S2T-RLHF: Hierarchical Credit Assignment for Stable Preference-Based RLHF

Wei Chen[1] **Guanghui Zhu**[1*] **Yafei Li**[2] **Limin Wang**[1] **Yihua Huang**[1]
[1]State Key Laboratory of Novel Software Technology, Nanjing University
[2]School of Computer Science and Artificial Intelligence, Zhengzhou University
weichen@smail.nju.edu.cn, zgh@nju.edu.cn, ieyfli@zzu.edu.cn,
lmwang@nju.edu.cn, yhuang@nju.edu.cn

Abstract

Reinforcement learning from human feedback (RLHF) with preference-based reward models often exhibits unstable training dynamics. A key contributing factor is that standard RLHF relies on a single sequence-level scalar reward, which is propagated to token-level policy updates and leaves credit assignment within a response inherently ambiguous. Recent work has attempted to address this issue by refining rewards into denser token-level supervision, often relying on the implicit assumption that finer-grained credit assignment improves optimization. We argue that this assumption is incomplete: when preference signals are noisy and only defined at the response level, overly fine-grained reward refinement can amplify reward uncertainty and destabilize learning. To address this problem, we propose a granularity-aware principle for hierarchical credit assignment, emphasizing stability-oriented reward design rather than maximal allocation precision. Under this principle, sentences serve as a natural intermediate granularity, balancing semantic coherence with robustness to token-level noise. Guided by this view, we introduce S2T-RLHF. This sentence-to-token reward decomposition framework first allocates sequence-level preference rewards across sentences and then applies bounded token-level refinement within each sentence, without reward-model retraining or token-level supervision. Experiments across multiple datasets and optimization settings show that S2T-RLHF improves training stability and robustness while maintaining competitive preference alignment.

1 Introduction

Reinforcement learning from human feedback (RLHF) has become a central paradigm for preference-based alignment of large language models (LLMs), enabling substantial improvements in helpfulness, harmlessness, and human preference alignment [1–4]. Despite this empirical success, RLHF often exhibits pronounced instability in long-horizon language generation. Common pathologies include length bias, reward concentration, and occasional reward collapse [5–8]. These issues persist despite careful hyperparameter tuning and regularization, suggesting underlying causes beyond implementation artifacts.

A key limitation of the standard RLHF pipeline lies in its reward formulation. The learned reward model assigns a single scalar score to an entire generated response, providing supervision only at the sequence level. While effective for optimizing global preferences, this scalar feedback offers no guidance on how credit should be assigned within a long generation trajectory [9]. Consequently, reinforcement learning must propagate a global preference signal across many decoding steps,

*Corresponding author.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

often resulting in high-variance gradients, unstable policy updates, and limited controllability over fine-grained generation behavior.

We observe that many of these failures arise from a structural mismatch between the semantic granularity of language and the granularity at which rewards are assigned and optimized. At its core, this mismatch reflects a tension between how human preferences are expressed and how language models generate text. Human raters typically judge responses through sentences or semantic units, such as whether a claim is correct, whether an instruction is followed, whether a refusal is appropriate, or whether the tone is safe and helpful. In contrast, LLMs generate discrete tokens whose isolated contributions are sub-symbolic and context-dependent, and often lack interpretable meaning outside their surrounding linguistic structure. Directly deriving token-level rewards from a sequence-level scalar is therefore ill-posed. The global score entangles sentence-level semantic quality with token-level realizations, making fine-grained credit assignment inherently ambiguous and unstable.

Recent efforts [10, 11] have largely treated RLHF instability as a problem of insufficiently dense supervision, motivating increasingly detailed forms of token-level credit assignment. This line of work is often driven by the implicit assumption that finer-grained credit assignment is inherently beneficial for optimization. However, this assumption conflates precision with stability. When preference signals are noisy and defined only at the response level, refining rewards beyond the semantic resolution at which preferences are expressed can amplify reward uncertainty, propagate local artifacts from the reward model, and destabilize learning dynamics rather than improve them.

We therefore suggest that stabilizing RLHF does not hinge on pursuing maximal credit-assignment resolution, but rather on designing reward signals whose granularity aligns with the semantic structure of language. From this perspective, reward decomposition should prioritize semantically meaningful units that support stable optimization, rather than maximizing allocation precision. This granularity-aware perspective reframes dense reward design in RLHF as a problem of semantic alignment and learning stability, rather than one of credit-assignment precision alone.

To operationalize this perspective, we adopt a hierarchical view of credit assignment that respects the semantic structure of language. Rather than treating dense reward design as a flat allocation problem over tokens, we emphasize that credit assignment should reflect the inherent hierarchy of natural language: sentences express coherent semantic intent, while tokens refine that intent through lexical and syntactic choices.

Based on this principle, we present Sentence-to-Token RLHF (S2T-RLHF), a concrete instantiation of hierarchical credit assignment for stable preference-based RLHF. S2T-RLHF first allocates a sequence-level preference reward over semantically stable sentence-level units through a structured cooperative allocation mechanism, and then applies bounded token-level refinement within each sentence. The method uses only sequence-level preference rewards and requires no reward-model retraining or token-level supervision. Importantly, S2T-RLHF is intended not as the only possible realization of this principle, but as a simple and practical instantiation of stability-oriented hierarchical credit assignment. We further provide theoretical discussion in Appendices C–D, analyzing how sequence-level rewards can induce global advantage alignment, long-horizon variance amplification, and length-dependent gradient scaling, and how the two-stage S2T design mitigates these effects. We summarize our main contributions as follows:

- We propose a credit assignment design principle for RLHF, emphasizing that training stability should be achieved by aligning reward granularity with the semantic structure of language, rather than by pursuing increasingly fine-grained credit assignment.
- Guided by this principle, we develop S2T-RLHF, a hierarchical semantic credit assignment method that decomposes a sequence-level preference signal into sentence-level allocations followed by token-level refinement, enabling dense and semantically grounded supervision without requiring additional annotations or reward-model retraining.
- We empirically evaluate S2T-RLHF across multiple datasets and optimization settings, showing that hierarchical credit assignment improves training stability and robustness while maintaining competitive preference alignment against existing credit assignment methods.

Code and additional resources are available at `https://pasalab.github.io/S2T-RLHF`.

2 Related Work

Large Language Models and Preference Alignment. Large language models pretrained on large-scale corpora have achieved strong performance in instruction following and open-ended generation [12–15]. However, maximum-likelihood training, while effective for next-token prediction, cannot fully capture desired behavioral properties [1], including subjective quality judgments [16, 17], normative constraints [18, 19], and task-specific preferences [20, 21]. This mismatch has motivated preference alignment, which aims to guide model behavior toward outputs that better reflect such criteria. A prominent framework is reinforcement learning from human feedback (RLHF), which learns a reward model from preference comparisons and optimizes the policy via reinforcement learning [1, 3]. Related paradigms include direct policy optimization [22], contrastive preference learning [23], odds ratio preference optimization [24], and other feedback-driven methods [25].

RLHF and Fine-Grained Reward Allocation. RLHF is a widely studied framework for aligning language models with preference signals from human feedback. In a typical pipeline, a reward model is trained from preference comparisons [26, 27], and the language model is then optimized with policy-gradient methods [3, 22, 28], most commonly proximal policy optimization (PPO) [29], to maximize the predicted reward. This framework is widely used as a standard approach for preference-based policy optimization in language generation [3, 30].

However, conventional RLHF assigns a single scalar reward to each generated response, producing a coarse and sparse learning signal that complicates credit assignment and can destabilize optimization. Recent work has explored fine-grained reward mechanisms to address this limitation. Chan et al. leverage reward-model attention weights to densify sequence-level rewards at the token level [31], but attention weights may not reliably reflect causal token contributions. Yoon et al. propose TLCR [32], which trains a discriminator to assign token-level rewards, requiring an additional auxiliary model. Zhong et al. reformulate RLHF as a token-wise MDP and propose RTO [9], which derives token-level rewards from preference data but inherits limitations from DPO's preference modeling and introduces complexity in the multi-stage pipeline. Chai et al. propose MA-RLHF [5], which groups tokens into macro actions to alleviate long-horizon credit assignment, but predefined macro actions remain coarse and may limit semantic adaptability. Cao et al. apply Shapley values to distribute sequence-level rewards based on marginal token contributions [33]; however, the resulting credits are tightly coupled to reward-model scalar evaluations and can be sensitive to reward noise. Beyond these representative approaches, a range of recent methods have investigated fine-grained reward allocation [34, 35, 10, 11]. Despite their different formulations, most existing methods primarily pursue denser token-level supervision, leaving underexplored the question of what reward granularity is semantically meaningful and optimization-stable for preference-based RLHF.

3 Preliminary

Reinforcement Learning from Human Feedback. Reinforcement learning from human feedback (RLHF) is a framework for aligning pretrained language models with human preferences. Let $\pi_\theta(y \mid x)$ denote a conditional language model that generates a response $y = (t_1, \ldots, t_T)$ given a prompt x. A reward model $R_\phi(x, y) \in \mathbb{R}$ is trained on human preference data to assign a scalar quality score to each complete response. RLHF then optimizes the model parameters θ to maximize the expected reward under R_ϕ, typically with a KL-divergence regularization term that prevents π_θ from deviating too far from a reference model π_{ref} [3]:

$$\max_\theta \ \mathbb{E}_{y\sim\pi_\theta}\big[R_\phi(x,y)\big] - \beta\,\mathrm{KL}\big(\pi_\theta \,\|\, \pi_{\mathrm{ref}}\big). \tag{1}$$

where $\beta > 0$ controls the trade-off between reward maximization and divergence from the reference model. In standard implementations, the reward model evaluates responses holistically, yielding a single global sequence-level signal that does not distinguish the contributions of individual tokens.

Nash Bargaining. Nash bargaining [36–38] is a classical solution concept in cooperative game theory that prescribes how to divide joint gains among multiple parties in a manner that is both fair and Pareto-efficient. Consider n players with utilities $u_1, \ldots, u_n$ and disagreement points $d_1, \ldots, d_n$, where d_i represents the utility player i receives if no agreement is reached. Let $\mathcal{U} \subseteq \mathbb{R}^n$ denote the

feasible set of utility vectors. The Nash bargaining solution is defined as:

$$u^\star \in \arg \max_{u \in \mathcal{U},\, u_i \geq d_i} \sum_{i=1}^{n} \log\big(u_i - d_i\big). \tag{2}$$

This solution is characterized by symmetry, scale invariance, Pareto efficiency, and independence of irrelevant alternatives. It provides a principled framework for allocating utility among interacting components.

Dirichlet Distribution. The Dirichlet distribution is a probability distribution defined over the probability simplex, consisting of vectors $p = (p_1, \ldots, p_K)$ satisfying $p_k \geq 0$ and $\sum_{k=1}^{K} p_k = 1$. It is parameterized by a concentration vector $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_K) \in (0, \infty)^K$. We write $p \sim \mathrm{Dir}(\boldsymbol{\alpha})$ if p has density

$$\mathrm{Dir}(p; \boldsymbol{\alpha}) = \frac{\Gamma\Big(\sum_{k=1}^{K} \alpha_k\Big)}{\prod_{k=1}^{K} \Gamma(\alpha_k)} \prod_{k=1}^{K} p_k^{\alpha_k - 1}. \tag{3}$$

The mean of the Dirichlet distribution is given by $\mathbb{E}[p_k] = \alpha_k / \alpha_0$ for all k, where $\alpha_0 = \sum_{k=1}^{K} \alpha_k$ denotes the total concentration parameter, which controls how peaked or diffuse the distribution is over the simplex. As a distribution over normalized proportions, the Dirichlet distribution is widely used to model uncertainty in categorical probabilities and to represent relative weights over multiple outcomes, especially as a conjugate prior for categorical or multinomial variables.

4 Methodology

4.1 Semantically-Grounded Reward Decomposition

In standard RLHF, a single sequence-level scalar reward provides only coarse supervision over a long generation trajectory, leaving fine-grained credit assignment fundamentally underspecified. This mismatch often leads to unstable optimization dynamics, overly localized updates, and limited interpretability of learned behaviors. To address this challenge, we propose **S2T-RLHF**, a two-stage hierarchical reward decomposition framework that transforms a sequence-level preference signal into semantically grounded dense rewards without requiring additional annotations, reward-model retraining, or partial-sequence queries.

As illustrated in Fig. 1, S2T-RLHF decomposes reward assignment across semantic granularities. In the first stage, the global sequence-level reward is allocated across sentences, producing a structured and interpretable intermediate representation that captures the relative semantic contributions of reasoning steps or conversational units while preserving total reward mass. In the second stage, conditioned on this sentence-level allocation, each sentence reward is redistributed across its constituent tokens, enabling fine-grained credit assignment aligned with local lexical and syntactic realizations. By explicitly separating semantic allocation from token-level refinement, S2T-RLHF provides stable, semantically coherent dense rewards suitable for RLHF optimization.

4.2 Semantic Granularity and Design Rationale

Sentence-level and token-level representations capture different aspects of response quality. At the sentence level, complete semantic units and logical coherence shape the global structure and overall quality of a response. At the token level, variations mainly affect local precision and the expression of critical information. Thus, reward assignment at a single granularity cannot capture both global semantic quality and fine-grained lexical accuracy.

At the sentence level, reward decomposition should emphasize fairness across sentences. Notably, this goal runs counter to the standard reinforcement learning intuition of concentrating reward on the most salient components. Rather than amplifying a single high-reward sentence, sentence-level allocation aims to prevent semantically important sentences from being under-rewarded or suppressed. Relying solely on a sequence-level scalar reward can introduce length bias, as longer sentences exert disproportionate influence by containing more tokens. Moreover, sentences often play distinct but complementary roles within a logical structure, and uneven reward allocation can suppress essential components—an effect exacerbated by coupled instabilities in sequence-level RLHF, including

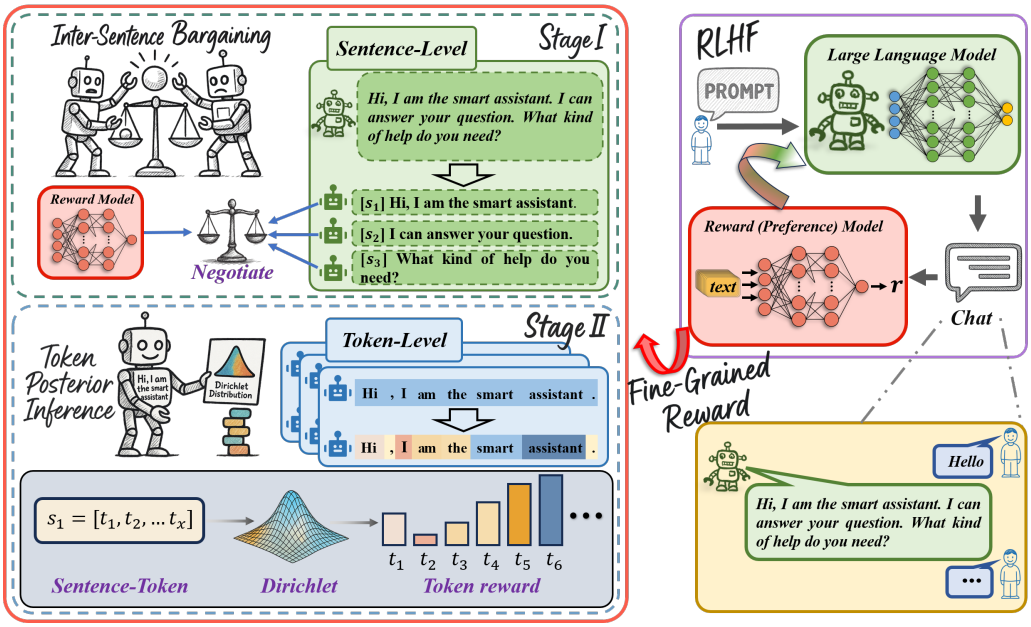


Figure 1: Overview of S2T-RLHF. A global sequence-level reward is first decomposed into sentence-level credits via inter-sentence bargaining (Stage I), and then refined into token-level rewards through Dirichlet posterior inference (Stage II), enabling stable and interpretable credit assignment in RLHF without modifying the reward model.

advantage alignment, variance amplification, and length-dependent scaling (Appendix C). These considerations motivate a sentence-level allocation principle that balances contributions across sentences under a shared reward budget.

In contrast, token-level credit assignment should emphasize relative importance rather than fairness. Tokens differ substantially in information density: key nouns, verbs, entities, numerical values, and logical operators carry significantly higher semantic weight than function words or fillers. Errors in critical tokens (e.g., factual entities or negations) can drastically alter meaning, whereas minor lexical variations often have limited impact. Uniform token-level reward assignment, therefore, suffers from signal dilution, obscuring informative learning signals. Token-level rewards should instead reflect relative importance under uncertainty, enabling sparse yet informative credit assignment within each sentence. This asymmetry between sentence-level allocation and token-level allocation motivates the use of distinct mechanisms in the two stages of S2T-RLHF.

4.3 Sentence-Level Reward Allocation

RLHF assigns rewards at the sequence level, leaving the contribution of individual sentences underdetermined. We resolve this ambiguity by formulating sentence-level reward allocation as a cooperative bargaining problem, where sentences negotiate their shares of a fixed global reward based on their semantic influence. We use Nash bargaining not for its game-theoretic interpretation, but because it induces a scale-invariant, symmetric, and smooth allocation under a fixed reward budget. Other mechanisms with similar properties could be substituted.

Sentence embedding and semantic perturbation. Given a response $x = [s_1, \ldots, s_K]$, each sentence s_i is mapped to a contextualized semantic embedding $z_i = \mathrm{Enc}(s_i) \in \mathbb{R}^d$ using the LLM encoder. These embeddings capture sentence-level semantics and form the atomic units for credit assignment. To estimate how the sequence-level reward depends on individual sentences, we construct semantic perturbations by replacing s_i with a set of alternative variants $\{s_i^{(k)}\}$ (Appendix B.3). For each perturbation, we measure the induced changes in both reward and representation:

$$\Delta R_{i,k} = R(x) - R(x^{(i \to k)}), \qquad \delta_{i,k} = z_i - z_i^{(k)}. \tag{4}$$

These paired differences capture the local sensitivity of the global reward to semantic variations of sentence i under the model's contextual encoding.

Latent semantic influence estimation. The perturbation signals above provide local observations of how sentence-level semantic changes affect the sequence-level reward. We summarize these observations by estimating a latent *semantic influence vector* $v_i \in \mathbb{R}^d$ for each sentence via ridge-regularized regression:

$$v_i = \arg\min_v \sum_k \left(\Delta R_{i,k} - \langle v, \delta_{i,k} \rangle\right)^2 + \lambda \|v\|^2. \tag{5}$$

This vector captures a reward-sensitive semantic direction for sentence i, reflecting how the reward model responds to small semantic perturbations and to the LLM's autoregressive contextual structure. The semantic bargaining framework does not depend on any specific analytic properties of the reward model. To model inter-sentence dependencies in the reward-relevant semantic space, we represent each sentence by its influence vector and define

$$M = V^\top V, \qquad V = [v_1, \ldots, v_K] \in \mathbb{R}^{d \times K}, \tag{6}$$

where M_{ij} measures the degree of alignment between the influence vectors of sentences i and j.

Semantic bargaining formulation. We introduce positive bargaining coefficients $\beta \in \mathbb{R}^K_{>0}$ to form a cooperative semantic direction from sentence-level influences, and define the latent utility of sentence i as its alignment with this direction:

$$h = \sum_{j=1}^K \beta_j v_j, \qquad \tilde{u}_i = \langle v_i, h \rangle = (M\beta)_i, \qquad u_i = \max(\tilde{u}_i, \varepsilon), \tag{7}$$

where $\varepsilon > 0$ ensures well-defined logarithms and reciprocal updates. To allocate reward proportionally to these latent utilities, we maximize the multiplicative bargaining objective

$$\mathcal{B}(\beta) = \sum_{i=1}^K \log u_i. \tag{8}$$

This objective follows the spirit of classical Nash bargaining and is used here as a principled allocation criterion rather than a global optimization guarantee. It yields a scale-invariant and symmetric cooperative allocation in the latent utility space. In practice, we compute the bargaining coefficients through a reciprocal fixed-point condition. In the idealized case without the ϵ-safeguard, any positive fixed point satisfies

$$M\beta = \beta^{\odot -1}, \qquad \text{equivalently} \qquad u_i \beta_i = 1 \quad \forall i, \tag{9}$$

where $u = M\beta$. This Nash-style balance condition does not require all latent utilities to be equal, and therefore allows non-uniform sentence-level allocation. Proposition A.2 gives the formal stationarity characterization.

Fixed-point solver and reward assignment. Instead of solving for τ explicitly, we approximate the bargaining equilibrium via a smooth damped fixed-point iteration:

$$\beta^{(t+1)} = (1-\eta)\,\beta^{(t)} + \eta \left(\max(M\beta^{(t)}, \varepsilon)\right)^{\odot -1}, \tag{10}$$

where $\eta \in (0, 1]$ and $\varepsilon > 0$. The update is used as a numerical solver rather than a new optimization algorithmic contribution. The resulting fixed-point equation admits at least one solution in the positive orthant under mild regularity conditions (Corollary A.3), and the safeguarded iteration is well-defined and preserves positivity (Lemma A.4); detailed diagnostics are provided in Appendix B.1. After convergence, sentence-level rewards are obtained by normalizing the latent utilities:

$$r_i = \frac{u_i}{\sum_j u_j} R(x), \qquad u = \max(M\beta, \varepsilon). \tag{11}$$

These normalized values form a semantically grounded decomposition of the global reward, preserve the total reward mass $\sum_i r_i = R(x)$, and serve as the input for the token-level inference stage.

4.4 Token-Level Reward Allocation

Although sentence-level rewards provide a useful intermediate signal, they do not specify how credit should be assigned within a sentence. Since human preferences may depend on a small subset of decisive tokens, uniform allocation can dilute informative learning signals. We therefore formulate token-level refinement as a simplex-constrained allocation problem. Given contextualized token representations, a lightweight Dirichlet Token Allocation Network (DTAN) predicts non-negative token weights that sum to one within each sentence. The resulting allocation is deterministic, normalized, differentiable, and serves as a bounded refinement of the sentence-level reward.

Dirichlet prior from contextual representations. Let $h_{s,t}$ denote the hidden representation of token t in sentence s. DTAN maps $h_{s,t}$ to a positive Dirichlet concentration parameter:

$$\alpha_{s,t} = \mathrm{softplus}(f_\phi(h_{s,t})), \tag{12}$$

where f_ϕ is a lightweight multilayer perceptron. The concentration parameters encode a contextual prior over token contributions: larger values indicate stronger allocation weight, while smaller values reflect lower confidence. Grounding this prior in token representations allows DTAN to capture semantic and positional cues within each sentence.

Dirichlet-based token allocation. Given the concentration parameters, we model token allocation within sentence s as $c_s \sim \mathrm{Dir}(\alpha_{s,1}, \ldots, \alpha_{s,L_s})$, where L_s is the sentence length. We use the Dirichlet expectation as the deterministic token contribution:

$$c_{s,t} = \frac{\alpha_{s,t}}{\sum_{k=1}^{L_s} \alpha_{s,k}}.$$

Thus $c_{s,t} \geq 0$ and $\sum_{t=1}^{L_s} c_{s,t} = 1$, yielding a stable and differentiable allocation of sentence-level reward mass across tokens.

Reward decomposition into token-level signals. In practice, the decomposition is implemented as per-token modulation of the sequence-level advantage in PPO, with additional clipping for stability (Appendix B.2). Conceptually, if the sentence-level reward is $R^{(s)}$, the token-level signal assigned to token t is

$$r_{s,t} = R^{(s)} \cdot c_{s,t}. \tag{13}$$

This decomposition preserves reward mass, $\sum_{t=1}^{L_s} r_{s,t} = R^{(s)}$, and yields a valid token-level assignment (Proposition A.6). The resulting signals are differentiable with respect to the DTAN parameters (Lemma A.7) and can be integrated into policy-gradient optimization (Corollary A.8).

Joint optimization with RLHF. DTAN is not trained during reward-model fitting; it is optimized jointly with the policy during the RLHF phase. The decomposed token-level signals are used to construct fine-grained advantage estimates for policy optimization, while the underlying reward model remains fixed. Let $A_{s,t}$ denote the advantage associated with token t in sentence s. For PPO-style optimization, the policy objective can be written as

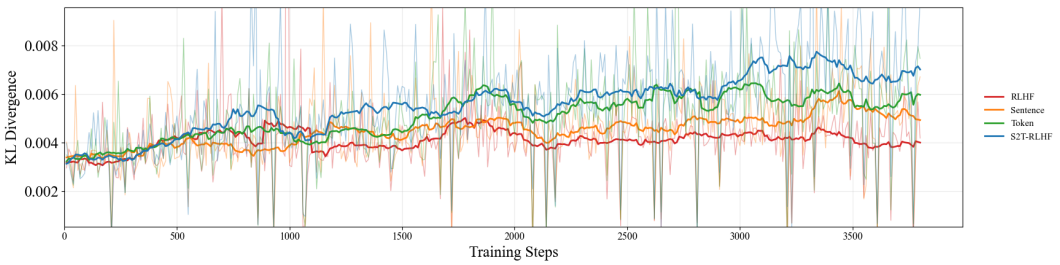
$$\mathcal{L}_{\mathrm{RLHF}}(\theta) = -\mathbb{E}\big[A_{s,t} \log \pi_\theta(y_{s,t} \mid x, y_{s,<t})\big]. \tag{14}$$

Through repeated policy updates, DTAN learns allocation weights that better guide policy improvement without requiring token-level supervision. This design enables bounded token-level refinement while remaining compatible with standard RLHF training pipelines.

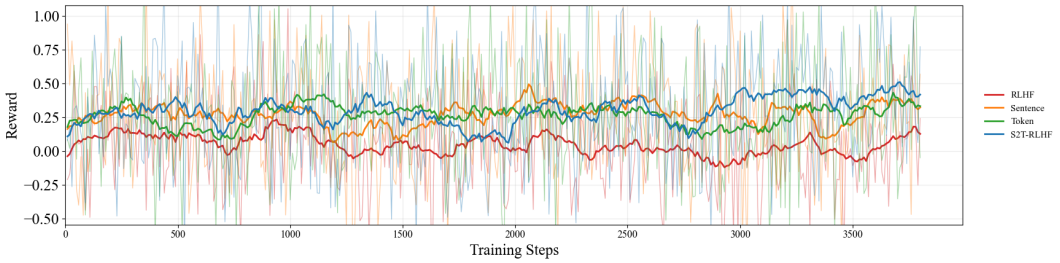
5 Experiment

5.1 Experimental Setup

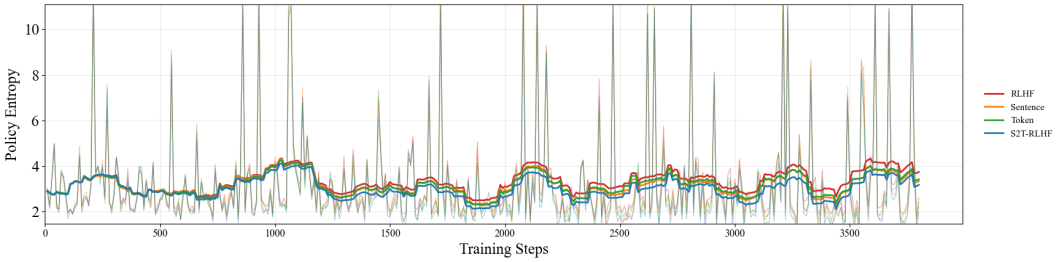
We use Gemma-2-9B (Team et al., 2024) as the policy model and RM-Gemma-2B (Dong et al., 2023) as the reward model. HH-RLHF (Bai et al., 2022) serves as the primary training and analysis dataset under a standard PPO-based RLHF pipeline. To assess robustness and cross-dataset generalization, we further evaluate the trained policies zero-shot on AdvBench, RealToxicityPrompts, ToxiGen, and SafeRLHF in Appendix H. Implementation details are deferred to Appendix E.1. Unless otherwise specified, all experiments are conducted on a single A100-SXM4-80GB GPU.



(a) KL divergence to the reference policy during training.



(b) Reward trajectory during training.



(c) Policy entropy evolution during training.

Figure 2: Training dynamics of the policy.

5.2 Main Results on Training Dynamics and Stability

We evaluate three structural variants within the S2T framework: (i) *sentence-only allocation*, which assigns reward at the sentence level and distributes it uniformly within each sentence; (ii) *token-only refinement*, which applies bounded token-level reweighting under uniform sentence-level allocation; and (iii) the full *S2T hierarchy*, which combines sentence-level allocation with bounded token-level refinement. For reference, we also include standard sequence-level RLHF, where a single scalar reward is propagated to token-level updates under vanilla PPO.

We characterize training dynamics using three complementary signals: KL divergence from the reference policy, policy entropy, and the mean reward trajectory over training steps. KL divergence reflects the magnitude and accumulation of policy updates, entropy captures changes in exploration and potential mode collapse, and the mean reward trajectory highlights temporal fluctuations in the learning signal. We visualize representative cases in Figures 2a–2c. Solid lines show moving-average smoothed trajectories, and faint lines indicate raw measurements.

Figure 2a shows that all methods exhibit comparable KL divergence in the early stage, indicating similar update scales when training is most fragile. In the mid-to-late stage, S2T progressively departs further from the reference policy than the other variants, yielding the ordering S2T $>$ Token $>$ Sentence $>$ RLHF in KL. Importantly, this larger KL does not coincide with instability. Figure 2b shows the same ordering in final mean reward, suggesting that the additional policy drift induced by S2T translates into consistent reward gains while avoiding collapse. Figure 2c further indicates that entropy remains in a comparable range across methods, with no signs of degenerate collapse. S2T exhibits slightly lower late-stage entropy, consistent with a more confident policy while maintaining stable optimization. Finally, the sentence-level variant already provides a favorable trade-off in the early-to-mid stage, achieving higher reward at smaller KL than token-level refinement, supporting

Table 1: Pairwise preference evaluation under a GPT-4o-mini evaluator.

METHOD	WIN (%)		LOSE (%)	TIE (%)
VS. RLHF	**53.0**	>	36.5	10.5
VS. ABC	**51.5**	>	38.0	10.4
VS. SCAR	**48.1**	>	43.1	8.8

Table 2: Pairwise preference evaluation under an RM-Gemma-7B evaluator.

METHOD	WIN (%)		LOSE (%)	TIE (%)
VS. RLHF	**51.1**	>	39.5	9.4
VS. ABC	**48.7**	>	43.8	7.5
VS. SCAR	**45.8**	>	44.1	10.1

our main claim that semantically aligned credit assignment improves reward efficiency and stabilizes learning.

Phase-wise stability metrics. To strengthen the trajectory-level evidence in Figures 2a–2c, we report phase-wise quantitative stability metrics aggregated over three random seeds in Appendix F. We compute summary statistics over an early window (0–1000 steps) and a late window (5000–6000 steps), enabling a controlled comparison of optimization stability across phases. Consistent with the trends in the trajectories, S2T-RLHF exhibits the lowest volatility of the optimization signals and avoids the larger policy drift observed for token-only refinement.

Early-stage stress tests. In addition, since RLHF failures often concentrate in the early stage, we conduct stress tests under more aggressive learning rates and KL settings and analyze the resulting early-stage dynamics in Appendix G. The stress tests highlight that token-only refinement is more prone to early-stage instability under aggressive settings, whereas sentence-level allocation remains comparatively stable. By combining sentence-level allocation with bounded within-sentence refinement, S2T attenuates the instability induced by fine-grained credit while retaining its late-stage optimization benefits. Overall, the sentence-level variant already improves early-stage stability, whereas token-level refinement mainly affects late-stage optimization behavior. By combining both, S2T yields more consistent training dynamics together with stronger final rewards than using either granularity in isolation.

Computational cost. We provide a detailed runtime analysis in Appendix J. S2T-RLHF introduces additional computation mainly from perturbation-based sentence influence estimation, while the bargaining solver and DTAN refinement are lightweight.

5.3 Preference Alignment Performance

We evaluate preference alignment performance using pairwise comparisons, focusing on whether sentence-to-token credit assignment preserves alignment quality while improving training stability.

Baselines. We compare S2T-RLHF with three representative approaches: **RLHF** [3], which applies sequence-level scalar rewards; **ABC** [31], which performs dense token-level credit assignment via attention-based allocation; and **SCAR** [33], which decomposes sequence-level rewards into token-level signals using Shapley-value-based marginal contributions.

Evaluation Protocol. We perform pairwise preference evaluation using two complementary evaluators. First, we adopt an *LLM-as-a-Judge* [39] protocol based on **GPT-4o-mini** [40, 41] to capture large-scale and consistent preference trends. Second, we evaluate the same model pairs using reward-model-based preferences from **RM-Gemma-7B** [42] to assess robustness with respect to the choice of evaluator. All methods are trained under identical optimization budgets and compared using the same prompts and evaluation procedure.

Results and Analysis. Table 1 reports win–tie–lose statistics under an *LLM-as-a-Judge* protocol. Across all baselines, S2T-RLHF attains a consistent win advantage (Win $>$ Lose), indicating that sentence-to-token credit assignment preserves strong preference alignment relative to RLHF, ABC, and SCAR. Table 2 reports the same comparison using a learned reward model (RM-Gemma-7B) as the evaluator. While absolute win rates shift across evaluators, the conclusion is unchanged: S2T-RLHF maintains a win advantage over each baseline (Win $>$ Lose), suggesting robustness to the choice of preference signal. Additional evaluations with Qwen-based and larger Gemma-based evaluators are provided in Appendix H.2, further supporting the robustness of the preference-alignment results across evaluator families and scales.

Table 3: Ablation preference evaluation under an *LLM-as-a-Judge* evaluator.

METHOD	WIN (%)		LOSE (%)	TIE (%)
VS. RLHF	**53.0**	>	36.5	10.5
VS. SENT.	**54.5**	>	35.4	10.1
VS. TOKEN	**48.6**	>	41.6	9.8

Table 4: Ablation preference evaluation under a reward-model evaluator.

METHOD	WIN (%)		LOSE (%)	TIE (%)
VS. RLHF	**51.1**	>	39.5	9.4
VS. SENT.	**49.5**	>	44.6	5.9
VS. TOKEN	**52.2**	>	43.5	4.3

Cross-dataset preference and reward dynamics. Additional pairwise preference results across datasets are reported in Appendix H.1. Across most settings, S2T-RLHF maintains a consistent win advantage over the baselines, reinforcing that the proposed stability-oriented decomposition does not sacrifice preference alignment. Appendix H.4 also reports training-time reward trajectories, where S2T-RLHF achieves higher rewards than baseline methods, highlighting more effective utilization of the reward signal.

Failure case of fine-grained decomposition. In addition to these alignment results, we observe a training collapse phenomenon for fine-grained token-level decomposition methods under more aggressive optimization. As illustrated in Appendix I, such overly fine-grained reward decomposition can amplify optimization noise and lead to unstable training behavior at larger learning rates.

Overall, preference evaluations verify that a stability-oriented reward decomposition achieves more stable training while maintaining strong alignment performance.

5.4 Ablation Study

Tables 3 and 4 report pairwise preference outcomes for ablations of the S2T hierarchy under an *LLM-as-a-Judge* protocol and a reward-model-based evaluator, respectively. Across evaluators, S2T-RLHF is consistently preferred over all granularity variants. This indicates that combining sentence-level allocation with bounded within-sentence refinement yields the most effective overall credit-assignment structure among the tested designs. Mechanism-swap ablations in Appendix H.3 further show that the gains come from both the two-stage hierarchy and the specific Nash/DTAN allocation mechanisms.

Discussion. The results suggest that the benefit of reward decomposition in RLHF should not be understood solely as a consequence of finer credit assignment. Instead, what matters is whether the reward signal is decomposed at a granularity that matches the semantic structure of language and remains stable under policy optimization. This distinction helps explain why token-only refinement can improve late-stage optimization but is more vulnerable to early-stage instability, whereas sentence-level allocation provides a more robust intermediate abstraction. From this perspective, S2T-RLHF is not merely a denser reward-shaping method, but an instance of a broader granularity-aware reward design principle for preference-based RLHF. We provide further theoretical discussion in Appendices C–D, which analyze the instability of sequence-level RLHF and connect it to the two-stage design of S2T-RLHF.

6 Conclusion

This work revisits reward design in RLHF from the perspective of semantic granularity and training stability. Through extensive empirical analysis, we show that RLHF instability is not an implementation artifact, but a structural consequence of driving token-level optimization with low-resolution, sequence-level rewards. Our results challenge the common assumption that increasingly fine-grained credit assignment is inherently beneficial, and instead highlight the importance of granularity-aware reward design that prioritizes semantic structure and stable learning dynamics. Within this perspective, we propose S2T-RLHF as a concrete instantiation, combining inter-sentence bargaining with lightweight token-level refinement to enable end-to-end credit assignment without modifying the reward model or requiring token-level supervision.

Acknowledgments and Disclosure of Funding

This work was partially supported by the National Natural Science Foundation of China (Grant Nos. 62572232 and 62372416), the Frontier Technology R&D Program of Jiangsu Province (No. BF2024005), the Fundamental and Interdisciplinary Disciplines Breakthrough Plan of the Ministry of Education of China (No. JYB2025XDXM118), the "111 Center" (No. B26023), the Nanjing University Kunpeng & Ascend Center of Cultivation, the Collaborative Innovation Center of Novel Software Technology and Industrialization, Jiangsu, China, and the Henan Provincial Major Science and Technology Grant (No. 261000210200).

The authors declare no competing interests.

References

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

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

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

[4] Raghav Sharma, Manan Mehta, and Sai Tiger Raina. Rlhf: A comprehensive survey for cultural, multimodal and low latency alignment methods. *arXiv preprint arXiv:2511.03939*, 2025.

[5] Yekun Chai, Haoran Sun, Huang Fang, Shuohuan Wang, Yu Sun, and Hua Wu. Ma-rlhf: Reinforcement learning from human feedback with macro actions. *arXiv preprint arXiv:2410.02743*, 2024.

[6] Ted Moskovitz, Aaditya K Singh, DJ Strouse, Tuomas Sandholm, Ruslan Salakhutdinov, Anca Dragan, and Stephen Marcus McAleer. Confronting reward model overoptimization with constrained RLHF. In *The Twelfth International Conference on Learning Representations*, 2024.

[7] Rui Yang, Ruomeng Ding, Yong Lin, Huan Zhang, and Tong Zhang. Regularizing hidden states enables learning generalizable reward model for llms. *Advances in Neural Information Processing Systems*, 37:62279–62309, 2024.

[8] Yuchun Miao, Sen Zhang, Liang Ding, Rong Bao, Lefei Zhang, and Dacheng Tao. Inform: Mitigating reward hacking in rlhf via information-theoretic reward modeling. *Advances in Neural Information Processing Systems*, 37:134387–134429, 2024.

[9] Han Zhong, Zikang Shan, Guhao Feng, Wei Xiong, Xinle Cheng, Li Zhao, Di He, Jiang Bian, and Liwei Wang. DPO meets PPO: Reinforced token optimization for RLHF. In *Forty-second International Conference on Machine Learning*, 2025.

[10] Liyuan Mao, Haoran Xu, Amy Zhang, Weinan Zhang, and Chenjia Bai. Information-theoretic reward decomposition for generalizable rlhf. *arXiv preprint arXiv:2504.06020*, 2025.

[11] Guofu Xie, Yunsheng Shi, Hongtao Tian, Ting Yao, and Xiao Zhang. Capo: Towards enhancing llm reasoning through verifiable generative credit assignment. *arXiv e-prints*, pages arXiv–2508, 2025.

[12] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

[13] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[14] Jason Wei, Maarten Bosma, Vincent Y Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. Finetuned language models are zero-shot learners. *arXiv preprint arXiv:2109.01652*, 2021.

[15] Wei Chen, Guanghui Zhu, Zhongliang Cai, and Yihua Huang. Unlocking latent personalization in llms. *arXiv preprint arXiv:2609.33182*, 2026.

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

[17] Dongyoung Kim, Kimin Lee, Jinwoo Shin, and Jaehyung Kim. Spread preference annotation: Direct preference judgment for efficient llm alignment. *arXiv preprint arXiv:2406.04412*, 2024.

[18] Richard Ngo, Lawrence Chan, and Sören Mindermann. The alignment problem from a deep learning perspective. *arXiv preprint arXiv:2209.00626*, 2022.

[19] Akifumi Wachi, Thien Tran, Rei Sato, Takumi Tanabe, and Youhei Akimoto. Stepwise alignment for constrained language model policy optimization. *Advances in Neural Information Processing Systems*, 37:104471–104520, 2024.

[20] Yifan Zhong, Chengdong Ma, Xiaoyuan Zhang, Ziran Yang, Haojun Chen, Qingfu Zhang, Siyuan Qi, and Yaodong Yang. Panacea: Pareto alignment via preference adaptation for llms. *Advances in Neural Information Processing Systems*, 37:75522–75558, 2024.

[21] Kasia Kobalczyk and Mihaela van der Schaar. Preference learning for AI alignment: a causal perspective. In *Forty-second International Conference on Machine Learning*, 2025.

[22] Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances in neural information processing systems*, 36:53728–53741, 2023.

[23] Joey Hejna, Rafael Rafailov, Harshit Sikchi, Chelsea Finn, Scott Niekum, W Bradley Knox, and Dorsa Sadigh. Contrastive preference learning: learning from human feedback without rl. *arXiv preprint arXiv:2310.13639*, 2023.

[24] Jiwoo Hong, Noah Lee, and James Thorne. Orpo: Monolithic preference optimization without reference model. *arXiv preprint arXiv:2403.07691*, 2024.

[25] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

[26] Leo Gao, John Schulman, and Jacob Hilton. Scaling laws for reward model overoptimization. In *International Conference on Machine Learning*, pages 10835–10866. PMLR, 2023.

[27] Dzmitry Bahdanau, Felix Hill, Jan Leike, Edward Hughes, Arian Hosseini, Pushmeet Kohli, and Edward Grefenstette. Learning to understand goal specifications by modelling reward. *arXiv preprint arXiv:1806.01946*, 2018.

[28] Weiran Yao, Shelby Heinecke, Juan Carlos Niebles, Zhiwei Liu, Yihao Feng, Le Xue, Rithesh Murthy, Zeyuan Chen, Jianguo Zhang, Devansh Arpit, et al. Retroformer: Retrospective large language agents with policy gradient optimization. *arXiv preprint arXiv:2308.02151*, 2023.

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

[30] Fan Wu, Huseyin A Inan, Arturs Backurs, Varun Chandrasekaran, Janardhan Kulkarni, and Robert Sim. Privately aligning language models with reinforcement learning. *arXiv preprint arXiv:2310.16960*, 2023.

[31] Alex James Chan, Hao Sun, Samuel Holt, and Mihaela Van Der Schaar. Dense reward for free in reinforcement learning from human feedback. In *International Conference on Machine Learning*, pages 6136–6154. PMLR, 2024.

[32] Eunseop Yoon, Hee Suk Yoon, SooHwan Eom, Gunsoo Han, Daniel Nam, Daejin Jo, Kyoung-Woon On, Mark Hasegawa-Johnson, Sungwoong Kim, and Chang Yoo. Tlcr: Token-level continuous reward for fine-grained reinforcement learning from human feedback. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 14969–14981, 2024.

[33] Meng Cao, Shuyuan Zhang, Xiao-Wen Chang, and Doina Precup. Scar: Shapley credit assignment for more efficient rlhf. *arXiv preprint arXiv:2505.20417*, 2025.

[34] Jiahui Li, Tai-wei Chang, Fengda Zhang, Kun Kuang, and Long Chen. R3hf: Reward redistribution for enhancing reinforcement learning from human feedback. *arXiv preprint arXiv:2411.08302*, 2024.

[35] Ahmad Rashid, Ruotian Wu, Julia Grosse, Agustinus Kristiadi, and Pascal Poupart. A critical look at tokenwise reward-guided text generation. *arXiv preprint arXiv:2406.07780*, 2024.

[36] Abhinay Muthoo. *Bargaining theory with applications*. Cambridge University Press, 1999.

[37] Aviv Navon, Aviv Shamsian, Idan Achituve, Haggai Maron, Kenji Kawaguchi, Gal Chechik, and Ethan Fetaya. Multi-task learning as a bargaining game. *arXiv preprint arXiv:2202.01017*, 2022.

[38] Yi Zeng, Xuelin Yang, Li Chen, Cristian Ferrer, Ming Jin, Michael Jordan, and Ruoxi Jia. Fairness-aware meta-learning via nash bargaining. *Advances in Neural Information Processing Systems*, 37:83235–83267, 2024.

[39] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in neural information processing systems*, 36:46595–46623, 2023.

[40] Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*, 2024.

[41] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[42] Hanze Dong, Wei Xiong, Deepanshu Goyal, Yihan Zhang, Winnie Chow, Rui Pan, Shizhe Diao, Jipeng Zhang, Kashun Shum, and Tong Zhang. Raft: Reward ranked finetuning for generative foundation model alignment. *Transactions on Machine Learning Research*, 2023.

[43] Akash Velu, Skanda Vaidyanath, and Dilip Arumugam. Hindsight-dice: Stable credit assignment for deep reinforcement learning. *arXiv preprint arXiv:2307.11897*, 2023.

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

[45] Andy Zou, Zifan Wang, Nicholas Carlini, Milad Nasr, J Zico Kolter, and Matt Fredrikson. Universal and transferable adversarial attacks on aligned language models. *arXiv preprint arXiv:2307.15043*, 2023.

[46] Samuel Gehman, Suchin Gururangan, Maarten Sap, Yejin Choi, and Noah A Smith. Realtoxicityprompts: Evaluating neural toxic degeneration in language models. *arXiv preprint arXiv:2009.11462*, 2020.

[47] Thomas Hartvigsen, Saadia Gabriel, Hamid Palangi, Maarten Sap, Dipankar Ray, and Ece Kamar. Toxigen: A large-scale machine-generated dataset for adversarial and implicit hate speech detection. *arXiv preprint arXiv:2203.09509*, 2022.

A Proofs

Lemma A.1 (Stationarity in open domains)**.** *Let $\mathcal{D} \subset \mathbb{R}^n$ be open and $\phi : \mathcal{D} \to \mathbb{R}$ continuously differentiable. Then $x^\star \in \mathcal{D}$ satisfies $\nabla\phi(x^\star) = 0$ if and only if $x^\star$ is a stationary point of the unconstrained problem $\min_{x\in\mathcal{D}} \phi(x)$.*

Proof. Since $\mathcal{D}$ is open, there are no boundary constraints, and first-order stationarity reduces to $\nabla\phi(x^\star) = 0$. □

Idealized analysis vs. practical safeguard. The results in this section analyze the idealized fixed-point equation $M\beta = \beta^{\odot-1}$ (i.e., without the ε-safeguard) to elucidate its Nash-style structure. Our implementation uses the safeguarded utilities $u = \max(M\beta, \varepsilon)$ to ensure well-defined logarithms and reciprocal updates; see Appendix B.1 for details.

Proposition A.2 (Stationarity of the fixed-point equilibrium)**.** *Any positive fixed point $\beta^\star \in \mathbb{R}^K_{++}$ satisfying $M\beta^\star = (\beta^\star)^{\odot-1}$ is a stationary point of the potential function*

$$\phi(\beta) := \frac{1}{2}\beta^\top M\beta - \sum_{i=1}^{K} \log \beta_i. \tag{15}$$

Moreover, with $u = M\beta^\star$, the fixed point satisfies the Nash-style equalization condition $u_i\beta_i^\star = 1$ for all i.

Proof. We first establish stationarity of ϕ. Since ϕ is continuously differentiable on the open domain $\mathbb{R}^K_{++}$. The gradient is

$$\nabla\phi(\beta) = M\beta - \beta^{\odot-1}, \tag{16}$$

where we used the symmetry of M for the quadratic term and $\nabla(-\sum_i \log\beta_i) = -\beta^{\odot-1}$ for the logarithmic barrier.

For any fixed point $\beta^\star$ satisfying $M\beta^\star = (\beta^\star)^{\odot-1}$, we have $\nabla\phi(\beta^\star) = 0$. Since $\mathbb{R}^K_{++}$ is open (no boundary constraints), this is equivalent to the first-order stationarity condition of ϕ (Lemma A.1).

For the Nash-style characterization, let $u := M\beta^\star$. The fixed-point equation $M\beta^\star = (\beta^\star)^{\odot-1}$ directly implies $u = (\beta^\star)^{\odot-1}$, yielding $u_i\beta_i^\star = 1$ for all i. This equalization condition characterizes the marginal utility balance in the Nash bargaining framework: since $\nabla_u \sum_i \log u_i = u^{\odot-1}$, the fixed point ensures $u^{\odot-1} = \beta^\star$, meaning marginal utilities in the u-space are proportional to the bargaining coefficients. □

Corollary A.3 (Existence of positive fixed points)**.** *If the interaction matrix $M = V^\top V$ is positive definite, then the fixed-point equation $M\beta = \beta^{\odot-1}$ admits at least one solution $\beta^\star \in \mathbb{R}^K_{++}$.*

Proof. Consider the potential function $\phi(\beta) = \frac{1}{2}\beta^\top M\beta - \sum_{i=1}^K \log\beta_i$ on $\mathbb{R}^K_{++}$.

Since $M \succ 0$, we have:

- The quadratic term $\frac{1}{2}\beta^\top M\beta$ is strictly convex and grows as $\|\beta\|^2$ when $\|\beta\| \to \infty$.
- The logarithmic barrier $-\sum_i \log\beta_i$ tends to $+\infty$ as any $\beta_i \to 0^+$.

Therefore, ϕ is coercive on $\mathbb{R}^K_{++}$ and attains its minimum at some $\beta^\star \in \mathbb{R}^K_{++}$. At this minimizer, the first-order condition $\nabla\phi(\beta^\star) = 0$ yields $M\beta^\star = (\beta^\star)^{\odot-1}$. □

Lemma A.4 (Well-definedness and positivity under safeguarded iteration)**.** *Let $\varepsilon > 0$ and define the safeguarded utility $u(\beta) := \max(M\beta, \varepsilon)$ elementwise. Then the damped iteration*

$$\beta^{(t+1)} = (1-\eta)\beta^{(t)} + \eta\, u(\beta^{(t)})^{\odot-1}, \quad \eta \in (0, 1], \tag{17}$$

is well-defined and preserves positivity: if $\beta^{(0)} \in \mathbb{R}^K_{>0}$, then $\beta^{(t)} \in \mathbb{R}^K_{>0}$ for all $t \geq 0$.

Proof. By construction, $u(\beta) \geq \varepsilon$ elementwise, hence $u(\beta)^{\odot -1} > 0$ elementwise. Since $\beta^{(t+1)}$ is a convex combination of two strictly positive vectors $\beta^{(t)}$ and $u(\beta^{(t)})^{\odot -1}$, positivity follows by induction. □

Remark A.5 (Practical implications). Corollary A.3 guarantees that when M is positive definite (which holds when the influence vectors $\{v_i\}_{i=1}^K$ are linearly independent), the Nash-style bargaining equilibrium exists. Lemma A.4 shows that the safeguarded iteration is well-defined and preserves positivity, avoiding division-by-zero or negative weights during optimization.

Proposition A.6 (Validity of token-level reward decomposition)**.** *Let $\alpha_{s,t} > 0$ for $t = 1, \ldots, L_s$, and define $c_{s,t} = \frac{\alpha_{s,t}}{\sum_{k=1}^{L_s} \alpha_{s,k}}$, while $r_{s,t} = R^{(s)} c_{s,t}$. Then $\{c_{s,t}\}_{t=1}^{L_s}$ lies on the probability simplex, and the induced token-level rewards satisfy $\sum_{t=1}^{L_s} r_{s,t} = R^{(s)}$.*

Proof. Since $\alpha_{s,t} > 0$, we have $c_{s,t} \geq 0$ and $\sum_{t=1}^{L_s} c_{s,t} = \frac{\sum_{t=1}^{L_s} \alpha_{s,t}}{\sum_{k=1}^{L_s} \alpha_{s,k}} = 1$; hence c_s lies on the simplex, and $\sum_{t=1}^{L_s} r_{s,t} = R^{(s)} \sum_{t=1}^{L_s} c_{s,t} = R^{(s)}$. □

Lemma A.7 (Differentiability of Dirichlet-based allocation)**.** *Assume $\alpha_{s,t} = \mathrm{softplus}(f_\phi(h_{s,t}))$ for all t. Then the mapping $\phi \mapsto c_{s,t}$ is continuously differentiable. Moreover, letting $S_s = \sum_{k=1}^{L_s} \alpha_{s,k}$, the partial derivatives satisfy:*

$$\frac{\partial c_{s,t}}{\partial \alpha_{s,k}} = \begin{cases} \frac{S_s - \alpha_{s,t}}{S_s^2}, & k = t, \\ -\frac{\alpha_{s,t}}{S_s^2}, & k \neq t. \end{cases} \tag{18}$$

Proof. Since $\alpha_{s,t} = \mathrm{softplus}(f_\phi(h_{s,t}))$ and $\mathrm{softplus}(\cdot)$ is smooth, each $\alpha_{s,t}$ is continuously differentiable with respect to ϕ. Consequently, $S_s = \sum_{k=1}^{L_s} \alpha_{s,k}$ is continuously differentiable and strictly positive, which implies that $c_{s,t} = \alpha_{s,t}/S_s$ is continuously differentiable by composition.

To compute the partial derivatives, consider $c_{s,t} = \alpha_{s,t}/S_s$ with $S_s = \sum_{j=1}^{L_s} \alpha_{s,j}$. Applying the quotient rule yields:

$$\frac{\partial c_{s,t}}{\partial \alpha_{s,k}} = \frac{\frac{\partial \alpha_{s,t}}{\partial \alpha_{s,k}} S_s - \alpha_{s,t} \frac{\partial S_s}{\partial \alpha_{s,k}}}{S_s^2}. \tag{19}$$

If $k = t$, then $\frac{\partial \alpha_{s,t}}{\partial \alpha_{s,k}} = 1$ and $\frac{\partial S_s}{\partial \alpha_{s,k}} = 1$, giving $\frac{S_s - \alpha_{s,t}}{S_s^2}$. If $k \neq t$, then $\frac{\partial \alpha_{s,t}}{\partial \alpha_{s,k}} = 0$ and $\frac{\partial S_s}{\partial \alpha_{s,k}} = 1$, yielding $-\frac{\alpha_{s,t}}{S_s^2}$. □

Corollary A.8 (Compatibility with policy-gradient methods)**.** *The token-level rewards $\{r_{s,t}\}$ defined in Proposition A.6 constitute a valid reward signal for policy-gradient optimization.*

Proof. By Proposition A.6, the decomposition satisfies $\sum_{t=1}^{L_s} r_{s,t} = R^{(s)}$ with $r_{s,t} = R^{(s)} c_{s,t}$. Thus $\{r_{s,t}\}$ can be used as step-wise rewards in a standard policy-gradient objective by treating each token generation as a decision step and computing returns/advantages from these rewards in the usual way. □

Lemma A.9 (Scale invariance of token allocation)**.** *Let $c_{s,t} = \alpha_{s,t} / \sum_k \alpha_{s,k}$. For any scalar $\lambda > 0$, define $\alpha'_{s,t} = \lambda \alpha_{s,t}$. Then the resulting allocation satisfies $c'_{s,t} = c_{s,t}$ for all t.*

Proof. By definition,

$$c'_{s,t} = \frac{\alpha'_{s,t}}{\sum_{k=1}^{L_s} \alpha'_{s,k}} = \frac{\lambda \alpha_{s,t}}{\sum_{k=1}^{L_s} \lambda \alpha_{s,k}} = \frac{\lambda \alpha_{s,t}}{\lambda \sum_{k=1}^{L_s} \alpha_{s,k}} = \frac{\alpha_{s,t}}{\sum_{k=1}^{L_s} \alpha_{s,k}} = c_{s,t}, \tag{20}$$

which holds for all t. □

Lemma A.10 (Monotonicity of token contributions)**.** *Fix $\{\alpha_{s,k}\}_{k \neq t}$. Then $c_{s,t}$ is strictly increasing in $\alpha_{s,t}$ and strictly decreasing in $\alpha_{s,k}$ for any $k \neq t$.*

Proof. Let $S_s = \sum_{j=1}^{L_s} \alpha_{s,j}$ and recall $c_{s,t} = \alpha_{s,t}/S_s$. Fixing $\{\alpha_{s,k}\}_{k \neq t}$, we have

$$\frac{\partial c_{s,t}}{\partial \alpha_{s,t}} = \frac{S_s - \alpha_{s,t}}{S_s^2} = \frac{\sum_{j \neq t} \alpha_{s,j}}{S_s^2} > 0, \tag{21}$$

since $\alpha_{s,j} > 0$ implies $\sum_{j \neq t} \alpha_{s,j} > 0$. For any $k \neq t$,

$$\frac{\partial c_{s,t}}{\partial \alpha_{s,k}} = -\frac{\alpha_{s,t}}{S_s^2} < 0. \tag{22}$$

Therefore $c_{s,t}$ is strictly increasing in $\alpha_{s,t}$ and strictly decreasing in each $\alpha_{s,k}$ for $k \neq t$. □

B Method Supplement

B.1 Stage-I Bargaining Solver: Well-definedness and Diagnostics

Although $M = V^\top V$ is positive semidefinite by construction, it is not guaranteed to be elementwise nonnegative; consequently, some entries of $M\beta$ may be non-positive for $\beta \in \mathbb{R}^K_{>0}$. To ensure the Nash-style objective $\sum_i \log u_i$ and the reciprocal fixed-point updates are well-defined, we apply a standard positivity safeguard:

$$u = \max(M\beta, \varepsilon) \quad \text{(elementwise)}, \qquad \varepsilon > 0. \tag{23}$$

In our implementation, we additionally monitor the fraction of non-positive entries in $M\beta$ during the iteration; the safeguard is used as a numerical guardrail and is rarely activated in practice. Lemma A.4 (Appendix A) shows that the safeguarded iteration is well-defined and preserves $\beta^{(t)} \in \mathbb{R}^K_{>0}$ for all t.

B.2 Advantage Construction under Reward Decomposition

In the main text, we present a conceptual formulation of sentence-to-token reward decomposition based on semantic allocation. This appendix describes how this decomposition is instantiated in practice under PPO-style RLHF training, where optimization is driven by advantage estimates rather than explicit token-level returns. We emphasize that the underlying PPO objective remains unchanged; the proposed decomposition refines the construction of the advantage signal used for credit assignment, together with additional engineering stabilizations for training robustness.

B.2.1 Sequence-Level Advantage

Given a generated response with sequence-level reward $R^{(\mathrm{seq})}$, we first construct a standard PPO-style sequence-level advantage

$$A^{(\mathrm{seq})} = R^{(\mathrm{seq})} - b, \tag{24}$$

where b denotes a PPO baseline, such as a value-function estimate or a batch-wise baseline. This scalar advantage serves as the global optimization signal and defines the overall scale of policy updates.

B.2.2 Token-Level Advantage under Reward Decomposition

As described in the main text, sentence-level bargaining decomposes the sequence-level reward $R^{(\mathrm{seq})}$ into normalized sentence allocation masses $\{\omega_s\}_{s=1}^S$, where $\omega_s \geq 0$ and $\sum_{s=1}^S \omega_s = 1$. The corresponding sentence-level reward is $R^{(s)} = \omega_s R^{(\mathrm{seq})}$, which preserves the total reward mass: $\sum_{s=1}^S R^{(s)} = R^{(\mathrm{seq})}$. Within each sentence, token-level allocation further produces normalized contribution scores $c_{s,t}$, with $c_{s,t} \geq 0$ and $\sum_{t=1}^{L_s} c_{s,t} = 1$. For practical advantage modulation, we define the token-level allocation weight as

$$w_{s,t} = \omega_s \cdot c_{s,t}, \tag{25}$$

where $w_{s,t} \geq 0$ represents the relative allocation mass assigned to token t in sentence s. The signed optimization direction remains controlled by the sequence-level advantage $A^{(\mathrm{seq})}$.

In practice, these quantities are flattened into a single non-negative per-token weight w_t over all generated tokens. Rather than constructing separate token-level returns, we incorporate this sentence-to-token decomposition by reweighting the sequence-level advantage. A natural token-level advantage is obtained by proportional modulation:

$$A_t^{\mathrm{raw}} = A^{(\mathrm{seq})} \cdot \frac{w_t}{\bar{w}}, \qquad \bar{w} = \frac{1}{T}\sum_{t=1}^{T} w_t, \tag{26}$$

where T denotes the number of generated tokens. Since the weights $\{w_t\}_{t=1}^T$ are non-negative allocation masses, $\bar{w}$ is well-defined for non-empty generated responses. This construction preserves the expected advantage magnitude,

$$\mathbb{E}_t[A_t^{\mathrm{raw}}] = A^{(\mathrm{seq})}, \tag{27}$$

ensuring consistency with the original PPO formulation while enabling fine-grained token-level credit assignment.

B.2.3 Stabilized Token-Level Advantage with Clipping

In practice, unconstrained token-level reweighting may introduce high variance due to extreme allocation scores. To improve optimization stability, we apply a bounded refinement by mixing the token-level modulation with the original sequence-level signal:

$$A_t = A^{(\mathrm{seq})} \cdot \Big[(1-\lambda) + \lambda \cdot \mathrm{clip}\Big(\frac{w_t}{\bar{w}}\Big)\Big], \tag{28}$$

where $\lambda \in [0,1]$ controls the strength of token-level modulation and $\mathrm{clip}(\cdot)$ enforces bounded influence.

This formulation preserves the global advantage scale while preventing excessively fine-grained token-level signals from overriding sentence-level alignment, resulting in more stable PPO updates without modifying the underlying optimization objective.

B.3 Semantic Perturbation Details

For sentence-level influence estimation, we generate local perturbations using a simple deletion-based procedure. Given a response $x = [s_1, \ldots, s_K]$, we perturb one sentence at a time while keeping all other sentences fixed. For each sentence s_i, we generate up to $K_p = 3$ variants by deleting one token from the sentence. When spaCy is available, deletion candidates are restricted to content words (nouns, verbs, adjectives, and adverbs); otherwise, we fall back to whitespace tokenization and delete one token at a time. Variants that remove the entire sentence are discarded.

Each variant $s_i^{(k)}$ is inserted back into the original response to obtain $x^{(i \to k)}$. We then compute the reward difference

$$\Delta R_{i,k} = R(x) - R(x^{(i \to k)}), \tag{29}$$

and the corresponding embedding difference

$$\delta_{i,k} = z_i - z_i^{(k)}. \tag{30}$$

The influence vector v_i is estimated by ridge regression:

$$v_i = \arg\min_v \sum_k \left(\Delta R_{i,k} - \langle v, \delta_{i,k} \rangle\right)^2 + \lambda \|v\|_2^2, \tag{31}$$

with $\lambda = 10^{-3}$. If no valid perturbation is available for a sentence, we set its influence vector to zero. This procedure requires no additional model training or external annotations.

C Discussion

Our results suggest that instability in RLHF is not an implementation artifact, but a structural consequence of driving token-level optimization with low-resolution, sequence-level preference signals. While reward decomposition can substantially improve stability, its benefit does not stem from more precise token-level allocation. Instead, it functions as a form of structured regularization that redistributes noisy preference signals in a way that aligns better with the semantic organization of language.

This perspective echoes recent findings in deep reinforcement learning, such as Hindsight-DICE [43], which argue that credit assignment instability often arises from structural properties of the learning objective rather than from estimation noise alone. From this viewpoint, increasingly fine-grained credit assignment is not inherently beneficial. Reward models provide noisy, indirect estimates of human preference rather than ground-truth supervision, and pushing allocation to very fine granularity can amplify spurious correlations and local artifacts. Our empirical results show that such over-refined advantages may degrade both performance and training stability, challenging the common assumption that finer credit assignment necessarily leads to better RLHF.

This observation motivates a granularity-aware view of reward design, where the goal is not maximal allocation precision but stable and semantically grounded learning dynamics. S2T-RLHF instantiates this principle by allocating reward at the sentence level to preserve high-level semantic coherence, while allowing only limited token-level refinement within sentences. Importantly, S2T should be viewed as one concrete realization of this design philosophy rather than its final form.

Finally, reward decomposition mitigates but does not eliminate RLHF instability. It does not correct reward model misspecification, nor does it guarantee robustness under arbitrary preference shifts or adversarial prompts. Understanding the fundamental stability limits of scalar preference supervision remains an important open question.

C.1 Sources of Instability in RLHF

We analyze the sources of instability in RLHF from a theoretical perspective, focusing on the interaction between sequence-level rewards and token-level policy optimization. Consider an autoregressive language model trained with a PPO-style RLHF objective, which can be written as $\mathcal{L}_{\mathrm{RLHF}}(\theta) = -\mathbb{E}\big[A_t \log \pi_\theta(y_t \mid x, y_{<t})\big]$, which matches the formulation in Eq. (14).

In standard RLHF, supervision is provided via a sequence-level reward. Equivalently, this can be viewed as a terminal reward assigned only at the final generation step:

$$r_t = \begin{cases} 0, & t < T, \\ R(y), & t = T, \end{cases} \tag{32}$$

where $R(y)$ denotes the scalar reward assigned to the complete generated sequence.

In practice, RLHF typically uses a discount factor $\gamma \approx 1$ (or $\gamma = 1$), in which case the token-level advantage can be approximated as

$$A_t \approx \gamma^{T-t} R(y) - V(s_t), \tag{33}$$

where $s_t = (x, y_{<t})$ and $V(s_t)$ denotes the value baseline. This formulation reveals a fundamental structural issue of sequence-level RLHF. Since the same scalar reward $R(y)$ influences the advantage of all tokens along the trajectory, the resulting advantages are highly correlated across time steps and often share the same sign, especially when the value function is imperfect. As a result, token-level policy updates are driven by a global signal that provides little discrimination among individual decisions, leading to high-variance and unstable optimization dynamics.

Directional Alignment of Token-Level Updates.

Remark C.1 (Advantage Alignment under Terminal Rewards). Consider sequence-level RLHF with a terminal reward $R(y)$ and a discount factor $\gamma \approx 1$. When the value function is imperfectly estimated, it is reasonable to assume that $|V(s_t)| \ll |R(y)|$ for a large portion of the trajectory. Under this assumption, the token-level advantage can be approximated as

$$A_t \approx \gamma^{T-t} R(y), \tag{34}$$

which implies that the advantages of most tokens along the trajectory share the same sign. As a consequence, a single policy update induces a coherent update direction across all tokens in the sequence:

$$\Delta\theta \propto \sum_{t=1}^{T} A_t \nabla_\theta \log \pi_\theta(a_t \mid s_t). \tag{35}$$

This observation reveals a structural mismatch between token-level optimization and sequence-level supervision. The policy is updated using a global scalar signal that provides limited discrimination among individual decisions. As a result, the optimization dynamics can be dominated by a small subset of tokens or patterns that exert a disproportionate influence on the reward model, thereby increasing the risk of over-optimization and reward hacking.

Variance Amplification over Long Horizons.

Remark C.2 (Variance Amplification under Long-Horizon Terminal Rewards). Consider sequence-level RLHF with a terminal reward $R(y)$ and discount factor $\gamma \approx 1$. Let the token-level advantage be defined as

$$A_t = \gamma^{T-t} R(y) - V(s_t), \tag{36}$$

where $V(s_t)$ denotes the value baseline. Treating $R(y)$ as a random variable induced by the reward model and $V(s_t)$ as an imperfect baseline estimate, the variance of the advantage admits the decomposition

$$\mathrm{Var}(A_t) = \mathrm{Var}(\gamma^{T-t} R(y) - V(s_t)) \geq \gamma^{2(T-t)} \mathrm{Var}(R(y)) - 2\gamma^{T-t} \mathrm{Cov}(R(y), V(s_t)). \tag{37}$$

In early stages of training, the covariance term $\mathrm{Cov}(R(y), V(s_t))$ is often limited, so the variance of A_t is dominated by the scaled reward variance. Under $\gamma \approx 1$, this implies that the advantages associated with earlier tokens exhibit substantially larger variance. When combined with the intrinsic noise of token-level policy gradients, such variance amplification makes PPO-style updates more sensitive to clipping and KL constraints, thereby increasing the difficulty of stable optimization.

Length-Dependent Gradient Scaling.

Remark C.3 (Length-Dependent Gradient Scaling under Terminal Rewards). Consider sequence-level RLHF with a terminal reward $R(y)$ and discount factor $\gamma \approx 1$. Assume that the token-level advantages A_t along a trajectory share the same sign, as discussed in Remark C.1. The policy gradient can then be written as:

$$\nabla_\theta J(\theta) \propto \sum_{t=1}^{T} A_t \nabla_\theta \log \pi_\theta(a_t \mid s_t), \tag{38}$$

where a_t denotes the generated token and s_t the corresponding autoregressive state.

Let $g_t = \nabla_\theta \log \pi_\theta(a_t \mid s_t)$. Under mild regularity assumptions, such as non-vanishing diagonal contributions and non-negative correlations across time steps, the expected squared norm of the gradient satisfies

$$\mathbb{E}\left[\left\|\sum_{t=1}^{T} A_t g_t\right\|^2\right] \quad \text{grows with } T, \tag{39}$$

at least linearly under these conditions. This implies that longer sequences exert a systematically larger influence on the optimization dynamics when rewards are assigned at the sequence level.

This length-dependent scaling does not necessarily encourage longer outputs per se. However, it introduces an implicit coupling between sequence length and gradient magnitude, making the optimization process sensitive to variations in sequence length and further contributing to instability in RLHF training.

Remarks C.1, C.2, and C.3 together characterize three complementary sources of instability in sequence-level RLHF, arising from directional alignment, variance amplification, and length-dependent gradient scaling, respectively.

C.2 Stabilizing Effects of Reward Decomposition

We now analyze how decomposing a sequence-level reward into localized sentence- and token-level signals alleviates the instability sources identified in Section C.1. Our analysis focuses on structural properties of the resulting reward signals, rather than on specific optimization heuristics. Importantly, the proposed decomposition preserves the total reward mass, ensuring that stabilization arises from improved credit assignment rather than implicit reward rescaling.

Breaking Advantage Alignment.

Remark C.4 (Localized Rewards Reduce Advantage Alignment). Consider a reward decomposition $\{r_t\}_{t=1}^T$ satisfying the conservation property $\sum_{t=1}^T r_t = R(y)$. Under the decomposed reward, the token-level advantage can be written as:

$$A_t^{\text{dec}} = \sum_{k=t}^{T} \gamma^{k-t} r_k - V(s_t). \tag{40}$$

When the reward components r_k are supported on localized tokens or sentence segments, the resulting advantages vary across time steps and no longer share a common global scaling factor.

As a consequence, token-level policy updates are driven by heterogeneous advantage signals rather than a single sequence-level scalar. This breaks the coherent alignment of update directions observed under terminal rewards and enables finer-grained credit assignment across the trajectory.

Reducing Effective Horizon and Variance.

Remark C.5 (Variance Reduction via Localized Reward Support). Under reward decomposition, the token-level advantage depends only on reward components within the future support of $\{r_k\}_{k\geq t}$. If the reward support is localized to a window of bounded length, the effective horizon contributing to A_t^{dec} is substantially shorter than the full sequence length.

As a result, the variance of the decomposed advantage is governed by the variability of local reward components rather than by the variance of a global terminal reward. This reduction in effective horizon mitigates variance amplification for early tokens and improves the signal-to-noise ratio of token-level policy gradients.

Decoupling Gradient Scale from Sequence Length.

Remark C.6 (Length-Decoupled Gradient Scaling under Reward Decomposition). Consider the policy gradient induced by a decomposed reward signal,

$$\nabla_\theta J(\theta) \propto \sum_{t=1}^{T} A_t^{\text{dec}} \nabla_\theta \log \pi_\theta(a_t \mid s_t). \tag{41}$$

When reward components are localized, the magnitude of A_t^{dec} depends primarily on local reward mass rather than on the total sequence length. Consequently, the overall gradient scale becomes less sensitive to variations in sequence length. This decoupling alleviates the implicit coupling between length and optimization pressure induced by terminal rewards, leading to more stable updates across responses of different lengths.

Together, Remarks C.4, C.5, and C.6 show that reward decomposition alleviates instability in RLHF by improving credit locality, reducing variance amplification, and decoupling gradient scale from sequence length, while preserving the total reward mass.

C.3 Limitations

Although S2T-RLHF improves the stability of preference-based RLHF, it has several limitations. First, the method still relies on a learned sequence-level reward model. Reward decomposition can redistribute the scalar reward into more localized learning signals, but it cannot correct reward-model misspecification, systematic bias, or preference errors in the original reward signal. If the reward model assigns high scores to undesirable responses, S2T-RLHF may still propagate this incorrect preference through the decomposed rewards.

Second, the proposed sentence-to-token hierarchy is most suitable for open-ended generation tasks where responses can be naturally segmented into semantically meaningful sentence-level units. For tasks whose correctness depends on tightly coupled symbolic reasoning, such as mathematical proof generation or code execution, sentence-level decomposition may not fully capture step-level dependencies or executable correctness. In such settings, process-level supervision or verifier-based feedback may be more appropriate.

Third, S2T-RLHF introduces additional computational overhead due to perturbation-based sentence influence estimation. Our runtime analysis in Appendix J shows that this overhead is mainly concentrated in Stage I, while the bargaining solver and DTAN refinement are lightweight. Future work may reduce this cost through more efficient perturbation selection, batched reward-model evaluation, caching, or amortized influence estimation.

Finally, while our experiments cover multiple datasets and stress-test settings, they are still limited to a specific policy model, reward-model family, and PPO-based RLHF pipeline. Extending the analysis to larger models, alternative reward models, other policy optimization algorithms, and more diverse task domains remains an important direction for future work.

D End-to-End Theoretical Justification of S2T-RLHF

The analysis in Appendix C focused on the problem side of RLHF instability. In particular, we showed that when a sequence-level terminal reward is propagated to token-level optimization, the resulting learning dynamics are prone to three structural failure modes: global advantage alignment, long-horizon variance amplification, and length-dependent gradient scaling. We now connect this instability analysis to the design of S2T-RLHF itself. Specifically, we show that the two-stage structure of S2T-RLHF is not an arbitrary composition of sentence-level and token-level modules: Stage I induces sentence-level reward localization, while Stage II introduces bounded within-sentence refinement. Together, these two properties provide an end-to-end justification for why the S2T design mitigates the instability mechanisms identified in Appendix C.

Lemma D.1 (Sentence-level localization and conservation)**.** *Let* $\omega_i = \frac{u_i}{\sum_{j=1}^{K} u_j}$ *and* $r_i = \omega_i R(x)$ *denote the Stage-I sentence allocation weights and sentence-level rewards, respectively. Then*

$$\omega_i \geq 0 \quad \forall i, \qquad \sum_{i=1}^{K} \omega_i = 1, \qquad \sum_{i=1}^{K} r_i = R(x). \tag{42}$$

Thus, Stage I redistributes the sequence-level reward over sentence-level semantic units while preserving the total reward mass.

Proof. By construction, $u_i = \max((M\beta)_i, \epsilon) > 0$ for all i, so the normalized coefficients

$$\omega_i = \frac{u_i}{\sum_{j=1}^{K} u_j} \tag{43}$$

are well defined, nonnegative, and satisfy

$$\sum_{i=1}^{K} \omega_i = \frac{\sum_{i=1}^{K} u_i}{\sum_{j=1}^{K} u_j} = 1. \tag{44}$$

Since $r_i = \omega_i R(x)$, it follows immediately that

$$\sum_{i=1}^{K} r_i = \sum_{i=1}^{K} \omega_i R(x) = \Big(\sum_{i=1}^{K} \omega_i\Big) R(x) = R(x). \tag{45}$$

Thus, Stage I preserves the total sequence-level reward while inducing a sentence-level allocation. □

Lemma D.2 (Bounded within-sentence refinement)**.** *For sentence* s*, Stage II defines token-level weights* $c_{s,t} = \alpha_{s,t} / \sum_{k=1}^{L_s} \alpha_{s,k}$ *with* $\alpha_{s,t} > 0$*, and token-level rewards* $r_{s,t} = R^{(s)} c_{s,t}$*. Hence* $\{c_{s,t}\}_{t=1}^{L_s}$ *lies on the probability simplex and preserves the sentence-level reward mass, i.e.,*

$$c_{s,t} \geq 0, \qquad \sum_{t=1}^{L_s} c_{s,t} = 1, \qquad \sum_{t=1}^{L_s} r_{s,t} = R^{(s)}. \tag{46}$$

Moreover, under the practical PPO instantiation $A_t = A^{(\mathrm{seq})}[(1-\lambda) + \lambda \,\mathrm{clip}(w_t/\bar{w})]$*, if* $\mathrm{clip}(w_t/\bar{w}) \in [c_{\min}, c_{\max}]$*, then*

$$|A_t| \leq |A^{(\mathrm{seq})}| \max\big\{\big|(1-\lambda) + \lambda c_{\min}\big|, \big|(1-\lambda) + \lambda c_{\max}\big|\big\}. \tag{47}$$

Therefore, Stage II yields a simplex-constrained and uniformly bounded within-sentence refinement.

Proof. The simplex property follows directly from the definition $c_{s,t} = \alpha_{s,t} / \sum_{k=1}^{L_s} \alpha_{s,k}$ and the positivity assumption $\alpha_{s,t} > 0$. In particular,

$$c_{s,t} \geq 0, \qquad \sum_{t=1}^{L_s} c_{s,t} = \frac{\sum_{t=1}^{L_s} \alpha_{s,t}}{\sum_{k=1}^{L_s} \alpha_{s,k}} = 1. \tag{48}$$

The conservation property is then immediate from $r_{s,t} = R^{(s)} c_{s,t}$:

$$\sum_{t=1}^{L_s} r_{s,t} = R^{(s)} \sum_{t=1}^{L_s} c_{s,t} = R^{(s)}. \tag{49}$$

For the practical PPO instantiation, write

$$m_t = (1-\lambda) + \lambda \,\mathrm{clip}(w_t/\bar{w}), \qquad A_t = A^{(\mathrm{seq})} m_t. \tag{50}$$

Under the clipping assumption $\mathrm{clip}(w_t/\bar{w}) \in [c_{\min}, c_{\max}]$, we have

$$m_t \in \big[(1-\lambda) + \lambda c_{\min},\ (1-\lambda) + \lambda c_{\max}\big]. \tag{51}$$

Therefore,

$$|A_t| = |A^{(\mathrm{seq})}| \cdot |m_t| \le |A^{(\mathrm{seq})}| \max\big\{\big|(1-\lambda) + \lambda c_{\min}\big|, \big|(1-\lambda) + \lambda c_{\max}\big|\big\}. \tag{52}$$

This establishes the claimed boundedness. □

Proposition D.3 (Mitigation of global advantage alignment)**.** *Consider standard sequence-level RLHF with terminal reward propagation, where the token-level advantage is approximated by*

$$A_t^{\mathrm{std}} \approx \gamma^{T-t} R(y) - V(s_t), \tag{53}$$

with $\gamma \approx 1$. When the value baseline is imperfect, the shared sequence-level scalar $R(y)$ dominates A_t^{std} over a substantial portion of the trajectory, inducing sequence-wide alignment in the token-level updates.

Under S2T-RLHF, the sequence-level reward is first decomposed into sentence-level rewards $\{R^{(s)}\}_{s=1}^{S}$, and each sentence-level reward is then refined by simplex-constrained, uniformly bounded within-sentence modulation. Consequently, the resulting update signal is no longer driven by a single sequence-wide scalar, and the residual alignment that would remain under sentence-only uniform allocation is further attenuated. In this sense, S2T-RLHF weakens both cross-sentence alignment and residual within-sentence alignment relative to standard sequence-level RLHF.

Proof. Under standard sequence-level RLHF, the approximation

$$A_t^{\mathrm{std}} \approx \gamma^{T-t} R(y) - V(s_t) \tag{54}$$

implies that, when $\gamma \approx 1$ and the value baseline is insufficiently accurate, a substantial fraction of the token-level advantages is dominated by the same global scalar $R(y)$. The resulting policy update therefore, takes the form

$$\Delta\theta \propto \sum_{t=1}^{T} A_t^{\mathrm{std}} \nabla_\theta \log \pi_\theta(a_t \mid s_t), \tag{55}$$

so that many tokens along the trajectory are driven by advantages with a similar sign and scale. This is the sequence-wide alignment effect identified in Appendix C.

Under S2T-RLHF, Lemma D.1 implies that the global reward $R(y)$ is replaced by a conserved sentence-level decomposition

$$\sum_{s=1}^{S} R^{(s)} = R(y), \tag{56}$$

So the optimization signal is no longer propagated only through a single sequence-wide scalar. Instead, different sentence groups inherit different coarse reward masses, which breaks the update homogeneity across sentences.

Lemma D.2 further shows that, within each sentence, Stage II applies a simplex-constrained and uniformly bounded refinement. Accordingly, token-level modulation within a sentence is neither uniform nor unconstrained: informative local variation can be preserved without allowing arbitrary amplification. This directly mitigates the residual within-sentence alignment that would persist under sentence-only uniform allocation.

Taken together, the two-stage S2T structure weakens the sequence-wide alignment induced by standard terminal-reward propagation and further reduces the residual alignment that remains at the sentence level. □

Proposition D.4 (Mitigation of variance amplification)**.** *Consider standard sequence-level RLHF with token-level advantage*

$$A_t^{\mathrm{std}} = \gamma^{T-t} R(y) - V(s_t), \tag{57}$$

where $R(y)$ is a sequence-level terminal reward. When the covariance term $\mathrm{Cov}(R(y), V(s_t))$ remains limited, the variance of A_t^{std} is dominated by the contribution of the global terminal reward, producing the long-horizon variance amplification identified in Appendix C.

Under S2T-RLHF, the sequence-level reward is first redistributed over sentence-level semantic units and is then refined through bounded within-sentence modulation. Consequently, exposure to a single full-sequence terminal reward is replaced by localized reward support together with uniformly bounded local refinement, which attenuates the variance amplification mechanism of standard sequence-level RLHF.

Proof. For standard sequence-level RLHF, Appendix C gives

$$\mathrm{Var}(A_t^{\mathrm{std}}) = \mathrm{Var}(\gamma^{T-t} R(y) - V(s_t)) \geq \gamma^{2(T-t)} \mathrm{Var}(R(y)) - 2\gamma^{T-t} \mathrm{Cov}(R(y), V(s_t)). \tag{58}$$

When the value baseline is inaccurate and the covariance term remains small, the dominant variance contribution comes from the shared terminal reward $R(y)$. In this regime, especially for early tokens, the advantage inherits variance exposure over the full effective horizon.

Under S2T-RLHF, Lemma D.1 replaces the single sequence-level scalar by a conserved sentence-level decomposition,

$$R(y) = \sum_{s=1}^{S} R^{(s)}, \tag{59}$$

So the reward signal is no longer propagated only through a full-sequence terminal quantity. Instead, it is localized to sentence-conditioned token groups, thereby reducing the effective support over which variance is propagated.

Lemma D.2 further implies that the subsequent token-level refinement is simplex-constrained and uniformly bounded. In particular, the practical PPO modulation satisfies

$$|A_t| \leq C_{\mathrm{clip}} \, |A^{(\mathrm{seq})}|, \tag{60}$$

where $C_{\mathrm{clip}} < \infty$ depends only on λ and the clipping range. Thus, the localized reward signal cannot be arbitrarily re-amplified at the token level.

Taken together, the two-stage S2T structure replaces full-horizon variance exposure under terminal-reward propagation by localized reward support with bounded local modulation, thereby mitigating the variance amplification mechanism identified in Appendix C. □

Proposition D.5 (Mitigation of length-dependent gradient scaling)**.** *Consider the policy gradient under standard sequence-level RLHF,*

$$\nabla_\theta J(\theta) \propto \sum_{t=1}^{T} A_t^{\mathrm{std}} \nabla_\theta \log \pi_\theta(a_t \mid s_t), \tag{61}$$

where A_t^{std} is induced by sequence-level terminal reward propagation. When the token-level advantages are dominated by the same global reward signal, the cumulative gradient magnitude inherits an explicit dependence on the number of tokens receiving that signal, yielding the length-dependent gradient scaling identified in Appendix C.

Under S2T-RLHF, the reward signal is first localized to sentence-level semantic units and is then redistributed within each sentence under simplex and boundedness constraints. Consequently, the update magnitude is no longer governed solely by a single sequence-wide scalar propagated over all T tokens, and the resulting dependence on total sequence length is attenuated relative to standard sequence-level RLHF.

Proof. Under standard sequence-level RLHF, when a substantial fraction of the token-level advantages is dominated by the same sequence-level reward $R(y)$, these advantages tend to share a similar sign along the trajectory. The policy gradient therefore takes the form

$$\nabla_\theta J(\theta) \propto \sum_{t=1}^{T} A_t^{\mathrm{std}} \nabla_\theta \log \pi_\theta(a_t \mid s_t),$$

so that the cumulative update magnitude grows with the number of tokens that inherit the common global signal. This induces the length-dependent coupling between optimization pressure and total sequence length analyzed in Appendix C.

Under S2T-RLHF, Lemma D.1 implies that the global reward is replaced by a conserved sentence-level decomposition. The update is therefore no longer driven exclusively by a single scalar propagated uniformly across the entire trajectory, but by a structured collection of sentence-local reward masses. This weakens the direct dependence of the update magnitude on the total number of generated tokens.

Lemma D.2 further shows that the subsequent token-level redistribution within each sentence is simplex-constrained and uniformly bounded. Accordingly, localized refinement cannot assign arbitrarily large token weights that would recreate pathological scaling at the token level after sentence-level localization.

Taken together, the two-stage S2T structure replaces sequence-wide scalar propagation by localized and bounded reward redistribution, thereby attenuating the length-dependent gradient scaling mechanism identified in Appendix C. □

Corollary D.6 (End-to-end justification of the two-stage design)**.** *The two-stage structure of S2T-RLHF admits an end-to-end theoretical justification relative to the instability mechanisms identified in Appendix C. In particular, Stage I induces sentence-level reward localization, while Stage II provides bounded within-sentence refinement. Their combination jointly attenuates global advantage alignment, long-horizon variance amplification, and length-dependent gradient scaling.*

Proof. Proposition D.3 shows that the two-stage structure weakens global advantage alignment by replacing sequence-wide reward propagation with sentence-localized and bounded token-level modulation. Proposition D.4 shows that the same structure attenuates variance amplification by combining localized reward support with uniformly bounded local refinement. Proposition D.5 further shows that this combination weakens the dependence of gradient magnitude on total sequence length.

Taken together, these results identify complementary roles for the two stages. Stage I localizes the reward signal at the sentence level, thereby removing the sequence-wide mismatch induced by terminal-reward propagation. Stage II refines this localized signal within each sentence under explicit simplex and boundedness constraints, preventing the local redistribution from degenerating into unconstrained token-level amplification. The stated corollary follows. □

E Implementation Details

E.1 Experiments Details

E.1.1 Training Hyperparameters

Table 5: Training hyperparameters.

PARAMETER	VALUE
PROMPT BATCH SIZE	8
KL COEFFICIENT	0.15
TOP-p	0.9
LEARNING RATE	3×10^{-6}
PPO CLIP RANGE	0.1
PPO EPOCHS	2
BATCH SIZE	16
MINI-BATCH SIZE	2
ENTROPY COEFFICIENT	0.005
VALUE LOSS COEFFICIENT	0.2
GAE λ	0.95

E.1.2 Inter-Sentence Bargaining Solver

Table 6: Hyperparameters of the inter-sentence bargaining solver.

PARAMETER	VALUE
MAXIMUM ITERATIONS	100
CONVERGENCE TOLERANCE	10^{-6}
DAMPING FACTOR	0.5
NUMERICAL STABILITY ϵ	10^{-8}

E.1.3 DTAN Architecture

Table 7: DTAN hyperparameters.

PARAMETER	VALUE
INPUT DIMENSION	3584
HIDDEN DIMENSION	1024
ACTIVATION	`GELU`

E.1.4 LoRA Configuration

Table 8: LoRA hyperparameters used for policy fine-tuning.

PARAMETER	VALUE
LORA RANK (r)	8
LORA SCALING (α)	16
LORA DROPOUT	0.05

F Quantitative Stability Metrics Across Seeds

To substantiate qualitative descriptors such as "smoother" and "less oscillatory," we quantify training stability under the default configuration used in the main experiments and aggregate results over three random seeds.

Logged trajectories. We record training dynamics at T logging checkpoints using the same logging frequency and evaluation protocol for all methods and seeds. At each checkpoint, we evaluate the current policy on a fixed set of prompts and log: (i) KL divergence to the reference policy, (ii) the PPO policy objective, (iii) dispersion of reward-model scores within the evaluation batch (reward variance), and (iv) policy entropy. All quantities are computed identically to the main training logs for every method and seed. In particular, entropy is computed consistently by token-averaging over generated responses and then averaging over prompts.

Metrics. From the logged trajectories, we report four scalar stability summaries: (i) **KL-Mean**, the time-average KL, measuring overall policy drift; (ii) **Obj-Δ**, the mean absolute step-to-step change of the PPO objective, quantifying objective oscillations; (iii) **RVar-Mean**, the time-average reward variance within the evaluation batch, capturing dispersion of the preference signal; and (iv) **Ent-Var**, the temporal variance of policy entropy, measuring volatility in exploration behavior. Entropy is included mainly as a diagnostic to rule out degenerate collapse; its mean level need not be monotonic with stability, whereas **Ent-Var** directly reflects fluctuations. Lower values indicate more stable and better-controlled optimization dynamics. We report each metric as the mean across three seeds, with 95

Results. Tables 9 and 10 summarize stability statistics over three seeds in two training phases. Across both windows, sentence-structured credit assignment substantially reduces optimization oscillations: **Obj-Δ** drops by an order of magnitude compared to vanilla RLHF, and S2T-RLHF achieves the smallest **Obj-Δ** among all variants. This supports our thesis that aligning reward decomposition with sentence-level semantics stabilizes PPO optimization, and that bounded within-sentence refinement further improves control.

For policy drift, token-only refinement exhibits the largest **KL-Mean** in the late window, consistent with amplified drift under overly fine-grained credit assignment. S2T-RLHF reduces this drift relative to token-only refinement while maintaining the most stable objective dynamics. Entropy volatility remains comparable across methods, with S2T-RLHF showing reduced **Ent-Var** in the early adaptation phase.

Finally, **RVar-Mean** increases for decomposition-based variants in the late phase. Importantly, reward-score dispersion reflects evaluator variability rather than update volatility; with bounded refinement, S2T-RLHF can still yield smoother and better-controlled optimization even when reward scores are more dispersed.

Table 9: Stability metrics in the early training window (0–1000 steps). Lower is better (↓).

Method	RVar-Mean ↓	KL-Mean ↓	Obj-Δ ↓	Ent-Var ↓
RLHF	0.1071 ± 0.3397	0.00405 ± 0.00248	0.01134 ± 0.03368	3.1156 ± 1.7894
Sentence	0.2704 ± 0.3914	0.00372 ± 0.00126	0.00132 ± 0.02201	3.0894 ± 1.8529
Token	0.2314 ± 0.3832	0.00455 ± 0.00177	0.00622 ± 0.02341	3.0754 ± 1.8073
S2T-RLHF	0.2477 ± 0.3657	0.00418 ± 0.00141	0.00101 ± 0.01299	2.8496 ± 1.7495

Table 10: Stability metrics in the late training window (5000–6000 steps). Lower is better (↓).

Method	RVar-Mean ↓	KL-Mean ↓	Obj-Δ ↓	Ent-Var ↓
RLHF	0.2413 ± 0.3319	0.00523 ± 0.00139	0.01106 ± 0.02562	2.9479 ± 1.7983
Sentence	0.4045 ± 0.4640	0.00619 ± 0.00190	0.00870 ± 0.02125	2.7376 ± 1.7785
Token	0.4266 ± 0.4147	0.00902 ± 0.00264	0.00952 ± 0.01699	2.7834 ± 1.7134
S2T-RLHF	0.4549 ± 0.4239	0.00837 ± 0.00289	0.00719 ± 0.01361	2.8489 ± 1.9072

G Early-Stage Training Dynamics

This section presents a focused analysis of early-stage training dynamics in RLHF. Empirically, we observe that training instability predominantly arises during the initial phase of optimization, where noise in credit assignment can be rapidly amplified before policy distributions stabilize. Accordingly, we provide a zoomed-in analysis of the first 500 optimization steps for different credit assignment strategies, complementing the full training trajectories reported in the main paper.

Beyond temporal localization, we further investigate whether the observed stability gains of S2T-RLHF depend on the learnable capacity of DTAN. In the standard S2T-RLHF setup, DTAN is initialized randomly and optimized jointly with the policy during RLHF training; we denote this variant as **S2T-noPT**. To assess the role of pretraining, we also consider a variant in which DTAN is first pretrained using the same data and reward model, and then held fixed during subsequent RLHF optimization; we denote this variant as **S2T-PT**.

Across settings, pretraining DTAN does not improve training stability and, in some cases, slightly increases early-stage reward variability. These results indicate that the stability benefits of S2T-RLHF arise from the structural regularization imposed by bounded, non-adaptive token-level refinement, rather than from increased representational capacity.

To improve interpretability of highly noisy early-stage trajectories, we visualize training curves using a smoothed estimate of the underlying dynamics. Specifically, for each metric we apply a centered sliding-window average over optimization steps, which acts as a low-pass filter to suppress high-frequency stochastic fluctuations induced by minibatch sampling and reward noise, while preserving long-term training trends. We additionally overlay the raw unsmoothed trajectories with reduced opacity to ensure faithful representation of the original data. This visualization strategy follows standard practice in reinforcement learning for analyzing optimization stability and convergence behavior, enabling clearer comparison of structural differences across credit assignment mechanisms.

G.1 Early-stage dynamics under learning rate 5×10^{-6}

Figure 3 summarizes the first 500 PPO updates under a conservative learning rate (5×10^{-6}), where stochasticity is moderate and behavioral differences primarily reflect the structure of credit assignment. We compare standard RLHF, sentence-only and token-only decompositions, and S2T-RLHF with/without DTAN pretraining.

Entropy. All methods keep policy entropy within a narrow band, indicating no early mode collapse in this regime. However, token-level decomposition exhibits visibly larger entropy fluctuations, consistent with higher sensitivity to local reward noise. Sentence-level decomposition and both S2T variants yield smoother entropy trajectories, suggesting more stable early exploration.

KL to the reference policy. Token-level decomposition shows a clearer upward drift in KL even at this small step size, implying more aggressive deviation from the reference distribution. In contrast, standard RLHF and sentence-level decomposition maintain tighter KL control, and S2T-RLHF matches or improves upon this behavior, indicating that structured sentence-to-token allocation can regulate early policy drift.

Objective, reward variance, and critic learning. Token-level decomposition attains higher short-term objective values, but with substantial variability, consistent with noisy (high-variance) advantage estimates. S2T-RLHF exhibits a smoother and more monotonic objective evolution and lower reward variance, indicating reduced amplification of stochastic learning signals. Value-function loss decreases across all methods, but S2T-RLHF shows the most consistent early convergence. DTAN pretraining does not improve stability and can slightly increase variability.

Takeaway. Under 5×10^{-6}, S2T-RLHF achieves stable early optimization while avoiding the larger variance and stronger drift induced by fine-grained token-level credit assignment. The limited benefit of DTAN pretraining supports the view that stability gains mainly arise from structural regularization rather than added capacity.

G.2 Early-stage dynamics under learning rate 8×10^{-6}

Figure 4 reports the first 500 updates under a moderately larger learning rate (8×10^{-6}), which acts as a stronger stress test where optimization noise is more easily amplified.

Entropy. Entropy remains bounded across methods but becomes visibly more volatile than at 5×10^{-6}. Token-level decomposition shows the largest fluctuations, while S2T-RLHF maintains comparatively smooth entropy evolution, indicating more robust exploration behavior as step sizes increase.

KL to the reference policy. Differences are most pronounced in KL. Token-level decomposition exhibits rapid and sustained KL growth, reflecting accelerated policy drift under higher learning rates. Standard RLHF and sentence-level decomposition retain tighter control, and S2T-RLHF achieves consistently lower KL growth throughout early training. The pretrained DTAN variant tends to drift slightly more than the non-pretrained one.

Objective, reward variance, and critic learning. Token-level decomposition can reach higher objective values later in this window, but these gains coincide with large oscillations and higher reward variance. In contrast, S2T-RLHF improves the objective more smoothly and maintains more consistent reward statistics. Value-function loss decreases for all methods, but S2T-RLHF shows a more monotonic and stable convergence pattern.

Takeaway. At 8×10^{-6}, instability from token-level credit assignment becomes more apparent, primarily through variance amplification and excessive drift. S2T-RLHF remains stable and better controlled, again suggesting that structured allocation, not additional capacity, drives the robustness gains.

G.3 Early-stage dynamics under learning rate 1×10^{-5}

Figure 5 reports early-stage dynamics under an aggressive learning rate (1×10^{-5}), near the boundary where RLHF training often becomes unstable.

Entropy. Entropy stays bounded on average but exhibits markedly larger fluctuations than in lower learning-rate settings. Token-level decomposition remains the most volatile, indicating unstable exploration under aggressive updates. Sentence-level decomposition partially moderates this effect, while S2T-RLHF shows smoother entropy evolution.

KL to the reference policy. Token-level decomposition displays the fastest and most persistent KL growth, consistent with excessive policy drift. S2T-RLHF substantially mitigates this drift relative to token-only methods, while standard RLHF and sentence-level decomposition typically retain the tightest KL control. As in other regimes, DTAN pretraining does not reduce drift and often slightly worsens it.

Objective, reward variance, and critic learning. Token-level decomposition can produce larger objective values, but with frequent oscillations and elevated reward variance, consistent with high-variance advantages under large step sizes. S2T-RLHF yields a more gradual and stable objective improvement, alongside lower reward variance and smoother critic convergence.

Takeaway. Under 1×10^{-5}, fine-grained token-level credit assignment leads to pronounced instability, which is most clearly reflected in increased policy drift and amplified reward variability, whereas S2T-RLHF remains comparatively robust, supporting the need for structured and bounded credit assignment in aggressive optimization regimes.

G.4 Early-stage dynamics under reduced KL constraint

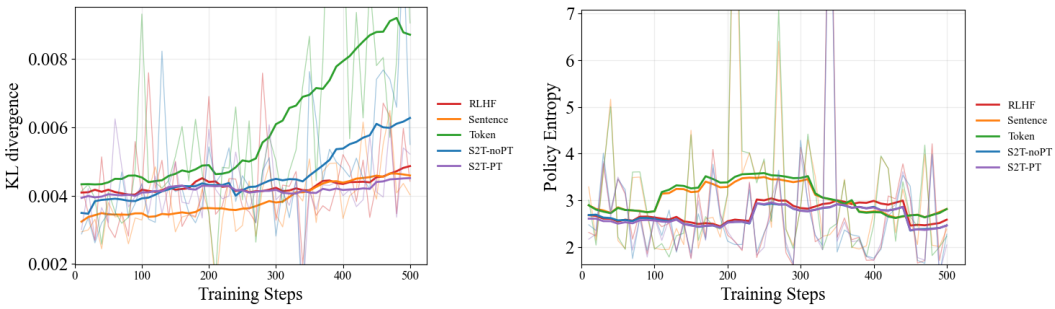
Figure 6 repeats the early-stage analysis under a stricter KL constraint (0.1), which directly limits deviation from the reference policy and complements the learning-rate-based stress tests.

As expected, the tighter KL constraint bounds policy drift for all methods, compressing differences in KL trajectories relative to looser settings. Nevertheless, stability differences remain.

Entropy and objective. Entropy variance is reduced overall, but token-level decomposition continues to exhibit larger fluctuations than sentence-level and S2T-RLHF variants, indicating persistent sensitivity to stochastic reward signals even under stronger regularization. With tighter KL control, objective gains are also more conservative: token-level decomposition still shows slightly higher short-term objectives, but with noticeable variability, whereas S2T-RLHF remains smoother and more consistent.

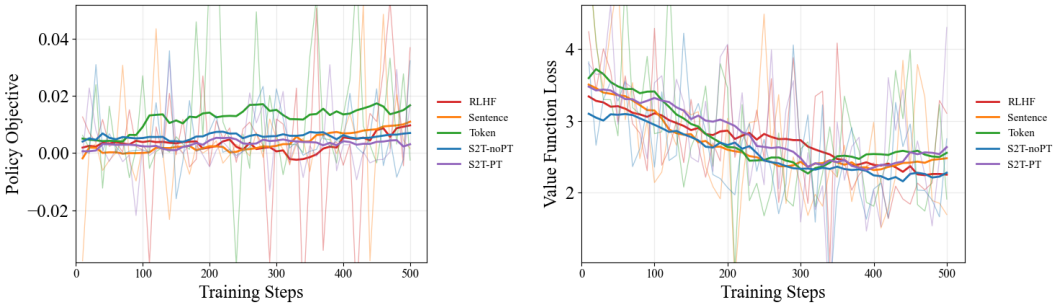
Reward variance and critic learning. Reward variance decreases across methods under stronger KL regularization, yet token-level decomposition still exhibits relatively higher fluctuations. S2T-RLHF maintains lower and more stable reward variance and shows smoother value-loss convergence, suggesting more reliable critic learning under bounded token-level refinement.

Takeaway. A stricter KL constraint stabilizes RLHF for all methods but does not eliminate structural differences across credit assignment strategies. S2T-RLHF remains more stable than token-level decomposition even when policy updates are tightly constrained, implying that the gains cannot be explained solely by KL regularization and instead arise from structured reward allocation.



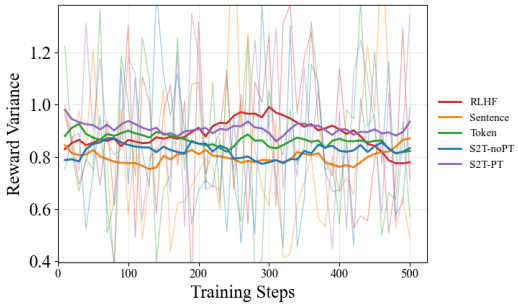
(a) KL Divergence

(b) Policy Entropy



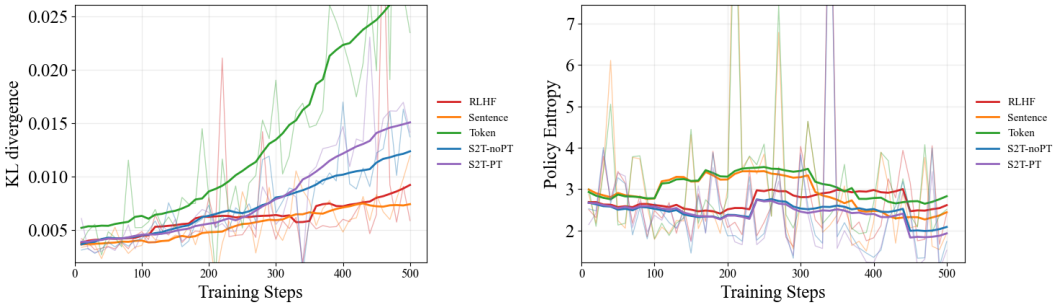
(c) Policy Objective

(d) Value Function Loss



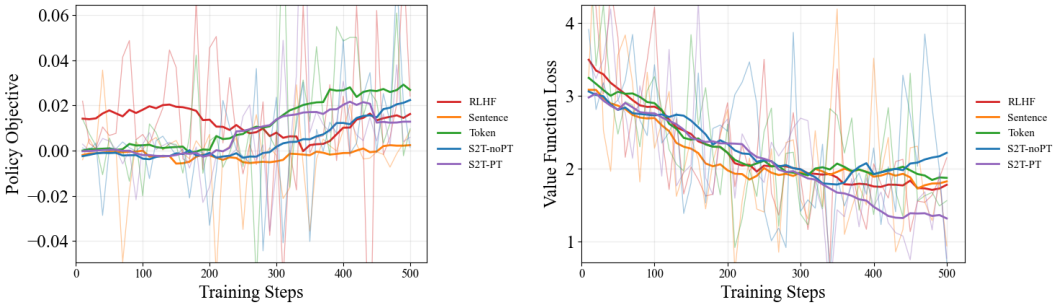
(e) Reward Variance

Figure 3: Early-stage training dynamics under learning rate 5×10^{-6}.



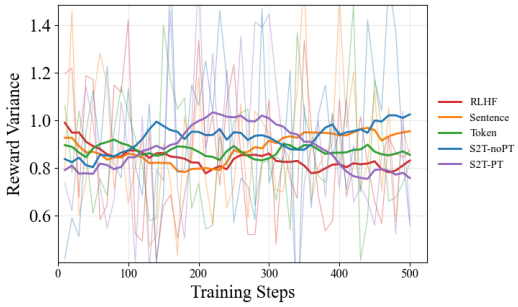
(a) KL Divergence

(b) Policy Entropy



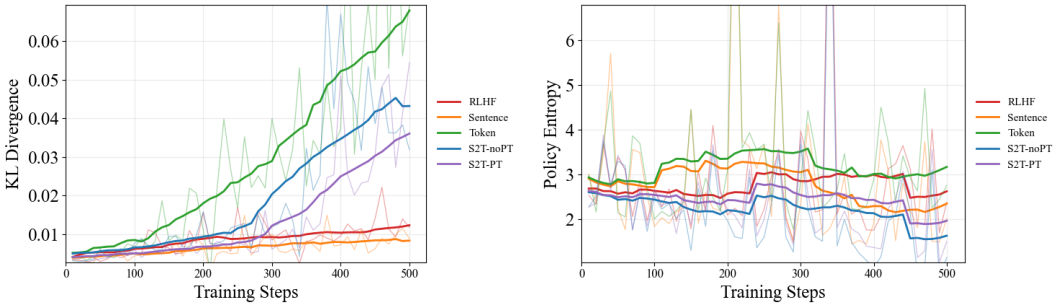
(c) Policy Objective

(d) Value Function Loss



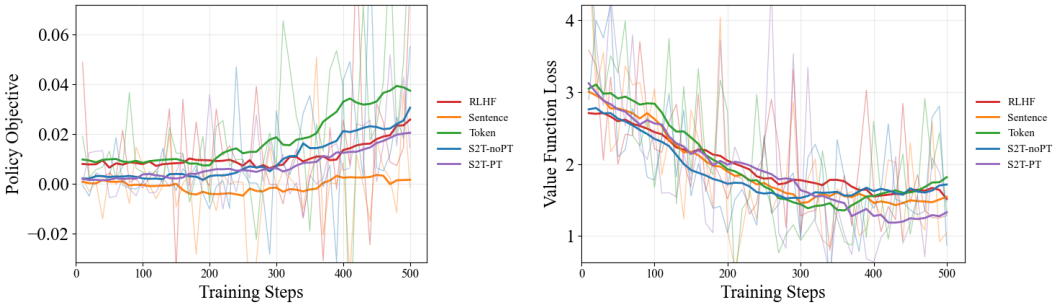
(e) Reward Variance

Figure 4: Early-stage training dynamics under learning rate 8×10^{-6}.



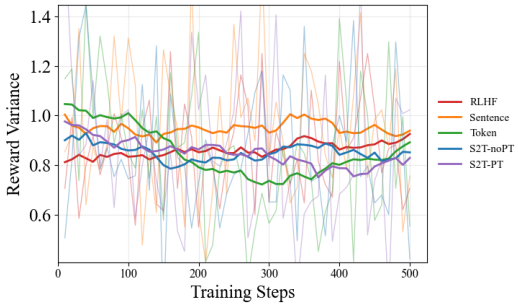
(a) KL Divergence

(b) Policy Entropy



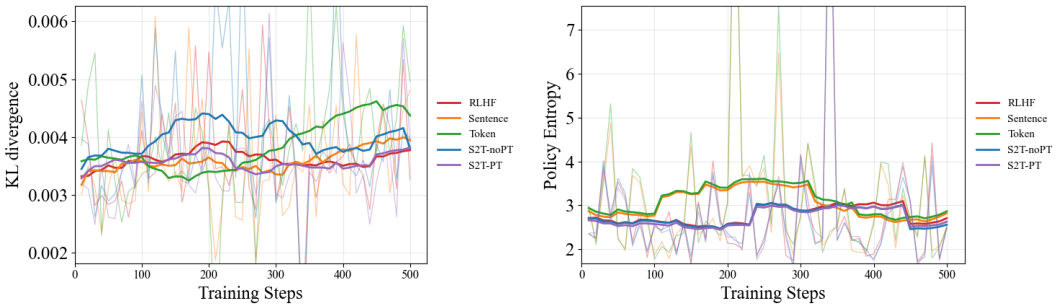
(c) Policy Objective

(d) Value Function Loss



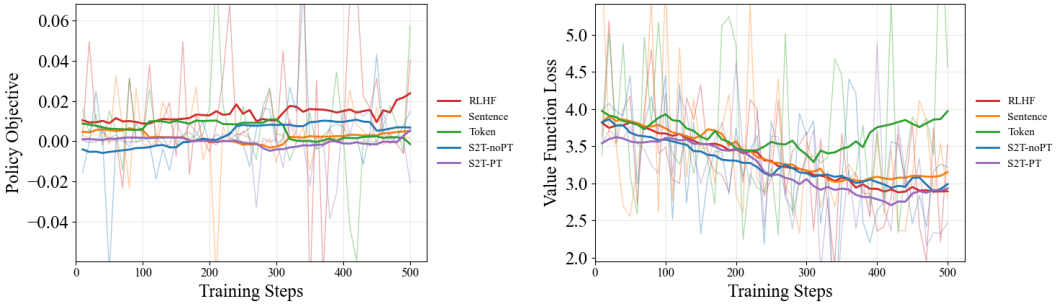
(e) Reward Variance

Figure 5: Early-stage training dynamics under learning rate 1×10^{-5}.



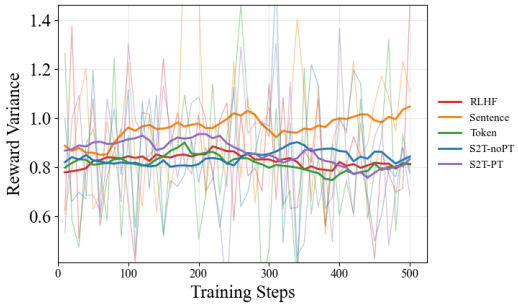
(a) KL Divergence

(b) Policy Entropy



(c) Policy Objective

(d) Value Function Loss



(e) Reward Variance

Figure 6: Early-stage training dynamics under KL constraint 0.1.

H Supplementary Experimental Results

H.1 Cross-Dataset Preference Evaluation

Datasets. We evaluate the robustness and generalization of S2T-RLHF across a diverse collection of alignment and safety benchmarks. As an in-distribution reference, we use **HH-RLHF** [44], which consists of human preference comparisons targeting helpfulness and harmlessness. To assess out-of-distribution generalization under adversarial settings, we include **AdvBench** [45], a benchmark designed to probe jailbreak-style and instruction-following vulnerabilities. We further consider toxicity-oriented datasets, including **RealToxicityPrompts** [46] and **ToxiGen** [47], which focus on harmful, biased, and offensive language generation. In addition, we evaluate on **SafeRLHF**, a curated safety preference dataset covering refusal behavior, harmlessness, and policy compliance. Together, these datasets span a wide range of prompt distributions, risk profiles, and alignment objectives, enabling a comprehensive evaluation of cross-dataset generalization beyond the training domain. Unless otherwise specified, models are trained on HH-RLHF and evaluated zero-shot on the remaining datasets.

Reward Models. We decouple the reward models used for training and evaluation to assess robustness across reward-model choices. During RLHF optimization, all methods are trained using **RM-Gemma-2B** [42], which provides the scalar preference signal for policy updates under a standard PPO-based RLHF pipeline. For evaluation, we adopt a stronger and independently trained reward model, **RM-Gemma-7B** [42], to measure preference alignment and generalization. RM-Gemma-7B follows a standard instruction-tuned reward modeling pipeline and is widely used in recent RLHF studies. Unless otherwise specified, all reported preference alignment results are evaluated using RM-Gemma-7B.

Table 11: Pairwise preference win rates against RLHF under RM-Gemma-7B across datasets.

Dataset	vs. RLHF		vs. ABC		vs. SCAR	
	Win↑	Lose	Win↑	Lose	Win↑	Lose
HH-RLHF	**51.1**	39.5	**48.7**	43.8	**45.8**	44.1
Toxicity	**48.4**	45.4	46.9	**48.9**	**51.6**	44.4
AdvBench	**52.5**	43.4	**52.5**	41.4	44.4	**49.6**
ToxiGen	**46.6**	43.4	42.4	**46.4**	**47.7**	47.5
SafeRLHF	**53.5**	45.3	**48.7**	48.1	**50.1**	45.9

Table 11 reports pairwise preference outcomes evaluated under RM-Gemma-7B across multiple datasets. Overall, S2T-RLHF attains a win advantage over RLHF on all listed datasets, suggesting that introducing stability-oriented decomposition does not harm preference alignment under the same evaluator and can yield modest improvements. Against stronger baselines, namely ABC and SCAR, the results are more dataset-dependent. S2T-RLHF wins on several datasets, for example Toxicity and SafeRLHF against SCAR, while trailing in a small number of pairings, for example Toxicity against ABC and AdvBench against SCAR. Taken together, these cross-dataset outcomes indicate that S2T-RLHF is generally effective for preference alignment, and the remaining failures are concentrated in a few difficult dataset and baseline combinations rather than reflecting a systematic degradation.

H.2 Model and Reward-Model Robustness

To assess whether the preference-alignment conclusions depend on a specific policy backbone or training reward model, we conduct additional experiments with two model–reward-model configurations. In the Gemma-based setting, we use Gemma-2-2B as the policy model and RM-Gemma-7B as the training reward model. In the Qwen-based setting, we use Qwen3-4B as the policy model and Skywork-Reward-V2-Qwen3-4B as the training reward model. All other training and evaluation settings, including prompts, compared model pairs, decoding configuration, and win–tie–lose computation, are kept the same as in the main preference evaluation. Preference alignment is evaluated using the same GPT-4o-mini LLM-as-a-Judge protocol as in main experiments. Notably, the experiments with Qwen3-4B were conducted on Huawei Ascend 910B NPUs.

Table 12 reports the pairwise preference results. Across both model–reward-model configurations, S2T-RLHF maintains a clear win advantage over standard RLHF and ABC, and also achieves higher win rates than loss rates against SCAR. These results suggest that the preference-alignment conclusion is not tied to a single policy backbone or training reward model.

Table 12: Model and reward-model robustness results evaluated using GPT-4o-mini as the LLM-as-a-Judge.

Training setup	Method	Win (%) ↑	Lose (%) ↓	Tie (%)
Gemma-2-2B / RM-Gemma-7B	vs. RLHF	58.1	28.4	13.5
Gemma-2-2B / RM-Gemma-7B	vs. ABC	50.9	31.8	17.3
Gemma-2-2B / RM-Gemma-7B	vs. SCAR	45.6	35.3	19.1
Qwen3-4B / Skywork-Reward-V2-Qwen3-4B	vs. RLHF	56.8	29.8	13.4
Qwen3-4B / Skywork-Reward-V2-Qwen3-4B	vs. ABC	50.9	31.5	17.6
Qwen3-4B / Skywork-Reward-V2-Qwen3-4B	vs. SCAR	46.3	35.6	18.1

Table 12 reports the pairwise preference results. Across both model–reward-model configurations, S2T-RLHF maintains a clear win advantage over standard RLHF and ABC, and also achieves higher win rates than loss rates against SCAR. In the Gemma-based setting, S2T-RLHF improves over RLHF by a large margin, with 58.1% wins versus 28.4% losses, and remains favorable against ABC and SCAR. The Qwen-based setting shows a similar pattern, with 56.8% wins against RLHF and consistent win advantages over both ABC and SCAR. The margins against SCAR are smaller than those against RLHF and ABC, which is expected since SCAR is a strong fine-grained credit-assignment baseline. Overall, these results suggest that the preference-alignment conclusion is not tied to a single policy backbone or training reward model.

H.3 Mechanism-Swap Ablations

The stage-level ablations in the main text examine whether sentence-level allocation, token-level refinement, and their combination are useful. However, they do not isolate whether the gains come only from the hierarchical sentence-to-token structure or also from the specific allocation mechanisms used in each stage. To address this question, we conduct mechanism-swap ablations while keeping the overall two-stage structure fixed.

For Stage I, we replace the Nash-based sentence allocator with two simpler alternatives: uniform sentence allocation and length-proportional allocation. Uniform allocation assigns the same reward share to each sentence, while length-proportional allocation distributes reward according to sentence length. For Stage II, we replace DTAN with uniform within-sentence allocation and a linear+softmax token allocator. These variants preserve the same sentence-to-token decomposition pipeline, but change the mechanism used to allocate reward mass within each stage.

Table 13 reports the Stage-I mechanism-swap results. The Nash allocator achieves the best overall trade-off, with the lowest KL-Mean and Obj-Δ, as well as the highest win rate against RLHF. Compared with uniform and length-proportional allocation, Nash allocation provides smoother optimization dynamics while preserving stronger preference alignment.

Table 13: Stage-I mechanism-swap ablations. Lower KL-Mean and Obj-Δ indicate more stable optimization. Preference results are evaluated against RLHF.

Stage-I allocator	KL-Mean ↓	Obj-Δ ↓	Win (%) ↑	Tie (%)	Lose (%) ↓
Uniform	0.00455	0.00622	51.4	9.8	38.8
Length-prop.	0.00443	0.00229	51.1	7.6	41.3
Nash (ours)	0.00418	0.00101	53.3	8.4	38.3

Table 14 reports the Stage-II mechanism-swap results. DTAN achieves the strongest overall configuration, with the lowest KL-Mean and Obj-Δ, and the highest win rate against RLHF. Uniform within-sentence allocation is stable but less effective in preference alignment, while the linear+softmax allocator improves win rate but yields larger objective oscillations than DTAN. This suggests that

normalized token-level refinement is helpful, but the specific DTAN parameterization provides a better stability–alignment trade-off.

Table 14: Stage-II mechanism-swap ablations. Lower KL-Mean and Obj-Δ indicate more stable optimization. Preference results are evaluated against RLHF.

Stage-II allocator	KL-Mean $\downarrow$	Obj-Δ $\downarrow$	Win (%) $\uparrow$	Tie (%)	Lose (%) $\downarrow$
Uniform	0.00572	0.00132	48.5	12.4	39.1
Linear+softmax	0.00452	0.00192	52.9	8.6	38.5
DTAN (ours)	0.00418	0.00101	53.3	8.4	38.3

Overall, these mechanism-swap ablations show that the gains of S2T-RLHF are not explained by hierarchical decomposition alone. The Nash-based Stage-I allocator improves sentence-level reward localization, while DTAN provides bounded and adaptive within-sentence refinement. Their combination yields the best stability–alignment trade-off among the tested configurations.

H.4 Training-time Reward Monitoring.

Figure 7 visualizes the reward trajectories recorded during training, as measured by the optimization reward model (RM-Gemma-2B). This diagnostic is included to illustrate how different credit assignment strategies interact with the training reward signal over time, rather than to serve as a performance metric. We observe that token-level methods (e.g., ABC and SCAR) tend to exhibit larger short-term reward fluctuations, suggesting more aggressive and localized policy updates. In contrast, S2T-RLHF maintains comparatively smoother reward trajectories throughout training, indicating more structured and controlled utilization of the reward signal. Importantly, these differences in training-time reward dynamics do not necessarily translate into large gaps in final preference alignment scores (Table 11), highlighting the limited interpretability of raw reward values during RLHF optimization and motivating the need for stability-aware evaluation.

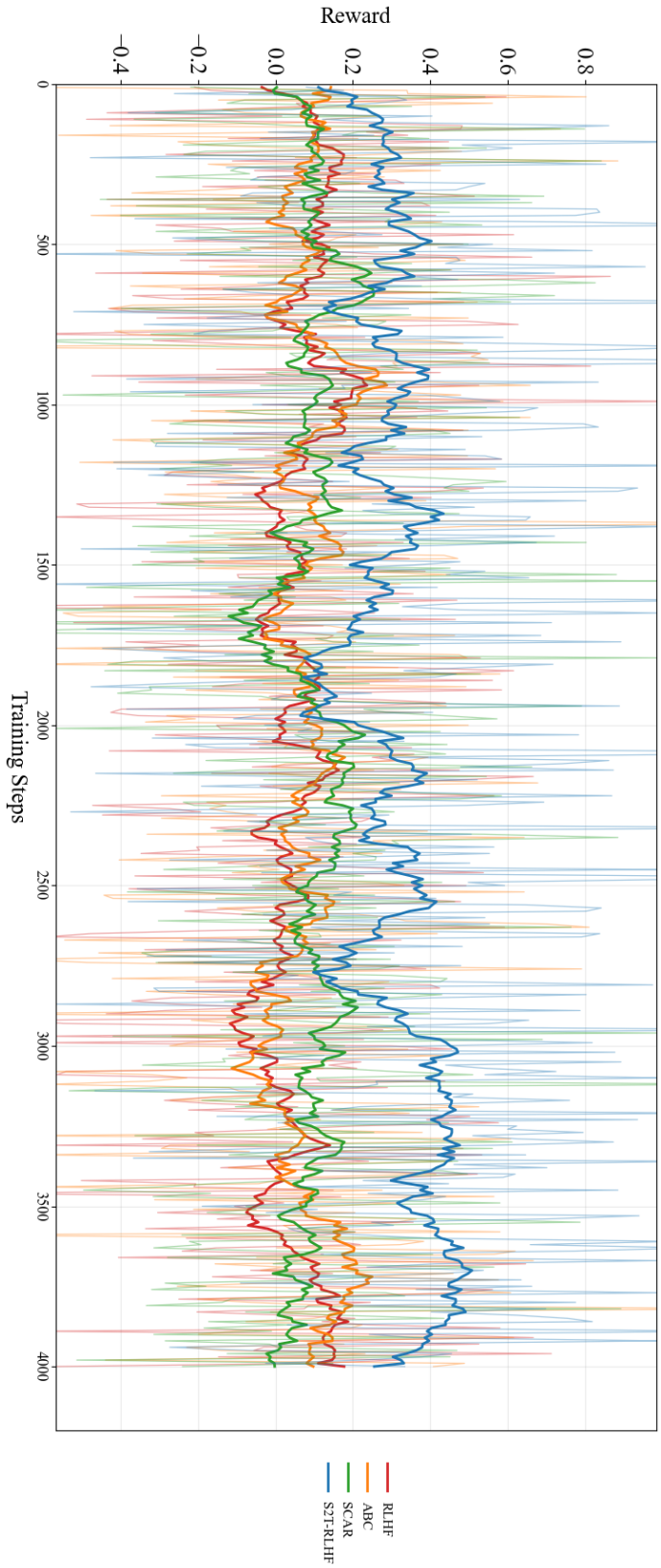
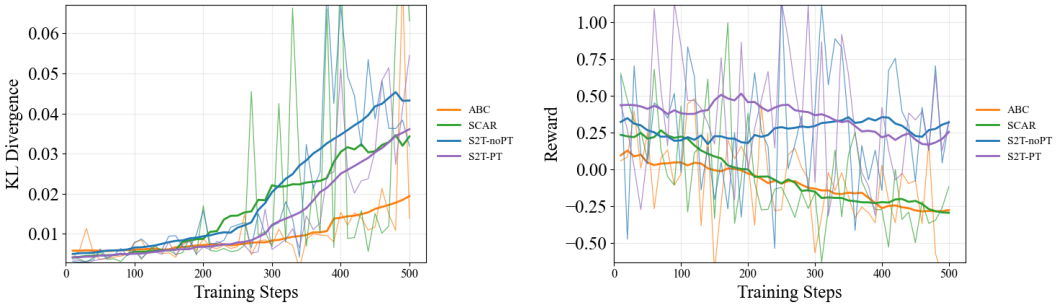


Figure 7: Training-time reward trajectories under the optimization reward model (RM-Gemma-2B).

I A Case Study on Noise Amplification under Fine-Grained Reward Decomposition.

Figure 8 presents an illustrative training run under a more aggressive learning rate of 1×10^{-5}. While all methods exhibit increasing KL divergence, indicating faster policy drift from the reference model, their reward dynamics differ markedly. For ABC and SCAR, the KL increase is accompanied by pronounced oscillations and a sustained degradation of the training reward that crosses below zero and remains negative for a substantial portion of training. This pattern, characterized by continued policy drift together with worsening reward, is consistent with noise amplification under highly localized credit signals rather than improved preference optimization. Moreover, ABC exhibits a clear regime change in the later stage, around $\sim$450 steps, where the KL curve becomes highly volatile with large spikes, indicating unstable and noise-driven updates.

In contrast, both variants of S2T-RLHF maintain smoother and more monotonic KL growth under the same learning rate, without entering the negative-reward regime observed for ABC and SCAR. Although the policy still departs from the reference, the reward signal remains comparatively stable throughout training, suggesting that sentence-level allocation with bounded token-level refinement mitigates noise amplification during aggressive optimization. Overall, this case study supports the view that overly fine-grained reward decomposition can destabilize RLHF training by concentrating noisy preference signals, especially under larger learning rates.



(a) KL Divergence

(b) Reward Mean

Figure 8: Illustrative training dynamics under fine-grained reward decomposition with a larger learning rate (1×10^{-5}).

J Computational Resources and Runtime Analysis

Hardware and training budget. All experiments were conducted on a single A100-SXM4-80GB GPU. Unless otherwise specified, all methods were trained under the same PPO optimization budget, prompt batch size, mini-batch size, KL coefficient, LoRA configuration, and evaluation protocol. We use Gemma-2-9B as the policy model and RM-Gemma-2B as the reward model.

Per-step runtime comparison. Table 15 reports the average per-step runtime of different credit assignment methods. S2T-RLHF incurs additional computation compared with standard RLHF and ABC, but remains more efficient than SCAR. Specifically, S2T-RLHF requires 21.18 seconds per PPO step, corresponding to an 81.5% overhead over RLHF, whereas SCAR requires 26.98 seconds per step, corresponding to a 131.2% overhead. This result shows that S2T-RLHF introduces nontrivial but moderate computational overhead relative to Shapley-based token decomposition.

Table 15: Per-step runtime comparison. All methods are evaluated under the same hardware and PPO training budget.

Method	Time / step (s) ↓	Overhead vs. RLHF (%) ↓
RLHF	11.67	0.0
ABC	13.24	13.5
SCAR	26.98	131.2
S2T-RLHF	21.18	81.5

Component-level runtime breakdown. To identify the source of the additional cost, Table 16 decomposes the runtime of S2T-RLHF into the RLHF backbone, perturbation-based sentence influence estimation, sentence-level bargaining, and DTAN refinement. The breakdown shows that the overhead is dominated by sentence influence estimation, which requires 8.81 seconds per step. In contrast, sentence-level bargaining takes only 0.12 seconds per step, and DTAN refinement takes 0.61 seconds per step. Thus, the main computational bottleneck of S2T-RLHF is reward-model evaluation under sentence perturbations, rather than the bargaining solver or the token-level refinement module.

Table 16: Component-level runtime breakdown of S2T-RLHF. Minor discrepancies are due to rounding.

Component	Time / step (s) ↓	Added over RLHF (s)	Share of total step time (%)
Standard RLHF backbone	11.67	–	55.1
Sentence influence	8.81	8.81	41.6
Sentence bargaining	0.12	0.12	0.6
DTAN refinement	0.61	0.61	2.9
S2T-RLHF total	21.18	9.54	100.0

Discussion. These results indicate that the computational cost of S2T-RLHF is primarily associated with Stage I, especially perturbation-based sentence influence estimation. This cost is expected because Stage I requires additional reward-model evaluations to estimate how sentence-level semantic perturbations affect the sequence-level reward. By contrast, the Nash-style bargaining solver and DTAN-based token refinement are lightweight. Future implementations may further reduce the overhead by caching reward-model evaluations, reducing the number of perturbations, batching sentence perturbations more aggressively, or replacing perturbation-based influence estimation with a learned amortized estimator.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction state the paper's main contribution as a granularity-aware sentence-to-token reward decomposition method for improving RLHF training stability. These claims are supported by the method description, theoretical discussion, ablations, training-dynamics analysis, and cross-dataset evaluations, while the paper also limits its scope to preference-based RLHF with sequence-level reward models.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The paper discusses limitations in Appendix C.3, including reliance on a learned sequence-level reward model, limited suitability for tasks requiring tightly coupled symbolic or process-level correctness, computational overhead from perturbation-based sentence influence estimation, and evaluation scope limited to specific policy/reward models and a PPO-based RLHF pipeline. The paper also reports computational cost and runtime breakdown in Appendix J.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best

judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: The main assumptions and idealized settings are stated in the theoretical appendices. Appendix A presents the stationarity, existence, positivity, and validity results for the Nash-style sentence allocation and Dirichlet token allocation, while Appendix B describes the practical safeguards used in implementation. Appendix C states the assumptions behind the instability analysis of sequence-level RLHF, and Appendix D connects these assumptions to the end-to-end two-stage design. The theoretical claims are therefore presented with their corresponding assumptions, proof scope, and practical safeguards.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The paper specifies the models, datasets, training pipeline, evaluation protocols, computational setting, and main hyperparameters in Section 5 and Appendix E. Additional stability metrics, seed aggregation, and cross-dataset evaluation details are provided in Appendices F–I.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example

(a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
(b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
(c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: We provide an anonymized code repository with reproduction instructions, and all datasets used in the experiments are publicly available.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: The paper specifies the models, datasets, PPO training setup, evaluation protocols, hyperparameters, optimizer-related settings, and computational resources in Section 5 and Appendix E.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: We report seed-aggregated stability metrics with mean and standard deviation over three random seeds in Appendix F.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
   - If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We report the main computational setting, including the use of a single A100-SXM4-80GB GPU, and provide training configurations and hyperparameters

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research conforms to the NeurIPS Code of Ethics and uses anonymized submission materials, public datasets, and standard evaluation protocols.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.

- If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: The paper discusses potential positive impacts for more stable and robust RLHF, as well as limitations and risks from reward-model misspecification, preference shifts, and adversarial prompts.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: This work does not release a new pretrained language model, scraped dataset, or other high-risk asset; the released materials are limited to code for reproducing the proposed method.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We cite the original sources of the models, datasets, and baseline methods used in the experiments, and respect their publicly stated licenses and terms of use.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The released code is documented in the anonymized repository, including instructions for reproducing the proposed method.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: This work does not involve new crowdsourcing or human-subject studies; no participant instructions, screenshots, or compensation details are applicable.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.

- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: This work does not involve new crowdsourcing or human-subject studies; experiments use existing public datasets and automatic evaluation protocols.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [Yes]

    Justification: The paper describes the use of LLMs in the core method and experiments, including the policy model, reward models, and LLM-as-a-Judge evaluation protocol.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.