i, C_i) we concatenate x_i with each c_{ij}, tokenize the sequence, and feed it into f_{rm}, which is expected to satisfy:

$$f_{\text{rm}}(x_i, c_{i1}) > \cdots > f_{\text{rm}}(x_i, c_{iK}), \quad (5)$$

where c_{i1} corresponds to the ground-truth class.

Since classification lacks the rich candidate structure of generative tasks, we design a composite ranking loss with two components: $\mathcal{L}_{\text{top1}}$ and $\mathcal{L}_{\text{pairwise}}$. The Top-1 loss enforces the ground-truth

candidate to score higher than all others:

$$\mathcal{L}_{\text{top1}} = -\frac{1}{K-1}\sum_{j=2}^{K}\log\sigma(f_{\text{rm}}(x_i, c_{i1}) - f_{\text{rm}}(x_i, c_{ij})), \tag{6}$$

where σ is the sigmoid function.

The pairwise loss constrains the relative order among non-true candidates:

$$\mathcal{L}_{\text{pairwise}} = -\frac{2}{(K-1)(K-2)}\sum_{2\le j<k\le K} \log\sigma(f_{\text{rm}}(x_i, c_{ij}) - f_{\text{rm}}(x_i, c_{ik})). \tag{7}$$

The overall objective is:

$$\mathcal{L}_{\text{rm}} = \alpha\mathcal{L}_{\text{top1}} + (1-\alpha)\mathcal{L}_{\text{pairwise}}, \tag{8}$$

where $\alpha \in (0, 1]$ balances Top-1 preference and pairwise ranking consistency.

For binary classification, the loss simplifies to:

$$\mathcal{L}_{\text{rm}} = -\log\sigma(f_{\text{rm}}(x_i, c_{i1}) - f_{\text{rm}}(x_i, c_{i2})). \tag{9}$$

This training procedure enables the RM to capture finer-grained preference signals than conventional supervision.

3.5 Reinforcement Learning Optimization

In the final stage, we further optimize the policy model f_θ using PPO (Schulman et al., 2017) to improve decision quality and predictive reliability. A frozen reference model f_{ref}, with the same architecture as f_θ, is used to measure deviation from the supervised baseline.

Given an input x_i, the policy model f_θ outputs a class distribution $p_\theta(y_i \mid x_i)$, while the reference model f_{ref} outputs $p_{\text{ref}}(y_i \mid x_i)$. An action a_i is defined as a class label sampled from $p_\theta(y_i \mid x_i)$, and its raw reward $f_{\text{rm}}(x_i, a_i)$ is computed by the RM. To prevent the policy from drifting excessively from the supervised distribution, we introduce an adaptive KL-penalty (Ziegler et al., 2019) term into the reward and obtain the modified reward:

$$\hat{f_{\text{rm}}}(x_i, a_i) = f_{\text{rm}}(x_i, a_i) - \beta_i D_{\text{KL}}(p_\theta(\cdot \mid x_i) \,||\, p_{\text{ref}}(\cdot \mid x_i)), \tag{10}$$

where D_{KL} denotes the KL divergence between the two distributions. The KL coefficient β is dynamically adjusted by an adaptive controller:

$$\beta_{i+1} = \text{clip}(\beta_i \cdot (1 + \eta \cdot \frac{D_{\text{KL}} - D_{\text{target}}}{D_{\text{target}}}), \beta_{\min}, \beta_{\max}), \tag{11}$$

which keeps the KL divergence close to a target value D_{target}; η is the update rate.

During PPO optimization, ClaHF maximizes the following clipped surrogate objective for each sample by comparing the probability ratio between the current and old policies:

$$\mathcal{L}_{\text{policy}} = \mathbb{E}_{x_i\sim\mathcal{D}}[-\min(r_i A_i, \text{clip}(r_i, 1-\epsilon, 1+\epsilon)A_i)], \tag{12}$$

where $r_i = \frac{p_\theta(a_i \mid x_i)}{p_{\text{ref}}(a_i \mid x_i)}$ is the probability ratio of the new and old policies, and A_i is the advantage function defined as:

$$A_i = \hat{f_{\text{rm}}}(x_i, a_i) - V(x_i), \tag{13}$$

with V denoting the output of the value function network. The value function is introduced to reduce training variance and stabilize policy learning. It is optimized using mean squared error:

$$\mathcal{L}_{\text{value}} = \mathbb{E}_{x_i\sim\mathcal{D}}[(V(x_i) - \hat{f_{\text{rm}}}(x_i, a_i))^2]. \tag{14}$$

The final optimization objective in the RL stage is:

$$\mathcal{L}_{\text{rl}} = \mathcal{L}_{\text{policy}} + \gamma\mathcal{L}_{\text{value}}, \tag{15}$$

where γ controls the relative weight of the value loss.

Through this design, ClaHF tightly couples preference-driven reward modeling with classification decisions, continuously refining decision boundaries while maintaining stability with respect to the policy model.

4 Experiments

4.1 Setups

Datasets. We select eight datasets to evaluate the effectiveness of ClaHF from three perspectives: binary classification, multi-class classification, and software engineering code tasks. Specifically, the binary classification tasks include: (1) SST-2 (Socher et al., 2013), (2) CoLA (Warstadt et al., 2019), and (3) MRPC (Dolan and Brockett, 2005). The multi-class classification tasks include: (4) AG News (Zhang et al., 2015), (5) SST-5 (Socher et al., 2013), and (6) Emotion (Saravia et al., 2018). The code tasks include: (7) Devign (Zhou et al., 2019), and (8) BigCloneBench (Wang et al., 2020). The data splits and additional details are provided in Appendix A.

Baselines. As existing RLHF frameworks are primarily designed for generative tasks and no mature solution is available for classification, we only

Backbone	Method	AG News			SST-5			Emotion		
		Acc (↑)	F1 (↑)	ECE (↓)	Acc (↑)	F1 (↑)	ECE (↓)	Acc (↑)	F1 (↑)	ECE (↓)
BERT	SFT	$91.91_{\pm 0.0}$	$91.90_{\pm 0.0}$	$8.35_{\pm 0.3}$	$52.49_{\pm 0.1}$	$51.28_{\pm 0.1}$	$13.44_{\pm 0.7}$	$92.25_{\pm 0.2}$	$87.71_{\pm 0.5}$	$6.51_{\pm 0.4}$
	ClaHF	$92.18_{\pm 0.1}$	$92.17_{\pm 0.1}$	$6.96_{\pm 0.5}$	$54.30_{\pm 0.1}$	$51.50_{\pm 0.8}$	$11.36_{\pm 0.3}$	$92.85_{\pm 0.1}$	$88.05_{\pm 0.4}$	$5.74_{\pm 0.5}$
RoBERTa	SFT	$94.44_{\pm 0.0}$	$94.44_{\pm 0.0}$	$7.47_{\pm 0.3}$	$54.35_{\pm 0.7}$	$52.71_{\pm 0.9}$	$10.39_{\pm 0.5}$	$92.55_{\pm 0.1}$	$88.40_{\pm 0.4}$	$7.23_{\pm 0.1}$
	ClaHF	$94.61_{\pm 0.0}$	$94.60_{\pm 0.0}$	$5.52_{\pm 0.2}$	$55.87_{\pm 0.4}$	$54.17_{\pm 0.6}$	$8.43_{\pm 0.3}$	$92.93_{\pm 0.1}$	$88.93_{\pm 0.1}$	$6.80_{\pm 0.1}$
T5	SFT	$94.51_{\pm 0.2}$	$94.51_{\pm 0.1}$	$5.27_{\pm 0.2}$	$56.19_{\pm 0.4}$	$53.11_{\pm 0.3}$	$12.09_{\pm 0.5}$	$91.80_{\pm 0.1}$	$87.11_{\pm 0.4}$	$6.19_{\pm 0.3}$
	ClaHF	$94.79_{\pm 0.1}$	$94.79_{\pm 0.1}$	$4.52_{\pm 0.1}$	$57.47_{\pm 0.8}$	$54.72_{\pm 0.9}$	$11.02_{\pm 0.4}$	$92.20_{\pm 0.1}$	$88.11_{\pm 0.1}$	$4.74_{\pm 0.3}$
OPT	SFT	$93.77_{\pm 0.1}$	$93.77_{\pm 0.1}$	$9.60_{\pm 0.5}$	$52.85_{\pm 0.4}$	$49.72_{\pm 0.5}$	$10.42_{\pm 0.8}$	$92.28_{\pm 0.4}$	$87.74_{\pm 0.7}$	$6.62_{\pm 0.1}$
	ClaHF	$94.00_{\pm 0.1}$	$94.01_{\pm 0.1}$	$5.76_{\pm 0.2}$	$54.34_{\pm 0.3}$	$52.43_{\pm 0.2}$	$7.71_{\pm 0.8}$	$92.92_{\pm 0.1}$	$88.46_{\pm 0.2}$	$6.27_{\pm 0.2}$
Qwen3	SFT	$93.85_{\pm 0.1}$	$93.83_{\pm 0.1}$	$7.07_{\pm 0.9}$	$51.06_{\pm 0.6}$	$48.92_{\pm 0.6}$	$8.84_{\pm 0.3}$	$92.57_{\pm 0.1}$	$88.11_{\pm 0.4}$	$6.75_{\pm 0.2}$
	ClaHF	$94.17_{\pm 0.1}$	$94.16_{\pm 0.1}$	$5.22_{\pm 0.5}$	$52.67_{\pm 0.1}$	$49.55_{\pm 1.0}$	$7.61_{\pm 0.4}$	$93.62_{\pm 0.1}$	$90.18_{\pm 0.1}$	$5.06_{\pm 0.2}$
	Average	**+0.25**	**+0.25**	**-1.96**	**+1.54**	**+1.32**	**-1.78**	**+0.61**	**+0.93**	**-0.94**

Table 2: Test performance of SFT and ClaHF on three multi-class datasets. All values are reported as the mean and standard error over five random seeds.

compare ClaHF with standard SFT, where backbone models are trained on labeled data using cross-entropy loss. To verify the generality of ClaHF, we adopt nine widely used pre-trained LMs as backbones, including BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019), OPT (Zhang et al., 2022), T5 (Raffel et al., 2020), and Qwen3 (Yang et al., 2025) for natural language tasks, as well as CodeBERT (Feng, 2020), CodeT5 (Wang et al., 2021), CodeT5+ (Wang et al., 2023), and CodeGen (Nijkamp et al., 2022) for code-related tasks.

Implementation details. All experiments are conducted on 8 NVIDIA RTX 4090 GPUs. Pre-trained weights are loaded via Hugging Face (Wolf et al., 2020). We use the AdamW optimizer (Loshchilov and Hutter, 2017) for all experiments and train each model for 10 epochs. During the SFT stage, the batch size is set to 16 and the learning rate to 2e-5. For RM training, the learning rate is 1e-5 and the preference loss weight α is set to 0.8. During the RL stage, the learning rate is 1e-6 and the value loss coefficient γ is set to 0.25. Additional hyper-parameters and training details are reported in Appendix B.

4.2 Main Results

In this section, we systematically evaluate ClaHF on multi-class, binary, and code classification tasks and compare it with the corresponding SFT baselines across a wide range of backbone models, in order to assess its cross-model and cross-task generalization ability. To measure performance under class-imbalance scenarios, we report both Accuracy (*Acc*) and macro-F1 (*F1*). For CoLA, we report the commonly used Matthews Correlation Coefficient (*MCC*) (Chicco et al., 2021). In addition, we introduce Expected Calibration Error (*ECE*) (Guo et al., 2017) to evaluate predictive reliability.

4.2.1 Results on Multi-class Tasks

Table 2 summarizes the results of ClaHF and SFT on three multi-class datasets. ClaHF consistently outperforms SFT across all backbones and datasets, showing simultaneous improvements in Acc, F1, and ECE. This indicates that incorporating preference signals effectively complements instance-wise supervision and enables the model to better distinguish decision boundaries between different classes.

On the fine-grained sentiment dataset SST-5, where semantic differences are subtle, ClaHF achieves particularly notable gains, with an average improvement of 1.54 percentage points in Acc accompanied by a significant reduction in ECE. This demonstrates that ClaHF yields stronger discriminative ability and more reliable probability calibration in complex multi-class scenarios.

Similar trends are observed on the Emotion dataset. For example, with Qwen3, ClaHF improves Acc from 92.57% to 93.62% while reducing ECE from 6.75 to 5.06, indicating that ClaHF not only boosts accuracy but also fundamentally promotes more reliable and well-calibrated predictive distributions.

4.2.2 Results on Binary Tasks

Compared with multi-class classification, binary tasks are more sensitive to decision boundaries and impose stricter requirements on model confidence and distributional consistency. Therefore, we report these results separately.

Backbone	Method	SST-2			MRPC			CoLA	
		Acc (↑)	F1 (↑)	ECE (↓)	Acc (↑)	F1 (↑)	ECE (↓)	MCC (↑)	ECE (↓)
BERT	SFT	$89.46_{\pm 0.24}$	$89.43_{\pm 0.24}$	$9.99_{\pm 0.14}$	$82.18_{\pm 0.27}$	$79.44_{\pm 0.96}$	$9.84_{\pm 0.46}$	$55.07_{\pm 0.34}$	$14.30_{\pm 0.48}$
	ClaHF	$90.10_{\pm 0.09}$	$90.09_{\pm 0.10}$	$9.04_{\pm 0.55}$	$83.28_{\pm 0.29}$	$80.53_{\pm 0.29}$	$8.64_{\pm 0.21}$	$56.10_{\pm 0.24}$	$12.16_{\pm 0.57}$
RoBERTa	SFT	$92.57_{\pm 0.12}$	$92.57_{\pm 0.12}$	$8.24_{\pm 0.47}$	$87.25_{\pm 0.18}$	$85.57_{\pm 0.05}$	$6.25_{\pm 0.29}$	$59.73_{\pm 0.62}$	$11.77_{\pm 0.29}$
	ClaHF	$93.19_{\pm 0.23}$	$93.19_{\pm 0.23}$	$6.40_{\pm 0.13}$	$87.86_{\pm 0.15}$	$86.28_{\pm 0.23}$	$5.12_{\pm 0.20}$	$61.21_{\pm 0.82}$	$8.43_{\pm 0.37}$
T5	SFT	$93.85_{\pm 0.23}$	$93.85_{\pm 0.23}$	$7.10_{\pm 0.60}$	$84.58_{\pm 0.12}$	$81.34_{\pm 0.27}$	$8.15_{\pm 0.30}$	$58.10_{\pm 0.48}$	$13.58_{\pm 0.51}$
	ClaHF	$94.63_{\pm 0.04}$	$94.64_{\pm 0.04}$	$5.03_{\pm 0.13}$	$85.78_{\pm 0.28}$	$83.68_{\pm 0.28}$	$6.83_{\pm 0.27}$	$59.23_{\pm 0.46}$	$11.75_{\pm 0.33}$
OPT	SFT	$90.52_{\pm 0.46}$	$90.51_{\pm 0.45}$	$8.76_{\pm 0.52}$	$83.61_{\pm 0.25}$	$80.69_{\pm 0.37}$	$7.55_{\pm 0.19}$	$56.42_{\pm 0.26}$	$13.95_{\pm 0.77}$
	ClaHF	$90.99_{\pm 0.23}$	$90.99_{\pm 0.23}$	$6.24_{\pm 0.31}$	$84.69_{\pm 0.22}$	$81.88_{\pm 0.85}$	$6.44_{\pm 0.26}$	$57.64_{\pm 0.35}$	$12.90_{\pm 0.43}$
Qwen3	SFT	$90.57_{\pm 0.16}$	$90.57_{\pm 0.16}$	$8.22_{\pm 0.36}$	$84.41_{\pm 0.15}$	$81.69_{\pm 0.54}$	$6.85_{\pm 0.13}$	$56.94_{\pm 0.17}$	$13.68_{\pm 0.66}$
	ClaHF	$91.47_{\pm 0.29}$	$91.47_{\pm 0.29}$	$7.09_{\pm 0.20}$	$85.02_{\pm 0.18}$	$82.97_{\pm 0.26}$	$5.58_{\pm 0.36}$	$58.30_{\pm 0.27}$	$12.44_{\pm 0.34}$
	Average	**+0.68**$_{\pm 0.07}$	**+0.69**$_{\pm 0.07}$	**-1.70**$_{\pm 0.29}$	**+0.92**$_{\pm 0.13}$	**+1.32**$_{\pm 0.27}$	**-1.21**$_{\pm 0.04}$	**+1.24**$_{\pm 0.08}$	**-1.92**$_{\pm 0.41}$

Table 3: Test performance of SFT and ClaHF on three binary datasets.

Backbone	Method	Devign			BigCloneBench		
		Acc (↑)	F1 (↑)	ECE (↓)	Acc (↑)	F1 (↑)	ECE (↓)
CodeBERT	SFT	$62.15_{\pm 0.19}$	$58.77_{\pm 0.50}$	$13.48_{\pm 0.81}$	$97.12_{\pm 0.14}$	$97.11_{\pm 0.14}$	$4.01_{\pm 0.43}$
	ClaHF	$63.14_{\pm 0.11}$	$59.61_{\pm 0.28}$	$9.64_{\pm 0.47}$	$97.58_{\pm 0.20}$	$97.58_{\pm 0.20}$	$2.49_{\pm 0.34}$
CodeT5	SFT	$62.99_{\pm 0.18}$	$62.35_{\pm 0.20}$	$12.02_{\pm 0.48}$	$96.80_{\pm 0.17}$	$96.79_{\pm 0.17}$	$4.93_{\pm 0.83}$
	ClaHF	$63.49_{\pm 0.36}$	$62.50_{\pm 0.56}$	$9.06_{\pm 0.24}$	$97.29_{\pm 0.16}$	$97.29_{\pm 0.16}$	$2.46_{\pm 0.26}$
CodeT5+	SFT	$63.20_{\pm 0.17}$	$62.87_{\pm 0.25}$	$11.13_{\pm 0.42}$	$97.87_{\pm 0.10}$	$97.87_{\pm 0.10}$	$2.08_{\pm 0.06}$
	ClaHF	$64.04_{\pm 0.15}$	$63.57_{\pm 0.14}$	$8.60_{\pm 0.53}$	$98.14_{\pm 0.07}$	$98.13_{\pm 0.07}$	$1.79_{\pm 0.10}$
CodeGen	SFT	$62.38_{\pm 0.29}$	$59.29_{\pm 0.24}$	$15.11_{\pm 0.44}$	$97.60_{\pm 0.14}$	$97.60_{\pm 0.14}$	$2.43_{\pm 0.27}$
	ClaHF	$63.45_{\pm 0.11}$	$61.95_{\pm 0.87}$	$11.13_{\pm 0.71}$	$97.98_{\pm 0.09}$	$97.98_{\pm 0.09}$	$1.92_{\pm 0.05}$
	Average	**+0.85**$_{\pm 0.12}$	**+1.09**$_{\pm 0.55}$	**-3.33**$_{\pm 0.35}$	**+0.40**$_{\pm 0.05}$	**+0.40**$_{\pm 0.05}$	**-1.20**$_{\pm 0.50}$

Table 4: Test performance of SFT and ClaHF on code datasets.

As shown in Table 3, ClaHF consistently outperforms SFT on SST-2, MRPC, and CoLA across all backbones, with particularly large gains in ECE. On SST-2, ClaHF achieves simultaneous improvements in Acc and F1, demonstrating that preference signals help the model better capture the relative ordering between positive and negative samples.

The advantage of ClaHF is even more pronounced on MRPC, which requires fine-grained judgments of semantic equivalence and is highly sensitive to boundary cases. Compared with SFT, ClaHF substantially improves performance and reduces ECE by about 1.21, indicating enhanced decision stability near hard examples. On CoLA, ClaHF improves MCC by approximately 1.24 while reducing ECE by about 1.92, further confirming its dual benefits in refining decision boundaries and improving confidence calibration.

4.2.3 Results on Code Tasks

Compared with natural language, source code is a structured and formal language with strict syntactic and semantic constraints, which places higher demands on model discrimination and stability.

Table 4 presents the results on Devign and BigCloneBench. Overall, ClaHF achieves consistent improvements across all code backbones. On Devign, ClaHF reduces ECE by an average of 3.33 while also yielding small but consistent gains in Acc and F1, demonstrating its effectiveness on structured code data. On BigCloneBench, ClaHF again shows stable advantages, especially in terms of ECE, indicating that it maintains reliable confidence calibration even in high-accuracy regimes—an essential property for practical software engineering applications.

4.3 Ablation Study

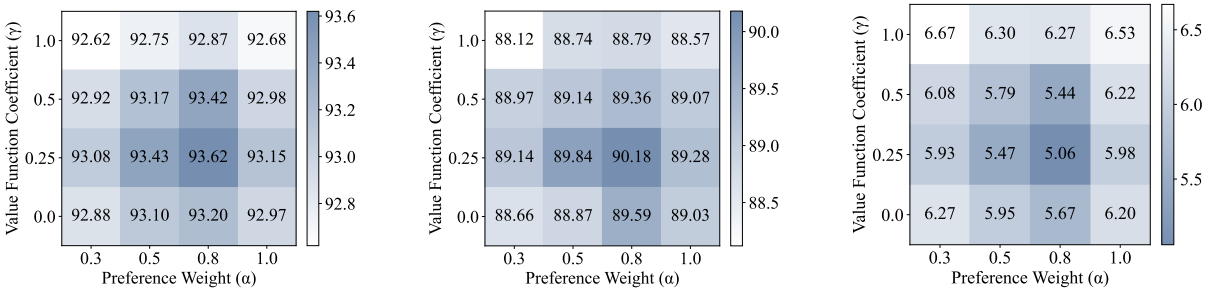
To further investigate the contribution of each stage in ClaHF, we conduct ablation studies from two perspectives. Unless otherwise specified, all experiments in this section are conducted on the Emotion dataset (Saravia et al., 2018) with the Qwen3 (Yang et al., 2025) backbone, and all other settings are kept identical.

4.3.1 Effect of RM Training Objectives

Table 5 reports the impact of different RM training objectives on the final classification performance. In this group of experiments, we only vary the

Method	$\mathcal{L}_{\text{sft}}$	$\mathcal{L}_{\text{top1}}$	$\mathcal{L}_{\text{pairwise}}$	Acc (↑)	F1 (↑)	ECE (↓)
SFT	✓	–	–	$92.57_{\pm 0.14}$	$88.11_{\pm 0.40}$	$6.75_{\pm 0.22}$
ClaHF-Pair	✓	–	✓	$92.78_{\pm 0.14}$	$88.88_{\pm 0.12}$	$6.44_{\pm 0.21}$
ClaHF-Top1	✓	✓	–	$93.15_{\pm 0.20}$	$89.28_{\pm 0.40}$	$5.98_{\pm 0.15}$
ClaHF (Full)	✓	✓	✓	$\mathbf{93.62}_{\pm 0.10}$	$\mathbf{90.18}_{\pm 0.12}$	$\mathbf{5.06}_{\pm 0.21}$

Table 5: Effect of RM training objectives on classification performance.



(a) Heatmap of Acc. (b) Heatmap of F1. (c) Heatmap of ECE.

Figure 3: Performance heatmaps of ClaHF in terms of Acc, F1, and ECE over a continuous $\alpha \times \gamma$ grid.

loss components used in the RM stage. To isolate the effect of preference signals, we construct two variants of ClaHF: (1) ClaHF-Pair: applies equal constraints to all candidates. (2) ClaHF-Top1: only enforces the margin between the Top-1 candidate and the remaining candidates.

As shown in the table, all three ClaHF variants outperform SFT, confirming that preference signals effectively complement single-label supervision. However, ClaHF-Pair yields only limited improvements, indicating that enforcing pairwise ranking consistency alone is insufficient to provide strong discriminative signals and behaves similarly to conventional learning-to-rank methods.

When both Top-1 and pairwise constraints are combined, the model achieves the best performance across Acc, F1, and ECE, which significantly exceeds SFT and the two variants. This result indicates that the two types of preference signals are complementary: the Top-1 constraint emphasizes discriminability of the optimal decision, while the pairwise constraint preserves the global ranking structure. Additional studies are reported in Appendix C.

4.3.2 Stability Analysis

We further analyze the joint effect of the preference weight α and the value coefficient γ over a continuous hyperparameter space to assess the stability of ClaHF and the structure of its optimal region.

Figure 3 presents heatmaps of Acc, F1, and ECE over a continuous $\alpha \times \gamma$ grid. The performance surfaces vary smoothly rather than exhibiting abrupt oscillations, indicating that ClaHF is robust to hyperparameter perturbations and that small changes within reasonable ranges do not lead to dramatic performance drops.

Moreover, the optimal performance does not concentrate at a single point but spans a coherent region. Specifically, when $\alpha \in [0.7, 0.9]$ and $\gamma \in [0.2, 0.3]$, the model achieves near-optimal Acc and F1 while maintaining low ECE. The existence of this structured optimal region demonstrates that the improvements brought by ClaHF are not the result of accidental tuning, but remain stable across a range of reasonable configurations.

5 Conclusion

We propose ClaHF, a human-feedback-inspired RL framework for text classification, which for the first time systematically introduces the complete RLHF pipeline into classification tasks. ClaHF is tailored to the decision structure of classification and incorporates dedicated designs for preference data construction, reward modeling, and policy optimization, enabling RL to effectively optimize discriminative prediction objectives. Compared with SFT, ClaHF achieves stable and consistent improvements in both performance and calibration across a wide range of natural language and code classification tasks, demonstrating that preference-driven RL is also a powerful paradigm for enhancing the effectiveness and reliability of classification models.

Limitations

Despite the promising results of ClaHF on multiple text classification benchmarks, it has two main limitations. (1) ClaHF relies on automatically constructed preference data derived from original labels. Although this avoids additional human annotation costs, the quality of RM training is inevitably constrained by the noise and bias in the original datasets. Integrating ClaHF with real human feedback is a promising direction to further improve performance and robustness. (2) Our experiments focus on single-label text classification. The applicability of ClaHF to more complex discriminative settings, such as multi-label and hierarchical classification, remains unexplored and is left for future work.

Acknowledgments

References

Matthew Almeida, Yong Zhuang, Wei Ding, Scott E Crouter, and Ping Chen. 2021. Mitigating class-boundary label uncertainty to reduce both model bias and variance. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 15(2):1–18.

Amir Atapour-Abarghouei, Stephen Bonner, and Andrew Stephen McGough. 2021. Rank over class: The untapped potential of ranking in natural language processing. In *2021 IEEE International Conference on Big Data (Big Data)*, pages 3950–3959. IEEE.

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, and 1 others. 2022. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*.

Stephen Casper, Xander Davies, Claudia Shi, Thomas Krendl Gilbert, Jérémy Scheurer, Javier Rando, Rachel Freedman, Tomasz Korbak, David Lindner, Pedro Freire, and 1 others. 2023. Open problems and fundamental limitations of reinforcement learning from human feedback. *arXiv preprint arXiv:2307.15217*.

Davide Chicco, Matthijs J Warrens, and Giuseppe Jurman. 2021. The matthews correlation coefficient (mcc) is more informative than cohen's kappa and brier score in binary classification assessment. *Ieee Access*, 9:78368–78381.

Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. 2017. Deep reinforcement learning from human preferences. *Advances in neural information processing systems*, 30.

Tianzhe Chu, Yuexiang Zhai, Jihan Yang, Shengbang Tong, Saining Xie, Dale Schuurmans, Quoc V Le, Sergey Levine, and Yi Ma. 2025. Sft memorizes, rl generalizes: A comparative study of foundation model post-training. *arXiv preprint arXiv:2501.17161*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pages 4171–4186.

William B Dolan and Chris Brockett. 2005. Automatically constructing a corpus of sentential paraphrases. In *Proceedings of the third international workshop on paraphrasing (IWP2005)*.

Hanze Dong, Wei Xiong, Bo Pang, Haoxiang Wang, Han Zhao, Yingbo Zhou, Nan Jiang, Doyen Sahoo, Caiming Xiong, and Tong Zhang. 2024. Rlhf workflow: From reward modeling to online rlhf. *arXiv preprint arXiv:2405.07863*.

Tiyu Fang, Mingxin Zhang, Ran Song, Xiaolei Li, Zhiyuan Wei, and Wei Zhang. 2025. Human-guided data augmentation via diffusion model for surface defect recognition under limited data. *IEEE Transactions on Instrumentation and Measurement*.

Z Feng. 2020. Codebert: A pre-trained model for program-ming and natural languages. *arXiv preprint arXiv:2002.08155*.

Guilherme Fonseca, Washington Cunha, Gabriel Prenassi, Marcos André Gonçalves, and Leonardo Chaves Dutra Da Rocha. 2025. Instance-selection-inspired undersampling strategies for bias reduction in small and large language models for binary text classification. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9323–9340.

Zihao Fu, Anthony Man-Cho So, and Nigel Collier. 2023. A stability analysis of fine-tuning a pre-trained model. *arXiv preprint arXiv:2301.09820*.

Johannes Fürnkranz and Eyke Hüllermeier. 2010. Preference learning and ranking by pairwise comparison. In *Preference learning*, pages 65–82. Springer.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. 2017. On calibration of modern neural networks. In *International conference on machine learning*, pages 1321–1330. PMLR.

Miaomiao Ji, Yanqiu Wu, Zhibin Wu, Shoujin Wang, Jian Yang, Mark Dras, and Usman Naseem. 2025. A survey on progress in llm alignment from the perspective of reward design. *arXiv preprint arXiv:2505.02666*.

Ruili Jiang, Kehai Chen, Xuefeng Bai, Zhixuan He, Juntao Li, Muyun Yang, Tiejun Zhao, Liqiang Nie, and Min Zhang. 2025. A survey on human preference learning for aligning large language models. *ACM Computing Surveys*, 58(6).

Jaehyung Kim, Jinwoo Shin, and Dongyeop Kang. 2023. Prefer to classify: Improving text classifiers via auxiliary preference learning. In *International Conference on Machine Learning*, pages 16807–16828. PMLR.

Qian Li, Hao Peng, Jianxin Li, Congying Xia, Renyu Yang, Lichao Sun, Philip S Yu, and Lifang He. 2022. A survey on text classification: From traditional to deep learning. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 13(2):1–41.

Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. 2017. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988.

Adam Dahlgren Lindström, Leila Methnani, Lea Krause, Petter Ericson, Íñigo Martínez de Rituerto de Troya, Dimitri Coelho Mollo, and Roel Dobbe. 2024. Ai alignment through reinforcement learning from human feedback? contradictions and limitations. *arXiv preprint arXiv:2406.18346*.

Yiheng Liu, Tianle Han, Siyuan Ma, Jiayue Zhang, Yuanyuan Yang, Jiaming Tian, Hao He, Antong Li, Mengshen He, Zhengliang Liu, and 1 others. 2023. Summary of chatgpt-related research and perspective towards the future of large language models. *Meta-radiology*, 1(2):100017.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Shuai Lu, Daya Guo, Shuo Ren, Junjie Huang, Alexey Svyatkovskiy, Ambrosio Blanco, Colin Clement, Dawn Drain, Daxin Jiang, Duyu Tang, and 1 others. 2021. Codexglue: A machine learning benchmark dataset for code understanding and generation. *arXiv preprint arXiv:2102.04664*.

Rafael Müller, Simon Kornblith, and Geoffrey E Hinton. 2019. When does label smoothing help? *Advances in neural information processing systems*, 32.

Erik Nijkamp, Bo Pang, Hiroaki Hayashi, Lifu Tu, Huan Wang, Yingbo Zhou, Silvio Savarese, and Caiming Xiong. 2022. Codegen: An open large language model for code with multi-turn program synthesis. *arXiv preprint arXiv:2203.13474*.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, and 1 others. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.

Sheng Ouyang, Yulan Hu, Ge Chen, Qingyang Li, Fuzheng Zhang, and Yong Liu. 2025. Towards reward fairness in rlhf: From a resource allocation perspective. *arXiv preprint arXiv:2505.23349*.

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

Elvis Saravia, Hsien-Chi Toby Liu, Yen-Hao Huang, Junlin Wu, and Yi-Shin Chen. 2018. Carer: Contextualized affect representations for emotion recognition. In *Proceedings of the 2018 conference on empirical methods in natural language processing*, pages 3687–3697.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1631–1642.

Benjamin Steenhoek, Michele Tufano, Neel Sundaresan, and Alexey Svyatkovskiy. 2025. Reinforcement learning from automatic feedback for high-quality unit test generation. In *2025 IEEE/ACM International Workshop on Deep Learning for Testing and Testing for Deep Learning (DeepTest)*, pages 37–44. IEEE.

Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F Christiano. 2020. Learning to summarize with human feedback. *Advances in neural information processing systems*, 33:3008–3021.

Xiaofei Sun, Xiaoya Li, Jiwei Li, Fei Wu, Shangwei Guo, Tianwei Zhang, and Guoyin Wang. 2023. Text classification via large language models. *arXiv preprint arXiv:2305.08377*.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. Glue: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP workshop BlackboxNLP: Analyzing and interpreting neural networks for NLP*, pages 353–355.

Wenhan Wang, Ge Li, Bo Ma, Xin Xia, and Zhi Jin. 2020. Detecting code clones with graph neural network and flow-augmented abstract syntax tree. In *2020 IEEE 27th International Conference on Software Analysis, Evolution and Reengineering (SANER)*, pages 261–271. IEEE.

Xiaofei Wang, Kimin Lee, Kourosh Hakhamaneshi, Pieter Abbeel, and Michael Laskin. 2022. Skill preferences: Learning to extract and execute robotic skills from human feedback. In *Conference on robot learning*, pages 1259–1268. PMLR.

Yue Wang, Hung Le, Akhilesh Gotmare, Nghi Bui, Junnan Li, and Steven Hoi. 2023. Codet5+: Open code large language models for code understanding and generation. In *Proceedings of the 2023 conference on empirical methods in natural language processing*, pages 1069–1088.

Yue Wang, Weishi Wang, Shafiq Joty, and Steven CH Hoi. 2021. Codet5: Identifier-aware unified pre-trained encoder-decoder models for code understanding and generation. *arXiv preprint arXiv:2109.00859*.

Alex Warstadt, Amanpreet Singh, and Samuel R Bowman. 2019. Neural network acceptability judgments. *Transactions of the Association for Computational Linguistics*, 7:625–641.

Bowen Wei and Ziwei Zhu. 2025. Protolens: Advancing prototype learning for fine-grained interpretability in text classification. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4503–4523.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi 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.

Jingda Wu, Zhiyu Huang, Wenhui Huang, and Chen Lv. 2022. Prioritized experience-based reinforcement learning with human guidance for autonomous driving. *IEEE Transactions on Neural Networks and Learning Systems*, 35(1):855–869.

Jiazheng Xu, Xiao Liu, Yuchen Wu, Yuxuan Tong, Qinkai Li, Ming Ding, Jie Tang, and Yuxiao Dong. 2023. Imagereward: Learning and evaluating human preferences for text-to-image generation. *Advances in Neural Information Processing Systems*, 36:15903–15935.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Junjie Ye, Yuming Yang, Yang Nan, Shuo Li, Qi Zhang, Tao Gui, Xuan-Jing Huang, Peng Wang, Zhongchao Shi, and Jianping Fan. 2025. Analyzing the effects of supervised fine-tuning on model knowledge from token and parameter levels. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 471–513.

Shu Zhang, Xinyi Yang, Yihao Feng, Can Qin, Chia-Chih Chen, Ning Yu, Zeyuan Chen, Huan Wang, Silvio Savarese, Stefano Ermon, and 1 others. 2024. Hive: Harnessing human feedback for instructional visual editing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9026–9036.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, and 1 others. 2022. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. *Advances in neural information processing systems*, 28.

Yaqin Zhou, Shangqing Liu, Jingkai Siow, Xiaoning Du, and Yang Liu. 2019. Devign: Effective vulnerability identification by learning comprehensive program semantics via graph neural networks. *Advances in neural information processing systems*, 32.

Daniel M Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. 2019. Fine-tuning language models from human preferences. *arXiv preprint arXiv:1909.08593*.

A Datasets

As described in Section 4.1, we evaluate ClaHF on eight representative text classification benchmarks. For all datasets, we set the maximum input length to 400 tokens for tokenization. The detailed descriptions are as follows.

SST-2 (Socher et al., 2013) is a single-sentence binary classification dataset consisting of sentences from movie reviews annotated with sentiment labels. The labels are 0 and 1, where 0 denotes negative and 1 denotes positive. The dataset contains 67,350 training samples, 873 validation samples, and 1,821 test samples. As part of the GLUE benchmark (Wang et al., 2018), the official dataset is available at `https://huggingface.co/datasets/nyu-mll/glue/viewer/sst2`.

CoLA (Warstadt et al., 2019) is a single-sentence binary classification dataset for linguistic acceptability. The corpus is collected from books and journals in linguistic theory. Each sentence is annotated as to whether it is grammatically acceptable, with labels 0 and 1, where 0 indicates ungrammatical and 1 indicates grammatical. The dataset contains 8,551 training samples, 1,043 validation samples, and 1,063 test samples. It is also part of the GLUE benchmark (Wang et al.,

2018) and is available at https://huggingface.co/datasets/nyu-mll/glue/viewer/cola.

MRPC (Dolan and Brockett, 2005) is a binary classification dataset consisting of sentence pairs extracted from online news sources, manually annotated to indicate whether the two sentences are semantically equivalent. The labels are 0 and 1, where 0 denotes not paraphrase and 1 denotes paraphrase. The dataset contains 3,668 training samples, 408 validation samples, and 1,725 test samples. It is also part of GLUE (Wang et al., 2018) and is available at https://huggingface.co/datasets/nyu-mll/glue/viewer/mrpc.

AG News (Zhang et al., 2015) is a single-sentence four-class news topic classification dataset collected from global news articles. The dataset covers four major categories: World (label 0), Sports (label 1), Business (label 2), and Science/Technology (label 3). It contains 120,000 training samples and 7,600 test samples. Since no validation split is provided, we randomly sample 20% of the training set as the validation set. The dataset is available at https://huggingface.co/datasets/fancyzhx/ag_news.

SST-5 (Socher et al., 2013) is a fine-grained single-sentence five-class sentiment classification dataset for movie reviews. The labels range from 0 to 4, representing very negative, negative, neutral, positive, and very positive, respectively. The dataset contains 8,544 training samples, 1,101 validation samples, and 2,210 test samples. It is available at https://huggingface.co/datasets/SetFit/sst5.

Emotion (Saravia et al., 2018) is a single-sentence six-class emotion classification dataset built from English Twitter messages. It covers six emotion categories: sadness (label 0), joy (label 1), love (label 2), anger (label 3), fear (label 4), and surprise (label 5). The dataset contains 16,000 training samples, 2,000 validation samples, and 2,000 test samples. It is available at https://huggingface.co/datasets/dair-ai/emotion.

Devign (Zhou et al., 2019) is a code defect detection dataset designed to identify whether a code snippet is potentially vulnerable to software attacks. The labels are 0 and 1, where 0 denotes safe code and 1 denotes vulnerable code. Following the CodeXGLUE (Lu et al., 2021), we split the dataset into training, validation, and test sets with an 8:1:1 ratio, resulting in 21,854 training samples and 2,732 samples for both validation and test sets. The dataset is available at https://huggingface.co/datasets/google/code_x_glue_cc_defect_detection.

BigCloneBench (Wang et al., 2020) is a manually annotated Java code clone detection benchmark for code clone detection and semantic similarity learning. The labels are 0 and 1, where 1 indicates that the code pair is semantically equivalent and 0 indicates non-equivalence. Since the original dataset is very large, we follow the CodeXGLUE (Lu et al., 2021) setting and use only 10% of the data for training and validation. The dataset is available at https://huggingface.co/datasets/google/code_x_glue_cc_clone_detection_big_clone_bench.

B Additional Implementation Details

In the preference data construction stage, for each ground-truth class in the eight datasets, we define a natural-language candidate semantic label, as shown in Table 6. The core motivation is to map the original discrete class label y_i into a semantic description c_j that can be directly interpreted by LMs.

In addition, during RM training, we concatenate each input sample with its candidate semantic label. Different datasets use different prefixes to explicitly distinguish task contexts and align the inputs with their semantic labels. The unified input format for the RM is $Prefix : x_i \,||\, c_{ij}$, the prefixes for different datasets are defined as follows:

- AG News: News
- SST-5 / SST-2 / CoLA / MRPC: Sentence
- Emotion: Text
- Devign / BigCloneBench: Code

C Additional Ablation Studies

This section further investigates ablation studies of ClaHF. Following Section 4.3, all experiments are conducted on the Emotion dataset with Qwen3 as the backbone model. All reported values are the mean and standard error over five runs with different random seeds.

Effect of the Value Function Loss in RL. Table 7 analyzes the influence of the value loss weight in the RL stage. In these experiments, all RM settings are fixed and α is set to 0.8, while only the value loss coefficient γ is varied.

Without the value loss (ClaHF-NoValue), the model still outperforms SFT, showing that

Dataset $\mathcal{D}$	**True Label y_i**	**Candidate Semantic Label c_j**
AG News	0	Prediction: the news topic is World
	1	Prediction: the news topic is Sports
	2	Prediction: the news topic is Business
	3	Prediction: the news topic is Science/Technology
SST-5	0	Prediction: the sentence is very negative
	1	Prediction: the sentence is negative
	2	Prediction: the sentence is neutral
	3	Prediction: the sentence is positive
	4	Prediction: the sentence is very positive
Emotion	0	Prediction: the emotion in the text is sadness
	1	Prediction: the emotion in the text is joy
	2	Prediction: the emotion in the text is love
	3	Prediction: the emotion in the text is anger
	4	Prediction: the emotion in the text is fear
	5	Prediction: the emotion in the text is surprise
SST-2	0	Prediction: the sentence is negative
	1	Prediction: the sentence is positive
CoLA	0	Prediction: the sentence is grammatically unacceptable
	1	Prediction: the sentence is grammatically acceptable
MRPC	0	Prediction: the two sentences are not semantically equivalent
	1	Prediction: the two sentences are semantically equivalent
Devign	0	Prediction: the code is non-vulnerable
	1	Prediction: the code is vulnerable
MRPC	0	Prediction: the two code snippets are not semantically similar
	1	Prediction: the two code snippets are semantically similar

Table 6: Ground-truth labels and their corresponding candidate semantic labels for the eight datasets.

Method	$\mathcal{L}_{\text{value}}$	Acc (↑)	F1 (↑)	ECE (↓)
NoValue	–	$93.20_{\pm 0.15}$	$89.59_{\pm 0.45}$	$5.67_{\pm 0.12}$
ClaHF	$\gamma = 0.25$	$\mathbf{93.62}_{\pm 0.10}$	$\mathbf{90.18}_{\pm 0.12}$	$\mathbf{5.06}_{\pm 0.21}$
ClaHF	$\gamma = 0.5$	$93.42_{\pm 0.13}$	$89.36_{\pm 0.26}$	$5.44_{\pm 0.12}$
ClaHF	$\gamma = 1.0$	$92.87_{\pm 0.17}$	$88.79_{\pm 0.34}$	$6.27_{\pm 0.34}$

Table 7: Effect of the value loss weight during RL.

Method	$\mathcal{L}_{\text{value}}$	Acc (↑)	F1 (↑)	ECE (↓)
SFT	–	$92.57_{\pm 0.14}$	$88.11_{\pm 0.40}$	$6.75_{\pm 0.22}$
ClaHF-Pair	–	$92.53_{\pm 0.13}$	$88.26_{\pm 0.23}$	$6.92_{\pm 0.38}$
ClaHF-Pair	$\gamma = 0.25$	$92.78_{\pm 0.14}$	$88.88_{\pm 0.12}$	$6.44_{\pm 0.21}$
ClaHF-Pair	$\gamma = 0.5$	$92.60_{\pm 0.10}$	$88.57_{\pm 0.09}$	$6.69_{\pm 0.28}$
ClaHF-Pair	$\gamma = 1.0$	$92.28_{\pm 0.14}$	$87.27_{\pm 0.18}$	$6.97_{\pm 0.22}$

Table 8: Effect of the value loss weight on ClaHF-Pair.

preference-driven policy optimization is effective by itself. However, its performance is clearly inferior to the full ClaHF, indicating that RL without value function regularization is more susceptible to high-variance reward signals.

When a moderate value loss is introduced, the model achieves the best results on Acc, F1, and ECE, validating the crucial role of the value function in stabilizing training and smoothing policy updates. As γ further increases, performance starts to degrade, suggesting that overly strong value constraints hinder effective preference-driven updates and thus weaken the benefits of ClaHF.

Table 8 reports a comparison between ClaHF-Pair with different value function loss weights γ and the SFT baseline. In this experiment, we keep the RM and all other PPO hyperparameters fixed and only vary the weight of the value function loss to analyze the model behavior in the absence of the Top-1 constraint, i.e., when equal constraints are imposed on all candidates without explicitly emphasizing the optimal candidate and the model is trained only to learn the relative ordering among candidates.

From the results, we observe that when the value function loss is not introduced and without the Top-1 constraint, the model can hardly outperform the SFT baseline on Acc, F1, and ECE, and even exhibits slight degradation on the calibration metric ECE. This indicates that relying solely on pairwise preference signals for policy updates makes it difficult for the model to stably learn effective discriminative boundaries.

After introducing a moderate value function loss, ClaHF-Pair achieves modest improvements over SFT, which further verifies that the value function can alleviate the instability caused by high-variance rewards in preference-based policy optimization. However, as γ increases further, the model performance drops and even falls below that of SFT, suggesting that an excessively large value loss weight overly suppresses policy updates when the Top-1 constraint is absent.

In summary, these results demonstrate that in ClaHF-Pair, although the value function loss can improve training stability, its effective range is clearly limited and requires careful tuning. This finding also provides additional evidence for our main conclusion that Top-1 and pairwise preference signals play complementary roles in ClaHF, and that their joint modeling is crucial for stable and efficient optimization of classification policies.

Further explanation of Figure 3. Further analysis of the interaction between α and γ reveals a clear synergistic effect. When α is small (i.e., the RM mainly relies on relative ranking signals), varying γ yields limited gains. In contrast, when α lies in the medium range, changes in γ have a more pronounced impact on both performance and calibration, suggesting that explicit preference signals facilitate the stabilizing role of the value function. However, when $\alpha = 1$, an excessively large γ suppresses the policy's responsiveness to preference signals and leads to performance degradation. This reveals a critical balance in ClaHF: preference guidance must be sufficiently strong to steer policy updates, while value constraints should stabilize training without over-regularization.

Finally, we observe that the distribution of ECE does not simply trade off with Acc and F1, but reaches its minimum near the same optimal region. This indicates that ClaHF simultaneously enhances discriminative performance and confidence calibration, rather than sacrificing one for the other.