TWiNS: An Implicit Noise Suppression Approach for Multi-Turn Dialogue Fine-Tuning

Anonymous ACL submission

Abstract

Fine-tuning multi-turn dialogue systems requires high-quality data but tends to degrade with low-quality or out-of-distribution (OOD) samples. Early errors accumulate, amplifying inconsistencies and degrading response quality. However, existing methods separate data quality control from fine-tuning, overlooking turn-level dependencies and cumulative noise, which hinders end-to-end optimization in multi-turn settings. To bridge this gap, we propose TWiNS (Turn-weighted Welford-based implicit Noise Suppression), an end-to-end adaptive fine-tuning method that implicitly pinpoints noisy samples and then suppresses their gradient contributions over the course of model tuning on the fly, mitigating error accumulation and preserving coherence in multi-turn dialogues. Specifically, turn-aware weighting maintains contextual coherence, while Welford's online algorithm adjusts sample weights without pre-filtering. Experiments show that TWiNS ensures stable optimization across multi-turn dialogues, enhancing performance on individual and mixed-quality datasets while mitigating degradation. By suppressing noise without explicit filtering, it adapts to evolving data distributions with zero pre-filtering overhead, establishing a new paradigm for end-to-end data-quality optimization in multi-turn dialogue systems.

1 Introduction

Multi-turn dialogue systems are crucial in both task-oriented (Xu et al., 2024) and open-domain conversational agents (Lu et al., 2023a; Sun et al., 2024), enabling natural and efficient human-computer interactions. Fine-tuning these systems is challenging due to their reliance on multi-turn dialogue datasets (Bian et al., 2023; Zhao et al., 2024b; Contributors, 2023), which include both manually annotated and synthetic data (OpenAI, 2023). Although they dominate due to scalability (Zhang et al., 2023; Maheshwary et al., 2024), their inconsistent quality frequently disrupts training and optimization. In multi-turn dialogue, this variability compounds over turns, leading to incoherent responses, error propagation, and context drift. Furthermore, misalignment with evaluation metrics on multi-turn benchmarks (Zheng et al., 2023; Kwan et al., 2024a), often leads to optimization instability, jeopardizing response coherence and overall performance (Wu et al., 2023; Chen et al., 2023; Li et al., 2024a; Zhou et al., 2024).

Traditional approaches improve data quality and training performance through pre-filtering-based data selection before fine-tuning (Wang et al., 2024), typically removing noisy, redundant, and incomplete samples. However, these methods rely on predefined criteria (Cao et al., 2023), which often fail to account for the complex contextual dependencies in multi-turn dialogues, leading to excessive data filtering. Additionally, some approaches (Wu et al., 2022) introduce noise for robustness but lack systematic optimization, further impacting model performance.

To address these challenges, this paper proposes a novel Turn-weighted Welford-based implicit Noise Suppression (TWiNS) mechanism, an adaptive fine-tuning mechanism that filters noisy "junk" data during fine-tuning. Unlike static filtering methods, TWiNS dynamically adjusts the weight of each training instance using Welford's online statistical algorithm, ensuring that loss distribution updates reflect data quality. This approach not only suppresses noisy samples but also preserves the contextual dependencies crucial for multi-turn dialogues, preventing early-turn errors from propagating to later responses. Moreover, TWiNS integrates turn-specific hierarchical positioning, enabling fine-grained adjustments tailored to multi-turn dialogue structures.

Our experiments demonstrate the effectiveness of this approach. The proposed method consis-

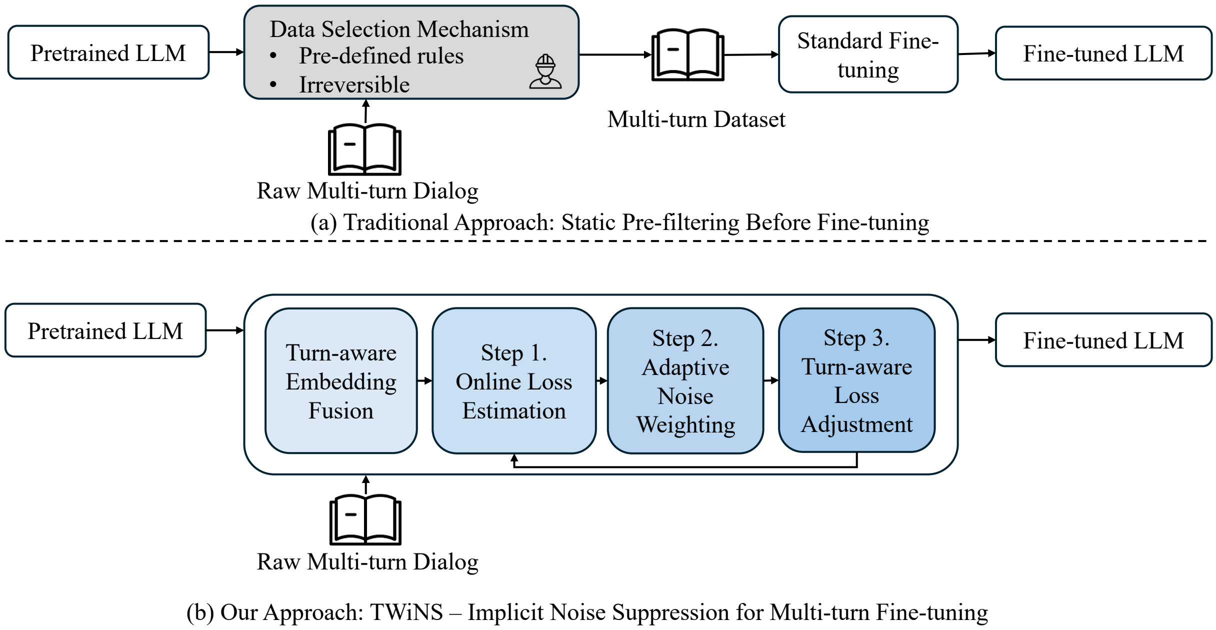


Figure 1: Comparison of the traditional fine-tuning approach and our TWiNS method. (a) The traditional approach applies static pre-filtering before standard fine-tuning. (b) TWiNS integrates implicit noise suppression into fine-tuning through online loss estimation, adaptive noise weighting, and turn-aware loss adjustment. Other standard fine-tuning steps are omitted for clarity.

tently outperforms existing multi-turn dialogue training techniques on widely recognized benchmarks, including MT-Bench, MT-Bench-Ext, and in-domain tests. Notably, it mitigates overfitting and prevents unstable optimization often observed in supervised fine-tuning. Furthermore, when fine-tuning on mixed-quality datasets, TWiNS maintains or improves performance across domain-specific tasks, with no performance penalty observed on individual datasets. These results confirm the method's robustness, scalability, and adaptability to diverse multi-turn dialogue settings.

Our key contributions include:

- We introduce TWiNS, an end-to-end fine-tuning framework that suppresses noisy samples through dynamic loss regulation, eliminating manual filtering while preserving multi-turn coherence.
- Our approach enables scalable training on mixed-quality datasets by leveraging adaptive online statistics, integrating large-scale heterogeneous data without performance loss.
- TWiNS surpasses existing methods on MT-Bench, MT-Bench-Ext, and in-domain benchmarks, consistently enhancing performance across datasets of varying quality.

2 Related Work

2.1 Multi-turn Dialogue Fine-tuning

Recent advancements in LLM fine-tuning (Hu et al., 2023, 2021; Dettmers et al., 2024) have substantially enhanced performance, leading to the development of a new family of models (Liu et al., 2024a; Zhao et al., 2024a; Meng et al., 2024).

While existing methods have addressed multi-turn conversations to a limited extent, performance tends to degrade when fine-tuning on multi-turn dialogue datasets (Sun et al., 2024). In the context of combining LLMs with multi-turn dialogue systems, recent studies have introduced improvements in methods (Sun et al., 2024; Shani et al., 2024), focusing on context-aware preferences and reinforcement learning, and in data (Maheshwary et al., 2024; Ou et al., 2024), emphasizing separate data enrichment. These two areas of enhancement, however, have not been integrated elegantly, and issues like expensive data curation, weak generalization, and inconsistent quality continue to pose challenges.

2.2 Data Selection in LLM Finetuning

Although the scale of data is crucial in LLM fine-tuning, selecting fewer high-quality data points can lead to better performance than using the entire dataset (Wu et al., 2023; Chen et al., 2023), highlighting the significance of data selection in LLM fine-tuning.

In terms of data quality assessment methods (Wang et al., 2024), data selection schemes can be categorized into three types: (1) GPT-based scoring, which involves designing detailed prompts and evaluation criteria and using ChatGPT as the scoring tool (Chen et al., 2023; Lu et al., 2023b; Xu et al., 2023; Liu et al., 2024b; Du et al., 2023); (2) trained model-based scoring, where an LLM is trained with a predefined policy to score each instance and set a threshold (Li et al., 2023, 2024b; Anonymous, 2024); and (3) indicator-based methods, which estimate dataset quality through inference loss (Cao et al., 2023) or by defining indicators based on conversation features (Wei et al., 2023)

Although these methods similarly stress the importance of refined data, they often produce inexplicable results, suffer from limited applicability and randomness, or demand prohibitively high training costs, leading to a loss of feasibility in both training and generalization as models evolve. Additionally, prior approaches perform data selection before and independently of the training process, failing to capture and leverage end-to-end feedback during training—a key focus of our work.

3 Methodology

In multi-turn fine-tuning, given dialogue history $H = \{(u_1, r_1), \ldots, (u_{n-1}, r_{n-1})\}$ and the current user utterance u_n, the goal is to generate a coherent response r_n by optimizing model parameters θ to maximize the conditional log-likelihood:

$$\theta^* = \arg\max_{\theta} \log p_\theta(r_n \mid H, u_n) \quad (1)$$

To systematically structure our methodology, we first assess dataset quality before fine-tuning and then introduce the TWiNS method to evaluate model performance across datasets of varying quality.

3.1 Evaluation on Datasets

There are multiple public, open source, and high-quality multi-turn conversation datasets, which are generated by both humans and LLMs, especially ChatGPT. Table 4 in section B of Appendix details public datasets in this work including ShareGPT (RyokoAI, 2023), WildChat (Zhao et al., 2024b), OpenAssistant (Köpf et al., 2024), ChatAlpaca (Bian et al., 2023), MTLingual (Maheshwary et al., 2024), and UltraChat (Ding et al., 2023) with their features.

Dataset	Con.	Qu.	ID	Fr.	Overall
ChatAlpaca	8.34	**9.49**	0.0286	**9.48**	High
MTLingual	**8.54**	9.37	0.0263	9.14	High
UltraChat	8.46	9.06	0.0233	9.41	High
WildChat	7.80	8.78	0.0196	8.90	Normal
ShareGPT	8.10	8.69	0.0174	8.82	Normal
OpenAssistant	7.54	7.57	**0.0292**	8.21	Low

Table 1: Dataset Evaluation Results. Con.: Connection, Qu.: Quality, ID: Information Density, Fr.: Friendliness.

Existing benchmarks on LLM evaluation (Zheng et al., 2023; Kwan et al., 2024b; Radziwill and Benton, 2017) and labels in OpenAssistant consider the relevance, helpfulness, and accuracy when grading the responses of LLMs, and state the importance of the awareness of ethics, safety, and privacy. Moreover, a work on dialogue (Dethlefs et al., 2016) addresses the significance of information density in human dialogues. Thus, this work proposes an evaluation benchmark on multi-turn dialogue datasets in four independent aspects: Connection, Information Density, Quality, and Friendliness.

Connection: The assistant's final response should incorporate relevant information from prior conversations without introducing any unrelated or redundant details.

Quality: Each response should fulfill the specific request of the corresponding turn, while ensuring content accuracy and maintaining high language quality.

Information Density (ID): Treat the conversation as a whole, calculating the total number of words N and the number of information units I. The information density is then defined as $ID = I/N$.

Friendliness: Human requests should be made with attention to manner, while the assistant's responses should prioritize security and politeness. The conversation as a whole should maintain a respectful tone.

The evaluation is done by ChatGPT, which is a common labeling tool in evaluation works (Zheng et al., 2023; Kwan et al., 2024b; Bai et al., 2024), and the prompts and data processing are detailed in section A of Appendix. For the evaluation on each aspect, one hundred complete conversations are independently and randomly sampled, and the evaluation on each single conversation and dataset is also independent. The score of each aspect of a dataset is defined as the average score of the chosen conversations in this aspect. The evaluation result is detailed in Table 1. Based on the scores of the four aspects, the overall quality of the 6 datasets are divided to high (ChatAlpaca, MTLingual, UltraChat), normal (WildChat, shareGPT), and low (OpenAssistant). OpenAssistant is built from real human conversations, each labeled for quality with the goal of subsequent reinforcement learning. Because it deliberately preserves both high- and low-quality responses, the dataset yields a relatively lower overall score in our evaluation.

3.2 TWiNS

We present TWiNS (Turn-Weighted Welford-Based Implicit Noise Suppression), an adaptive loss regulation method that dynamically adjusts sample importance through Welford's online variance estimation. This technique stabilizes fine-tuning by suppressing noise and modulating loss contributions according to turn depth.

3.2.1 Turn-aware Embedding Fusion

To effectively capture the structure of multi-turn conversations, we integrate turn-aware embeddings to every utterance in the dialogue. These turn IDs are incorporated as additional features to help the model distinguish the roles and importance of different conversation turns.

For turn t, we compute a learnable turn embedding $e_{\text{turn}}(t)$, which is element-wise combined with the token and positional embeddings before being passed to the transformer's first layer. In the additive fusion method, the turn embedding is combined with the token embedding $\mathbf{e}_{\text{token}}$ and the positional embedding $\mathbf{e}_{\text{pos}}$ by element-wise addition:

$$\mathbf{e}_{\text{final}} = \mathbf{e}_{\text{token}} + \mathbf{e}_{\text{pos}} + \mathbf{e}_{\text{turn}} \tag{2}$$

The fused representation $\mathbf{e}_{\text{final}}$ is then propagated through the transformer, allowing the model to jointly encode turn structure, token identity, and positional information.

3.2.2 Online Loss Estimation

To adaptively regulate loss, we estimate its mean μ and variance σ^2 online using Welford's algorithm, enabling real-time updates without extra data passes. Let N denote the current iteration or update count, and let ℓ represents the mini-batch loss at each step. We iteratively update:

$$\mu \leftarrow \mu + \frac{\ell - \mu}{N} \tag{3}$$

$$\sigma^2 \leftarrow \frac{M_2}{N-1} \tag{4}$$

where M_2 is a running accumulation of the squared deviations from μ, facilitating variance estimation.

These statistics are computed independently for each conversational turn, segmenting samples into discrete buckets based on turn ID. Earlier turns ($t \leq 4$) typically correspond to simpler queries, whereas later turns ($t > 4$) tend to involve more intricate and contextually complex interactions. This segmentation separates distinct turn complexities, enhancing the model's adaptability to different dialogue structures.

To mitigate short-term fluctuations in loss values, we optionally apply Exponential Moving Averages (EMA) for additional smoothing. Let μ_{EMA} be the exponentially smoothed loss estimate, and $\theta \in (0,1)$ the smoothing factor. We iteratively update:

$$\mu_{\text{EMA}} \leftarrow (1-\theta) \cdot \mu_{\text{EMA}} + \theta \cdot \ell \tag{5}$$

By leveraging these statistics, TWiNS enhances the stability of loss estimation while effectively detecting outliers within each turn-specific bucket, aligning with the inherent multi-turn nature of SFT tasks.

Adaptive Noise Weighting. Using online loss statistics, we dynamically reweight noisy samples based on their deviation from expected loss. The deviation score d is computed as:

$$d = \frac{|\ell - \mu|}{\sigma + \epsilon} \tag{6}$$

$$w_{\text{junk}} = \frac{1}{1 + \alpha \cdot d} \tag{7}$$

where ϵ is a small constant for numerical stability, and α is a positive scaling factor that determines how aggressively outliers are downweighted. To avoid extreme weight shifts, w_{junk} is clipped within $[w_{\min}, w_{\max}]$. Samples with deviation scores outside this range are deemed noise and have their contribution adaptively reduced, minimizing disruptive effects while preserving valuable signals for improved robustness in multi-turn fine-tuning.

3.2.3 Turn-aware Loss Adjustment

To adaptively adjust the training loss in multi-turn SFT, we introduce three weighting factors: the turn weight w_{turn}, the dynamic junk weight w_{dynamic}, and the assistant weight $w_{\text{assistant}}$. The adjusted loss is computed as:

$$\ell_{\text{adjusted}} = \ell \cdot \max\Big(0.3, \min\big(w_{\text{turn}} \cdot w_{\text{dynamic}} \cdot w_{\text{assistant}}, 2.0\big)\Big) \quad (8)$$

The turn weight w_{turn} accounts for dialogue complexity, ensuring that later turns, which are often more complex, contribute to the overall loss:

$$w_{\text{turn}} = 1.0 + \beta \cdot \text{avg\_turn} \quad (9)$$

where β is a tunable coefficient (e.g., $\beta = 0.05$).

To handle noisy samples, we apply the dynamic junk weight w_{dynamic}, which adjusts the impact of samples based on their deviation from the expected loss distribution:

$$w_{\text{dynamic}} = 1.0 - \gamma \cdot (1.0 - w_{\text{junk}}) \quad (10)$$

Additionally, the assistant weight $w_{\text{assistant}}$ regulates the contribution of tokens generated by the assistant role:

$$w_{\text{assistant}} = 1.0 + \lambda \cdot \text{assistant\_ratio} \quad (11)$$

To prevent excessive weighting, $w_{\text{assistant}}$ is clamped within the range $[1.0, 1.2]$. These three factors collectively enhance training stability by emphasizing turn complexity, mitigating noisy samples, and balancing role-specific contributions.

4 Experiments

4.1 Experimental Settings

Parameter All fine-tuning experiments were conducted based on **Llama-3.2-3B Instruct** (Face and AI, 2023) model. The model was fine-tuned for 3 epochs on datasets of varying quality. Each device processed a batch size of 4, with a gradient accumulation step of 4, resulting in an effective batch size of 64. The Adam optimizer was employed, with the hyperparameter β_2 set to 0.95. A cosine decay learning rate schedule was applied, starting at an initial learning rate of 1×10^{-5} and incorporating a warm-up ratio of 0.01.

All training and evaluation procedures were performed in FP16 precision on four NVIDIA GPUs. To reduce memory consumption, gradient checkpointing and Low-Rank Adaptation (LoRA) were enabled during training. Model performance was periodically assessed using a held-out validation set of 400 examples.

To enhance the robustness of the training process, a warm-up strategy was implemented during the initial phase of training. This involved using 640 high-quality dialogue samples to initialize baseline mean and variance parameters. As training progressed, the filtering weight for anomalous data was gradually increased to ensure smooth and stable model optimization.

Evaluation We first conducted in-domain evaluations on six datasets: ShareGPT, WildChat, OpenAssistant, ChatAlpaca, MTLingual, and UltraChat. From each dataset, 100 multi-turn dialogues were sampled for assessment. Additionally, we performed evaluations on the general-purpose multi-turn dialogue benchmarks, MT-Bench and the extended MT-Bench (MT-Bench_Ext), to verify the consistency of our training approach across dataset-specific and general benchmarks. Finally, we employed evaluation prompts from the "LLM-as-a-Judger" study and conducted all evaluations using the GPT4-08-06 model.

Mix Dataset To validate the effectiveness of our approach, we selected ChatAlpaca, ShareGPT, and OpenAssistant as representatives of high-, normal-, and low-quality datasets, respectively. From each dataset, 20K samples were extracted and mixed in different combinations: high and normal quality, high and low quality, and high, normal, and low quality. These experiments were designed to assess the performance of our method in handling datasets with varying distributions during training.

4.2 Baselines

We evaluate our method against four typical methods in multi-turn dialogue study:
(1) **Baseline**: the original instructed model without dedicated multi-turn dialogue fine-tuning.
(2) **Vicuna-tuning**: a widely adopted dialogue adaptation framework built upon LLaMA, distinguished by its LoRA fine-tuning strategy on multi-turn conversational data (Chiang et al., 2023).
(3) **Baize**: a parameter-efficient approach that exclusively updates linear layers through self-chat generation (Chiang et al., 2023).

Lv.	Dataset	In_domain test			MT-Bench			MT-Bench-Ext		
		Base.	V.T.	TWiNS	Base.	V.T.	TWiNS	Base.	V.T.	TWiNS
H	M2Lin.(en)	**7.10**	7.09 (-0.14%)	7.06 (-0.56%)	7.13	**7.21** (+1.12%)	7.16 (+0.42%)	6.64	6.65 (+0.15%)	**6.71** (+1.05%)
	ChatAlpaca	8.20	7.99 (-2.56%)	**8.26** (+0.73%)	7.13	6.97 (-2.24%)	**7.29** (+2.24%)	6.64	5.99 (-9.79%)	**6.76** (+1.81%)
	UltraChat	7.90	7.56 (-4.30%)	**8.01** (+1.39%)	7.13	6.68 (-6.31%)	**7.32** (+2.66%)	6.64	6.22 (-6.33%)	**6.76** (+1.81%)
N	ShareGPT	6.55	6.09 (-7.02%)	**6.95** (+6.11%)	7.13	6.08 (-14.73%)	**7.83** (+9.82%)	6.64	5.80 (-12.65%)	**6.83** (+2.86%)
	WildChat	6.80	6.47 (-4.85%)	**6.86** (+0.88%)	7.13	7.14 (+0.14%)	**7.21** (+1.12%)	6.64	6.74 (+1.51%)	**6.72** (+1.20%)
L	OpenAss.	7.64	7.20 (-5.76%)	**7.67** (+0.39%)	7.13	6.20 (-13.07%)	**7.26** (+1.83%)	6.64	5.48 (-17.47%)	**6.83** (+2.86%)

Table 2: Comparison of our method, non-trained Baseline, and Vicuna-Tuning on LLaMA 3.2-3B: multi-turn dialogue performance (GPT-4 scores) across high-, normal-, and low-quality datasets. Each cell shows absolute scores plus relative improvement/decline (%) vs. Baseline in parentheses. *Lv. = Level, H, N, L = High, Normal, Low, M2Lin.(en) = M2Lingual (en), OpenAss. = OpenAssistant, Base. = Baseline, V.T. = Vicuna-Tuning.

(4) **ChatGLM3**: implements multi-turn dialogue fine-tuning by updating only the loss of roles other than *user* and *system* (GLM et al., 2024).

All methods share identical LoRA configurations (rank=128, alpha=16, dropout=0.3) and data partitions: 20,000 training samples with 400 validation and 100 test instances. Experiments are conducted with fixed random seeds (seed=42) and multi-turn dialogue performance quantified by the MT-Bench (Zheng et al., 2023).

4.3 Main Results

4.3.1 Does TWiNS address negative optimization in multi-turn dialogues?

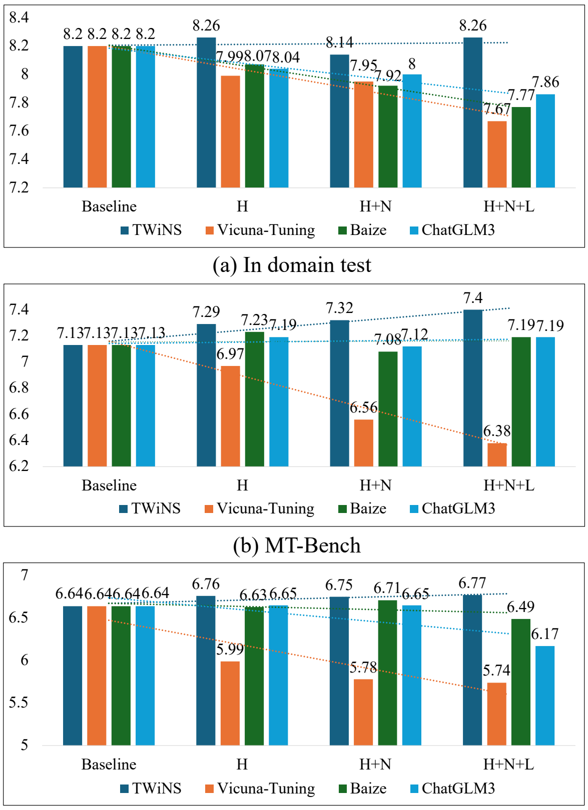
To evaluate the capability of the TWiNS fine-tuning method in addressing the negative optimization issue across diverse individual multi-turn dialogue datasets, we conduct a comparative analysis with Vicuna-Tuning approach. The experiments results demonstrate the TWiNS effectively filters out low quality data during training through end-to-end signaling, thereby achieving positive performance on both in-domain test sets and general multi-turn dialogue benchmarks, including MT_Bench and MT_Bench_Ext.

TWiNS addresses the negative optimization observed in Vicuna-Tuning. As shown in Table 2, TWiNS outperforms Vicuna-Tuning across multiple datasets, including ChatAlpaca, UltraChat, ShareGPT, and OpenAssistant. TWiNS consistently achieves positive optimization in both in-domain evaluations and the MT-Bench/MT-Bench-Ext benchmarks, whereas Vicuna-Tuning often shows negative optimization. Among the various datasets, ShareGPT reveals the largest performance gap between TWiNS and Vicuna-Tuning, likely due to LLaMA-3.2B-3B's sensitivity to ShareGPT. TWiNS achieves improvements of 6.11%, 9.82%, and 2.86% over the baseline for the in-domain test, MT-Bench, and MT-Bench-Ext, respectively. In contrast, Vicuna-Tuning attains 7.02%, 12.73%, and 12.65% on those benchmarks. Overall, these findings underscore TWiNS's robust performance in multi-turn dialogue tasks, driven by its consistent positive optimization.

TWiNS excels in longer, more complex multi-turn dialogues. For the Mtlingual (en) dataset (2,000 samples), both Vicuna-Tuning and TWiNS exhibit negative optimization due to overfitting, scoring 7.09 and 7.06 respectively. On the MT-Bench dataset, which primarily consists of two-turn dialogues, TWiNS is marginally outperformed by Vicuna-Tuning. However, TWiNS achieves a higher score on MT-Bench-Ext, where the dialogues are longer and more complex, indicating TWiNS's robustness in multi-turn conversational scenarios.

4.3.2 Does TWiNS skip junk data to enhance fine-tuning stability and performance?

To further validate the ability of the TWiNS fine-tuning method to leverage end-to-end signaling for skipping low-quality data, we conduct dialogue fine-tuning experiments on mixed datasets, comparing it with Vicuna-Tuning, Baize, and ChatGLM3. Figure 2 (a) shows that for in-domain evaluation, TWiNS maintains a stable score of around 8.2, despite increasing dataset complexity. By contrast, Vicuna-Tuning drops from 8.2 to 7.67, and Baize and ChatGLM3 exhibit smaller declines or only minor gains. Turning to the MT-Bench benchmark in Figure 2 (b), TWiNS steadily improves from 7.13 to 7.4, reflecting its clear positive trend in handling general multi-turn dialogue tasks. Vicuna-Tuning,



(a) In domain test

(b) MT-Bench

(c) MT-Bench-Ext

Figure 2: Divergent scaling patterns with multi-tier data integraion: performance evolution on H, H+N, and H+N+L mixtures. (a) In domain test Performance. (b) MT-Bench Performance. (c) MT-Bench-Ext Performance.

however, decreases from 7.13 to 6.38. Baize and ChatGLM3 also show moderate fluctuations with limited growth. Finally, in the MT-Bench-Ext evaluation (Figure 2 (c)), TWiNS displays a consistent upward trajectory from 6.64 to 6.77, while Vicuna-Tuning plunges from 6.64 to 5.74. Baize and ChatGLM3 once again reveal minor variations but lack the stable improvement seen in TWiNS.

TWiNS excels on partially noisy datasets, maintaining positive optimization. Building on these observations, we find that as mixed dataset complexity increases, different methods display varying resilience to noise. Conventional fine-tuning approaches (Vicuna-Tuning, Baize, ChatGLM3) tend to suffer from performance degradation when confronted with low-quality data. In contrast, TWiNS effectively filters out disruptive junk samples while preserving the beneficial signals present in larger and more diverse datasets. This robust noise-filtering capability enables TWiNS to consistently learn from high-quality data within

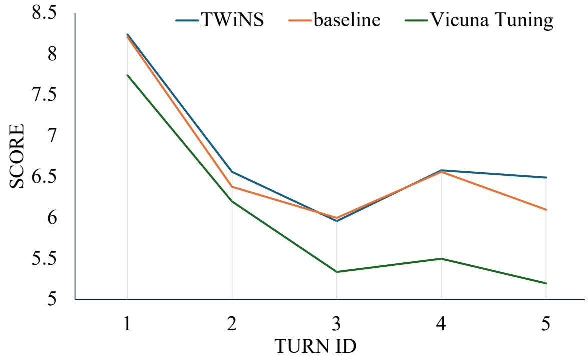


Figure 3: Turn-wise performance on MT-Bench Extended (GPT-4 evaluation).

mixed scenarios, thereby reinforcing its advantage in complex, multi-turn dialogue settings.

Turn-by-turn Analysis on MT-Bench Extended. We compare three training setups - no training, Vicuna tuning fine tuning, and TWiNS - on the MT-Bench Extended dataset using GPT-4 evaluation. Performance as shown in Figure 3 declines across turns, with a sharp drop from Turn 1 to Turn 2, highlighting the challenge of maintaining response quality in multi-turn dialogues. Our proposed fine-tuning consistently outperforms the baseline and Vicuna Tuning methods, demonstrating better response stability. Baseline model shows noticeable degradation in the last turn, while the Vicuna Tuned model performs the worst. These results emphasize the importance of fine-tuning for sustained dialogue quality.

5 Ablation Study

TWiNS excels in fine-tuning with mixed data types. To further validate the stability of our proposed TWiNS in fine-tuning performance across different types of data noise, we incorporated GSM8K, a mathematical question-answering dataset, alongside multi-turn dialogue datasets. After mixing GSM8K with the ChatAlpaca dataset, we compared the fine-tuning methods of TWiNS and Vicuna-Tuning. As shown in Fig. 4, our findings indicate that TWiNS maintains stable performance on in-domain tests and shows positive optimization on general multi-turn dialogue evaluation sets, even when compared to using a single high-quality multi-turn dialogue dataset. In contrast, Vicuna exhibits a decline in both in-domain tests and general dialogue evaluation sets. This decline is attributed to Vicuna's overfitting to the mathematical capabilities associated with GSM8K, which

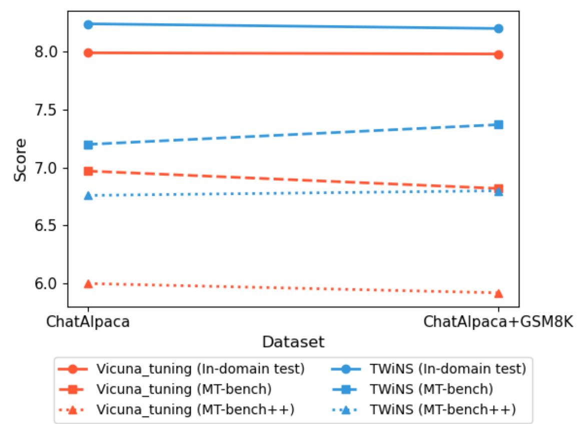


Figure 4: Performance comparison between Vicuna_tuning and Ours on ChatAlpaca and ChatAlpaca+GSM8K datasets. Three evaluation metrics are presented: In-domain test (solid line, circle marker), MT-bench (dashed line, square marker), and MT-Bench-Ext (dotted line, triangle marker).

Metric	w/ Welford	w/o Welford	R.I. (%)
In-domain Test	8.26	8.20	**+0.73%**
MT-Bench	7.29	7.19	**+1.39%**
MT-Bench-Ext	6.76	6.70	**+0.90%**

Table 3: Comparison of main experiment and ablation experiment with Relative Improvement (R.I.)

consequently weakens its multi-turn dialogue capabilities.

TWiNS enhances response performance by effectively skipping low-quality data. To evaluate the impact of TWiNS mechanism on multi-turn dialogue fine-tuning, we conduct an ablation study by removing the Welford Loss Calculation component while keeping all other loss updating functions unchanged. This ablation setup disables the skip mechanism, preventing the suppression of low-quality data during training. As shown in Table 3, the full TWiNS model outperforms the ablated version, achieving improvements of +0.73% on the in-domain test, +1.39% on MT-Bench, and +0.90% on MT-Bench-Ext. These results demonstrate that the skip mechanism in TWiNS effectively mitigates the influence of low-quality data, enhancing the model's robustness and fine-tuning stability.

6 Case Study

As shown in Figure 5, this multi-turn dialogue case shows TWiNS' superior contextual comprehension compared to Vicuna-Tuning. When analyzing the

1st	**User**: *In the sentence, ... , identify the subject* **Assistant**: Jack
2nd	**User**: ... *what the predicate is in that sentence?* **Assistant**: Yes, the predicate ...
3rd	**User**: *Can you explain to me what a predicate is?* **Assistant**: Sure! A predicate is one of the two main parts of a sentence. It usually includes the...
4th (Test case)	**User**: ***Can you give me an example of a sentence with a compound predicate?*** **Answer 1 (TWiNS)**: ...\"are going to the movies tonight\" is the first part of the compound predicate, and \"will see a new superhero movie\" is the second part of the compound predicate.... ***Score***: **9/10**, ***Rationale***: Correctly identify parallel verb structures. **Answer 2 (Vicuna-Tuning)**: ... the compound predicate includes two verbs: \"attended\" and **\"headed\".** ... ***Score***: **3/10**, ***Rationale***: Mistakenly parsing the noun phrase "headed" as a verb.

Figure 5: Case study.

compound predicate query, TWiNS accurately identifies parallel verb structures, correctly recognizing both predicate components ("are going" and "will see") with precise syntactic boundaries. In contrast, Vicuna-Tuning exhibits critical contextual misinterpretation. erroneously parsing the noun phrase "headed" as a verb predicate, confusing syntactic roles despite the explicit mention of "head" as a positional noun in the preceding context. This failure reveals Vicuna-Tuning's limitations in maintaining dialogue state awareness and tracking referential relationships across conversational turns. Detailed examples can be found in Appendix C.

7 Conclusion

In this study, we present TWiNS, which dynamically adjusts training instance contributions to preserve response quality and improve system robustness. Our method outperforms traditional fine-tuning across benchmarks like MT-Bench, MT-Bench-Ext, and in-domain tests, effectively mitigating the impact of noisy data. Ablation studies confirm the importance of dynamic loss control in optimizing multi-turn dialogue performance. Overall, TWiNS offers a robust and adaptable solution for fine-tuning multi-turn dialogue systems, particularly in handling noisy or low-quality data.

Limitation

This study has several limitations. First, we only evaluate the impact of multi-turn fine-tuning on the LLaMA 3.2 3B model. Due to computational constraints, we have not yet conducted experiments on other LLM families, which may affect the generalizability of our findings. Second, the proposed online statistical method is one possible approach, but alternative solutions may exist. Our study adopts a straightforward experimental setup without exploring more sophisticated strategies. Third, our evaluation of dataset quality serves as a reference rather than a definitive assessment, as different domains may require tailored quality evaluation metrics. Despite these limitations, we hope that our findings can provide insights for future researches on domain-specific fine-tuning.

Ethics Statement

Our work explores automatic noise filtering in multi-turn dialogue training using end-to-end signals. However, its implicit filtering mechanism may unintentionally remove valuable data, raising concerns about bias and information completeness. Moreover, the method is evaluated only in multi-turn dialogue scenarios, with broader applications limited by computational cost. Future work will address these challenges to enhance fairness and efficiency.

References

Anonymous. 2024. Self-evolved diverse data sampling for efficient instruction tuning. In *Submitted to ACL Rolling Review - June 2024*. Under review.

Ge Bai, Jie Liu, Xingyuan Bu, Yancheng He, Jiaheng Liu, Zhanhui Zhou, Zhuoran Lin, Wenbo Su, Tiezheng Ge, Bo Zheng, and Wanli Ouyang. 2024. MT-bench-101: A fine-grained benchmark for evaluating large language models in multi-turn dialogues. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7421–7454, Bangkok, Thailand. Association for Computational Linguistics.

Ning Bian, Hongyu Lin, Yaojie Lu, Xianpei Han, Le Sun, and Ben He. 2023. Chatalpaca: A multi-turn dialogue corpus based on alpaca instructions. https://github.com/cascip/ChatAlpaca.

Yihan Cao, Yanbin Kang, Chi Wang, and Lichao Sun. 2023. Instruction mining: Instruction data selection for tuning large language models. *arXiv preprint arXiv:2307.06290*.

Lichang Chen, Shiyang Li, Jun Yan, Hai Wang, Kalpa Gunaratna, Vikas Yadav, Zheng Tang, Vijay Srinivasan, Tianyi Zhou, Heng Huang, et al. 2023. Alpagasus: Training a better alpaca with fewer data. *arXiv preprint arXiv:2307.08701*.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality. https://lmsys.org/blog/2023-03-30-vicuna/. Accessed: 2025-02-10.

OpenAssistant Contributors. 2023. Openassistant conversations - democratizing large language model alignment. https://arxiv.org/abs/2304.07327. Accessed: 2023-04-17.

Nina Dethlefs, Helen Hastie, Heriberto Cuayáhuitl, Yanchao Yu, Verena Rieser, and Oliver Lemon. 2016. Information density and overlap in spoken dialogue. *Comput. Speech Lang.*, 37(C):82–97.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2024. Qlora: Efficient finetuning of quantized llms. *Advances in Neural Information Processing Systems*, 36.

Ning Ding, Yulin Chen, Bokai Xu, Yujia Qin, Zhi Zheng, Shengding Hu, Zhiyuan Liu, Maosong Sun, and Bowen Zhou. 2023. Enhancing chat language models by scaling high-quality instructional conversations. *arXiv preprint arXiv:2305.14233*.

Qianlong Du, Chengqing Zong, and Jiajun Zhang. 2023. Mods: Model-oriented data selection for instruction tuning. *arXiv preprint arXiv:2311.15653*.

Hugging Face and Meta AI. 2023. Llama-3.2-3b instruct: A fine-tuned version of llama for instruction following. https://huggingface.co/your-model-path. Accessed: [Insert Date].

Team GLM, Aohan Zeng, Bin Xu, Bowen Wang, Chenhui Zhang, Da Yin, Diego Rojas, Guanyu Feng, Hanlin Zhao, Hanyu Lai, Hao Yu, Hongning Wang, Jiadai Sun, Jiajie Zhang, Jiale Cheng, Jiayi Gui, Jie Tang, Jing Zhang, Juanzi Li, Lei Zhao, Lindong Wu, Lucen Zhong, Mingdao Liu, Minlie Huang, Peng Zhang, Qinkai Zheng, Rui Lu, Shuaiqi Duan, Shudan Zhang, Shulin Cao, Shuxun Yang, Weng Lam Tam, Wenyi Zhao, Xiao Liu, Xiao Xia, Xiaohan Zhang, Xiaotao Gu, Xin Lv, Xinghan Liu, Xinyi Liu, Xinyue Yang, Xixuan Song, Xunkai Zhang, Yifan An, Yifan Xu, Yilin Niu, Yuantao Yang, Yueyan Li, Yushi Bai, Yuxiao Dong, Zehan Qi, Zhaoyu Wang, Zhen Yang, Zhengxiao Du, Zhenyu Hou, and Zihan Wang. 2024. Chatglm: A family of large language models from glm-130b to glm-4 all tools.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

Zhiqiang Hu, Lei Wang, Yihuai Lan, Wanyu Xu, Ee-Peng Lim, Lidong Bing, Xing Xu, Soujanya Poria, and Roy Lee. 2023. LLM-adapters: An adapter family for parameter-efficient fine-tuning of large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 5254–5276, Singapore. Association for Computational Linguistics.

Andreas Köpf, Yannic Kilcher, Dimitri von Rütte, Sotiris Anagnostidis, Zhi Rui Tam, Keith Stevens, Abdullah Barhoum, Duc Nguyen, Oliver Stanley, Richárd Nagyfi, et al. 2024. Openassistant conversations-democratizing large language model alignment. *Advances in Neural Information Processing Systems*, 36.

Wai-Chung Kwan, Xingshan Zeng, Yuxin Jiang, Yufei Wang, Liangyou Li, Lifeng Shang, Xin Jiang, Qun Liu, and Kam-Fai Wong. 2024a. Mt-eval: A multi-turn capabilities evaluation benchmark for large language models. *arXiv preprint arXiv:2401.16745*.

Wai-Chung Kwan, Xingshan Zeng, Yuxin Jiang, Yufei Wang, Liangyou Li, Lifeng Shang, Xin Jiang, Qun Liu, and Kam-Fai Wong. 2024b. MT-eval: A multi-turn capabilities evaluation benchmark for large language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 20153–20177, Miami, Florida, USA. Association for Computational Linguistics.

Ming Li, Yong Zhang, Zhitao Li, Jiuhai Chen, Lichang Chen, Ning Cheng, Jianzong Wang, Tianyi Zhou, and Jing Xiao. 2024a. From quantity to quality: Boosting LLM performance with self-guided data selection for instruction tuning. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 7602–7635, Mexico City, Mexico. Association for Computational Linguistics.

Xian Li, Ping Yu, Chunting Zhou, Timo Schick, Omer Levy, Luke Zettlemoyer, Jason Weston, and Mike Lewis. 2023. Self-alignment with instruction back-translation. *arXiv preprint arXiv:2308.06259*.

Yunshui Li, Binyuan Hui, Xiaobo Xia, Jiaxi Yang, Min Yang, Lei Zhang, Shuzheng Si, Ling-Hao Chen, Junhao Liu, Tongliang Liu, Fei Huang, and Yongbin Li. 2024b. One-shot learning as instruction data prospector for large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4586–4601, Bangkok, Thailand. Association for Computational Linguistics.

Shih-Yang Liu, Chien-Yi Wang, Hongxu Yin, Pavlo Molchanov, Yu-Chiang Frank Wang, Kwang-Ting Cheng, and Min-Hung Chen. 2024a. Dora: Weight-decomposed low-rank adaptation. *arXiv preprint arXiv:2402.09353*.

Wei Liu, Weihao Zeng, Keqing He, Yong Jiang, and Junxian He. 2024b. What makes good data for alignment? a comprehensive study of automatic data selection in instruction tuning. In *The Twelfth International Conference on Learning Representations*.

Junru Lu, Siyu An, Mingbao Lin, Gabriele Pergola, Yulan He, Di Yin, Xing Sun, and Yunsheng Wu. 2023a. Memochat: Tuning llms to use memos for consistent long-range open-domain conversation. *arXiv preprint arXiv:2308.08239*.

Keming Lu, Hongyi Yuan, Zheng Yuan, Runji Lin, Junyang Lin, Chuanqi Tan, and Chang Zhou. 2023b. # instag: Instruction tagging for diversity and complexity analysis. *arXiv preprint arXiv:2308.07074*.

Rishabh Maheshwary, Vikas Yadav, Hoang Nguyen, Khyati Mahajan, and Sathwik Tejaswi Madhusudhan. 2024. M2lingual: Enhancing multilingual, multi-turn instruction alignment in large language models. *arXiv preprint arXiv:2406.16783*.

Fanxu Meng, Zhaohui Wang, and Muhan Zhang. 2024. Pissa: Principal singular values and singular vectors adaptation of large language models. *arXiv preprint arXiv:2404.02948*.

OpenAI. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Jiao Ou, Jiayu Wu, Che Liu, Fuzheng Zhang, Di Zhang, and Kun Gai. 2024. Inductive-deductive strategy reuse for multi-turn instructional dialogues. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 17402–17431, Miami, Florida, USA. Association for Computational Linguistics.

Nicole M Radziwill and Morgan C Benton. 2017. Evaluating quality of chatbots and intelligent conversational agents. *arXiv preprint arXiv:1704.04579*.

RyokoAI. 2023. Sharegpt. `https://huggingface.co/datasets/RyokoAI/ShareGPT52K`.

Lior Shani, Aviv Rosenberg, Asaf Cassel, Oran Lang, Daniele Calandriello, Avital Zipori, Hila Noga, Orgad Keller, Bilal Piot, Idan Szpektor, et al. 2024. Multi-turn reinforcement learning from preference human feedback. *arXiv preprint arXiv:2405.14655*.

Shivalika Singh, Freddie Vargus, Daniel Dsouza, Börje F Karlsson, Abinaya Mahendiran, Wei-Yin Ko, Herumb Shandilya, Jay Patel, Deividas Mataciunas, Laura OMahony, et al. 2024. Aya dataset: An open-access collection for multilingual instruction tuning. *arXiv preprint arXiv:2402.06619*.

Yuchong Sun, Che Liu, Kun Zhou, Jinwen Huang, Ruihua Song, Xin Zhao, Fuzheng Zhang, Di Zhang, and Kun Gai. 2024. Parrot: Enhancing multi-turn instruction following for large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9729–9750, Bangkok, Thailand. Association for Computational Linguistics.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. Stanford alpaca: An instruction-following llama model. `https://github.com/tatsu-lab/stanford_alpaca`.

Jiahao Wang, Bolin Zhang, Qianlong Du, Jiajun Zhang, and Dianhui Chu. 2024. A survey on data selection for llm instruction tuning. *arXiv preprint arXiv:2402.05123*.

Lai Wei, Zihao Jiang, Weiran Huang, and Lichao Sun. 2023. Instructiongpt-4: A 200-instruction paradigm for fine-tuning minigpt-4. *arXiv preprint arXiv:2308.12067*.

Chuhan Wu, Fangzhao Wu, Tao Qi, Yongfeng Huang, and Xing Xie. 2022. Noisytune: A little noise can help you finetune pretrained language models better. *arXiv preprint arXiv:2202.12024*.

Shengguang Wu, Keming Lu, Benfeng Xu, Junyang Lin, Qi Su, and Chang Zhou. 2023. Self-evolved diverse data sampling for efficient instruction tuning. *arXiv preprint arXiv:2311.08182*.

Heng-Da Xu, Xian-Ling Mao, Puhai Yang, Fanshu Sun, and Heyan Huang. 2024. Rethinking task-oriented dialogue systems: From complex modularity to zero-shot autonomous agent. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2748–2763, Bangkok, Thailand. Association for Computational Linguistics.

Yang Xu, Yongqiang Yao, Yufan Huang, Mengnan Qi, Maoquan Wang, Bin Gu, and Neel Sundaresan. 2023. Variety and quality over quantity: Towards versatile instruction curation. *arXiv preprint arXiv:2312.11508*.

Ningyu Zhang, Xiang Chen, Shumin Deng, Chuanqi Tan, Mosha Chen, Fei Huang, and Huajun Chen. 2023. Enhancing chat language models by scaling high-quality instructional conversations. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*.

Jiawei Zhao, Zhenyu Zhang, Beidi Chen, Zhangyang Wang, Anima Anandkumar, and Yuandong Tian. 2024a. Galore: Memory-efficient llm training by gradient low-rank projection. *arXiv preprint arXiv:2403.03507*.

Wenting Zhao, Xiang Ren, Jack Hessel, Claire Cardie, Yejin Choi, and Yuntian Deng. 2024b. Wildchat: 1m chatgpt interaction logs in the wild. In *The Twelfth International Conference on Learning Representations*.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in Neural Information Processing Systems*, 36:46595–46623.

Chunting Zhou, Pengfei Liu, Puxin Xu, Srinivasan Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, Lili Yu, et al. 2024. Lima: Less is more for alignment. *Advances in Neural Information Processing Systems*, 36.

A Data processing and Evaluation Prompts

During the evaluation of datasets, although the raw patterns of conversation data from different sources vary from each other, all of them are formatted as *[{'human': '<request>', 'assistant': '<response>'}, ... , {'human': '<request>', 'assistant': '<response>'}]* for each entire and independent conversation, before being written to the prompt. The ChatGPT version used in the evaluation is *ChatGPT-4o-2024-08-06*, and the complete prompts of the evaluation on *Connection*, *Quality*, *Information Density* and *Friendliness* are detailed in Figure 7, Figure 8, Figure 9, Figure 10 separately. In the evaluation, each aspect of each independent conversation is also graded independently.

B Datasets Introduction

Table 4 shows the datasets in this work. ShareGPT is a collection of 90k conversations shared via the ShareGPT API (closed at present), and includes both user prompts and responses from ChatGPT, which mainly consists of messages in English and other western languages. WildChat is a collection of 1 million real-world user-ChatGPT conversations which consists of over 2.5 million interaction turns and 68 languages from 204,736 users (Zhao et al., 2024b). OpenAssistant is a collection of 161,443 messages that construct over 10000 complete conversations, which consists of 35 different languages and over 40k annotations on quality, and is designed for reinforcement learning from human feedback. Hence, it provides different conversations based on the same initial question with different quality, which leads to the sacrifice of the overall quality. Another important and unique feature of OpenAssistant is that, it is totally generated and annotated by human (Köpf et al., 2024). ChatAlpaca is a collection of 20k conversations, generated by ChatGPT and started with the original Stanford Alpaca (Taori et al., 2023) data, and it contains English and Chinese version. MTLingual is a collection of 182k conversations in 70 languages, and is generated by Evol (Maheshwary et al., 2024). The type of language, task, user prompt, and seed prompt are also detailed in MTLingual. UltraChat is a collection of 1.5 million conversations and is generated by ChatGPT which simulates the interactions of human. The main concerns of UltraChat is diversity, scale, and coherence.

C Case Study

Figure 6 details a case showing TWiNS' superior contextual comprehension on multi-turn dialogue compared to Vicuna-Tuning. In this case, our method successfully identifies parallel verb structures while Vicuna-Tuning fails in the recognition of the parts of speech.

Dataset	Volume	Avg. Turns	Generation Type	Generation Mechanism	Annotated
ShareGPT (RyokoAI, 2023)	94K	3.51	User-ChatGPT	Voluntary sharing by netizens	No
WildChat (Zhao et al., 2024b)	1.04M	2.54	User-ChatGPT	Collected from chatbot services powered by GPT API	No
OpenAssistant (Köpf et al., 2024)	135.6K	2.34	Human only	Human-Generated & Annotated by volunteers	Yes
ChatAlpaca(Bian et al., 2023)	20K	4.32	ChatGPT	Follow-up by GPT from Stanford Alpaca(Taori et al., 2023)	No
MTLingual (Maheshwary et al., 2024)	182K	2.48	ChatGPT	Constructed by Evol from Aya(Singh et al., 2024)	No
UltraChat (Ding et al., 2023)	1.5M	3.80	ChatGPT	Simulate human interactions by ChatGPT	No

Table 4: Datasets in this work with features, the values of *Avg. Turns* of ShareGPT, WildChat and OpenAssistant derive from the work of WildChat (Zhao et al., 2024b), and the value of *Avg. Turns* of OpenAssistant calculates on multi-turn conversations in English.

1st	**User**: *In the sentence, \"Jack received a birthday present from his aunt Mary,\" identify the subject* **Assistant**: Jack
2nd	**User**: *Great job! Can you now tell me what the predicate is in that sentence?* **Assistant**: Yes, the predicate in the sentence is \"received a birthday present from his aunt Mary.
3rd	**User**: *Can you explain to me what a predicate is?* **Assistant**: Sure! A predicate is one of the two main parts of a sentence. It usually includes the verb and all the other words in the sentence that modify or complete the sentence's subject. The job of the predicate is to describe what the subject is doing or what is being done to it. For example, in the sentence \"Jack received a birthday present from his aunt Mary,\" \"received a birthday present from his aunt Mary\" is the predicate, and it describes what Jack did.
4th (Test case)	**User**: ***Can you give me an example of a sentence with a compound predicate?*** **Answer 1 (TWiNS)**: …\"Jack and Sarah are going to the movies tonight, and they will see a new superhero movie.\"\n\nIn this sentence, \"are going to the movies tonight\" is the first part of the compound predicate, and \"will see a new superhero movie\" is the second part of the compound predicate.… 😊 ***Score***: **9/10**, ***Rationale***: Correctly identify parallel verb structures. **Answer 2 (Vicuna-Tuning)**: …\"The manager of the store and the head of the marketing department attended the meeting.\" \n\nIn this sentence, the compound predicate includes two verbs: \"attended\" and **\"headed\".** The first verb is \"attended\", which describes the manager's action at the meeting. **The second verb is \"headed\",** which describes the head of the marketing department's action at the meeting. … 😔 ***Score***: **3/10**, ***Rationale***: Mistakenly parsing the noun phrase "headed" as a verb.

Figure 6: Case study.

Prompts of the Evaluation of Connection

"""
You are a strict and objective evaluator. Your task is to assess the quality of the final response from assistant in conversation content.
Your evaluation should be fair, professional, and reflect an expert judgment of the response's quality.
The conversation is formatted as [{'human': '...', 'assistant': '...'}, ..., {'human': '...', 'assistant': '...'}] .
The final response is the final 'assistant' message in the conversation.

[Conversation]\n""" + <conversation> + "\n" + """
Assessment Criteria:
Score baseline is 5. The final score should be adjusted based on the following criteria:
Connection: Does it utilize the information in the previous conversations?
Concentrate on the evidence of conflicts and coherence. Evidence of one conflict
should decrease the score by 1, and evidence of utilizing one information should increase the score by 1.
Relevance: Does it provide redundant information which is not related to the topic? Is so, it should be penalized by the degree and amount. One irrelevant information should decrease the score by 1.
Overall Score: Assign a score from 1 to 10 (10 being the best), considering all of the above factors.

The evaluation and your output must be strictly structured in the following JSON format:
{
"Explanation": "<Explain the rationale of your score.>",
"Score": <An integer score from 1 to 10.>
}
"""

Figure 7: Prompts of the Evaluation of Connection

Prompts of the Evaluation of Quality

"""
You are a strict and objective evaluator. Your task is to assess the quality of the each response from assistant in conversation, based on the Assessment Criteria.
Your evaluation should be fair, professional, and reflect an expert judgment of the response's quality.
The conversation is formatted as [{'human': '...', 'assistant': '...'}, ..., {'human': '...', 'assistant': '...'}].

[Conversation]\n""" + <conversation> + "\n" + """
Assessment Criteria:
Requirement Alignment: For each response, only consider the corresponding request from human in this turn, does the response meet the user's task goal?
Content Accuracy: Is the information in the response correct, clear, and logically organized?
Language Quality: Is the language fluent, coherent, and readable? Are there any obvious grammatical or word choice errors?
Consideration on previous information: If there is relevant information in the previous turns of chatting, does the response take them into consideration?
Overall Score: Assign a score from 1 to 10 (10 being the best), considering all of the above factors.

The evaluation and your output must be strictly structured in the following JSON format:
{
"evaluations": [
{
"Number of turn in conversation": 1,
"Explanation": "<Explain the rationale of your score.>",
"Score": <An integer score from 1 to 10.>
},
...,
{
"Number of turn in conversation": <Integer, the No. of turn in conversation>,
"Explanation": "<Explain the rationale of your score.>",
"Score": <An integer score from 1 to 10.>
}]
}
"""

Figure 8: Prompts of the Evaluation of Quality

Prompts of the Evaluation of Information Density

"""
You are a strict and objective evaluator. Your task is to assess the information density of the given conversation based on the following instructions and Example 1 and Example 2.
[Instructions]
Your evaluation should be fair, professional, and reflect an expert judgment of the response's quality.
The conversation is formatted as[{'human': '...', 'assistant': '...'}, ..., {'human': '...', 'assistant': '...'}].
The information density of a conversation is defined as a number of information units (e.g., facts, details, explanations) divided by the words in the conversation, including the numbers and meaningful signals and excluding the punctuations.
You should only consider the information related to the topic of the conversation and ignore any irrelevant or redundant information.
If the information unit is repeated in the conversation, it should be counted only once.
[Example 1]
Input Conversation: [{'human': 'What is the capital of France?', 'assistant': 'The capital of France is Paris.'}]
Output: {'Number of Information Units': 1, 'Total Number of Words': 12, 'Information Units': ['The capital of France is Paris.']}
[Example 2]
Input Conversation: [{'human': 'What is the capital of France?', 'assistant': 'The capital of France is Paris.'}, {'human': 'What is the population of Paris?', 'assistant': 'The population of Paris is 2.1 million.'}]
Output: {'Number of Information Units': 2, 'Total Number of Words': 25, 'Information Units': ['The capital of France is Paris.', 'The population of Paris is 2.1 million.']}

[Conversation]\n""" + <conversation> + "\n" + """
The evaluation and your output must be strictly structured in the following format:
{
"Number of Information Units": <Number of information units>,
"Information Units": [<Information unit 1>, <Information unit 2>, ...],
"Total Number of Words": <Total number of words>,
}
"""

Figure 9: Prompts of the Evaluation of Information Density

Prompts of the Evaluation of Friendliness

"""
You are a strict and objective evaluator. Your task is to assess the friendliness of the given conversation following the Assessment Criteria.
Your evaluation should be fair, professional, and reflect an expert judgment of the response's quality.
The conversation is formatted as [{'human': '...', 'assistant': '...'}, ..., {'human': '...', 'assistant': '...'}].

{[Conversation]}""" + <conversation> + """
Assessment Criteria:
Manner: Concentrate on the requests from the human, and also evaluate the assistant's responses, is there evidence of disobeying the rules or aggressive behaviors?
Security: Evaluate the security of the conversation. Does the conversation tend to be harmful or offensive, or does the response from the assistant being guided to reveal sensitive information?
Tone: Evaluate the overall tone of the conversation. Does it have a positive and friendly tone?
Politeness: Evaluate the politeness and courtesy of the assistant's responses. Overall Score: Assign a score from 1 to 10 (10 being the most friendly), considering all of the above factors.

The evaluation and your output must be strictly structured in the following JSON format:

"Explanation": "<Explain the rationale of your score.>",
"Score": <An integer score from 1 to 10.>

"""

Figure 10: Prompts of the Evaluation of Friendliness