RSDA: Restoring Stale Data Affinity via Dynamic Renovation Strategy for Mitigating Data Scarcity

Yidan Liang[1*], **Jia Zhu**[1*†], **Weijie Shi**[3], **Hanghui Guo**[2],
Yue Cui[4], **Jiawei Shen**[1], **Guoqing Ma**[1], **Jingjiang Liu**[1], **Qingyu Niu**[1], **Yilin Wang**[1],
Shimin Di[2], **Jiajie Xu**[5†]

[1]Zhejiang Key Laboratory of Intelligent Education Technology and Application, Zhejiang Normal University
[2]School of Computer Science and Engineering, Southeast University
[3]Department of Computer Science and Engineering, Hong Kong University of Science and Technology
[4]Alibaba Group, [5]School of Computer Science and Technology, Soochow University

Abstract

High-quality data is the cornerstone of advancing large language models. However, the field currently faces a critical dilemma: the supply of premium data is nearing depletion, while vast stale corpora remain underutilized. Our empirical analysis reveals that training models on such data directly often leads to performance degradation. We attribute this phenomenon to the **data affinity** gap, a misalignment stemming from the model's inability to effectively comprehend the data or inherent quality defects. To bridge this gap, we propose **Restoring Stale Data Affinity (RSDA)** framework. First, utilizing our proposed **potential entropy** metric, RSDA quantifies the latent value of samples to effectively identify stale data with higher renovation potential. Subsequently, the framework employs a **dynamic renovation strategy selection** mechanism to determine the optimal component-level strategy for each instance, transforming low-affinity stale samples into high-quality training data. Comprehensive experimental results demonstrate that RSDA effectively enhances data affinity, achieving performance improvements using less than 10% of the data volume, thereby underscoring that the latent potential of stale corpora remains largely untapped. Our code is available at https://github.com/wenfiii/RSDA.

1 Introduction

As the community transitions from model-centric scaling to Data-Centric, the focus has decisively shifted from expanding data quantity to enhancing data quality (Zhou et al., 2023a; Hoffmann et al., 2022; Kaplan et al., 2020). Recent findings confirm that high-quality, diverse instruction data is the true catalyst for alignment (Liu et al., 2023a, 2025; Wu et al., 2025). Crucially, prioritizing the use of high-quality samples over large amounts of low-quality or randomly selected data can significantly reduce training costs while maintaining or even improving model performance (Golovneva et al., 2022; Zhang et al., 2024; Cao et al., 2023). However, this paradigm faces an imminent bottleneck: the exhaustion of high-quality public text data, projected to occur as early as 2026 (Villalobos et al., 2022).

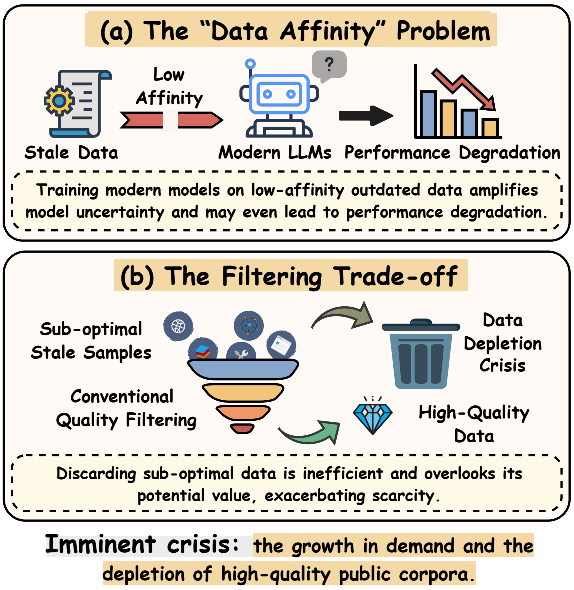

Figure 1: The dual challenges in the current LLMs landscape: Data Scarcity and the Negative Transfer caused by low data affinity.

The scarcity problem forces researchers to reconsider the value of massive stale data (Zhuang et al., 2024; Guo et al., 2025). However, our empirical analysis on stale datasets like GSM8K (Cobbe et al., 2021) reveals an issue: training modern large language models (LLMs) on these raw data samples incurs substantial computational costs; crucially, it leads to an average 5.39% performance degradation compared to the base model. We attribute this phenomenon to a lack of **data affinity**, which is

*These authors contributed equally.
†Corresponding author.
e-mail: liangyidan@zjnu.edu.cn, jiazhu@zjnu.edu.cn, xujj@suda.edu.cn

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 8280–8309
July 2-7, 2026

defined as the degree of structural and logical alignment between the training data distribution and the model's evolving preference (Li et al., 2024). Since LLMs have evolved to possess strong reasoning capabilities, training models on such low-affinity data constrains their potential, acting as a regression rather than an enhancement.

This creates a critical dilemma in the contemporary data-centric landscape. As illustrated in Figure 1, first, directly training models on low-affinity stale data compromises the model capabilities. Second, conventional filtering measures are becoming increasingly unsustainable. By discarding data that fail to meet static quality thresholds, existing approaches overlook the potential value embedded in suboptimal samples, while exacerbating the data depletion crisis. Therefore, it is meaningful to shift the research paradigm from assessing data quality to actively excavating data potential.

To address these challenges, we propose the Restoring Stale Data Affinity (RSDA) framework, a two-tiered architecture designed to renovate rather than discard data. Grounded in the theory of data affinity, RSDA shifts the focus from judging **current quality** to estimating **potential value**. In the first layer, utilizing our **potential entropy** metric, the framework executes a strategic triage to categorize raw corpora into three distinct sets: high-affinity data for retention, irrecoverable noise for discarding, and stale samples that warrant renovation. In the second layer, we implement a mark-driven dynamic renovation strategy selection mechanism. Moving beyond coarse-grained processing, this module operates at the component level, independently analyzing the *instruction*, *input*, and *output* segments of each sample. By assigning the optimal strategy prompt to each part, RSDA achieves a comprehensive and targeted renovation tailored to the specific deficiencies of each instance. Our contributions are summarized as follows:

- We introduce **data affinity** to explain model degradation caused by stale data and propose the **potential entropy** metric to quantify sample renovation value by balancing model uncertainty and data deficiencies.

- We propose the **Restoring Stale Data Affinity (RSDA)** framework, transitioning from passive "quality filtering" to active "strategic renovation". By leveraging a dynamic strategy selection mechanism for component-level targeted renovation, RSDA maximizes the utilization of stale data to bridge the data affinity gap.

- We conduct comprehensive experiments on multiple benchmarks. Our approach effectively reverses model degradation using less than 10% of the data volume, particularly in complex reasoning tasks, validating the unexploited potential of low-scoring data previously deemed as noise.

2 Related Work

2.1 SFT and Preference Alignment

While Supervised fine-tuning (SFT) remains the dominant paradigm, recent advancements have largely prioritized optimization algorithms over data quality. Research has explored theoretical alignment between SFT and human preferences (Wang et al., 2025a), training stability via feedback mechanisms (Zhu et al., 2025), and domain-specific adaptation (Li et al., 2025; Xu et al., 2024b; Behrouz et al., 2025). However, these methods predominately focus on algorithmic improvements, often overlooking the critical impact of intrinsic data affinity and quality on model performance.

2.2 Data Selection and Evolution

Addressing data scarcity and quality has bifurcated into two main streams, selection and evolution.

Data Selection (Filtering). The premise of selection is "Less is More." MoDS (Du et al., 2023) utilizes reward models and K-Center clustering to select high-quality subsets, proving that small, high-quality data can outperform full datasets. Similarly, the Clustering and Ranking (CaR) method (Ge et al., 2024) demonstrated that using only 1.96% of Alpaca's data could yield significant performance gains. While effective at removing noise, these "subtractive" methods inherently discard a vast amount of data that, while currently imperfect, holds semantic value, thus failing to mitigate the scarcity problem fundamentally.

Data Evolution (Augmentation). To generate more data, methods like PromDA (Wang et al., 2022) and Evol-Instruct (Xu et al., 2024a) employ heuristics or LLMs to rewrite instructions. Seo et al. (Seo et al., 2024) used LLMs to generate domain-adaptive samples to bridge distribution shifts. However, most existing augmentation strategies apply generic transformation rather than targeted repairs. Nguyen et al. (Nguyen et al., 2025) emphasized that traditional filtering and simple transfer learning are insufficient for resource-constrained environments. Our work bridges this gap by potential-aware selection, transforming stale data into high-affinity resources rather than simply filtering or randomly augmenting them.

3 Methodology

In this section, we present the **Restoring Stale Data Affinity (RSDA)** framework, a systematic approach designed to mitigate the affinity mismatch between stale corpora and modern LLMs. Diverging from traditional paradigms that rely on subtractive filtering based on static quality metrics, RSDA advocates for a shift towards dynamic potential estimation. The architecture operates in two tiers. First, quantifying the evolutionary value of data via potential entropy to identify potential samples. Secondly, executing dynamic strategy renovation to surgically reconstruct these samples.

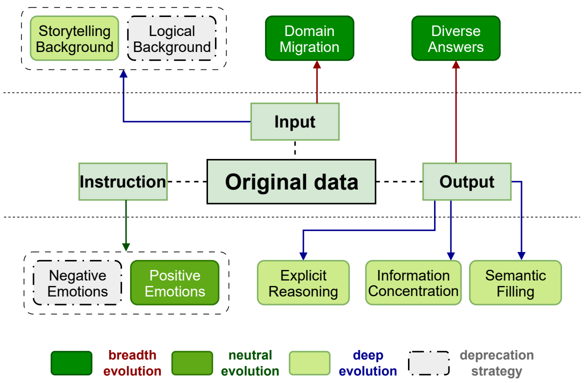

Figure 2: The Renovation Strategy Tree. Renovation directions are categorized into breadth evolution (red line), targeting semantic expansion; and deep evolution (blue line), targeting logical depth and information density. Neutral evolution (green line) addresses stylistic adjustments.

3.1 Construction of the Renovation Strategy

To scientifically determine the optimal directions for data renovation, we conducted a preliminary pilot study utilizing a subset of 1K samples from the GSM8K (Cobbe et al., 2021) dataset. We deconstruct the standard fine-tuning data into three atomic components: "***Instruction***, ***Input***, ***Output***", and structured our renovation strategies along two evolutionary dimensions, as illustrated in Figure 2. Breadth evolution focuses on expanding the **semantic scope and diversity** of the data. Strategies in this category aim to break the distributional constraints of the original corpus. In contrast, deep evolution focuses on enhancing the **information density and logical complexity** within the existing semantic scope. Strategies here aim to improve the quality and richness of the content. Moreover, we also consider *Neutral Evolution* strategies, which optimize the stylistic presentation without altering the core semantic logic.

The formulation of these renovation strategies is not arbitrary. Rather, it represents a synthesis from prior literature (Setlur et al., 2024; Chen et al., 2022) and empirical evidence derived from our extensive pilot studies. We strictly adhered to a univariate renovation principle during the pilot phase, applying exactly one strategy per sample to isolate its effect. We employed the Qwen series (Bai et al., 2023; Team et al., 2024; Qwen et al., 2025) for fine-tuning to investigate whether preference differences exist between different iterations within the same model series. Fine-tuning experiments on the Qwen series revealed that the model exhibits a strong preference for data enriched via deep evolution and specific breadth evolution strategies. Conversely, certain aggressive open-ended rewriting strategies led to semantic drift. Based on these empirical results, we pruned ineffective branches and retained a final set of 8 high-efficacy strategies for the RSDA framework.

Additional experimental results on model performance variations are provided in the Appendix A, along with detailed prompt descriptions for different strategies.

3.2 The RSDA Framework Architecture

Building upon the definitions of potential entropy and mark-driven strategy selection, we propose the Restoring Stale Data Affinity (RSDA) framework. As illustrated in Figure 3, RSDA operates as a closed-loop data iterative system designed to maximize the utility of existing corpora. Grounded in the theory of data affinity (Becker et al., 2024; Wang et al., 2025b), our framework shifts the paradigm from a static "judgment of quality" to a dynamic "estimation of potential", aiming to structurally align stale data with the model's preference manifold. As shown below, the workflow consists of three distinct phases:

Potential Entropy-Aware Selection. The process initiates with the potential entropy. Instead of performing a generic quality scan, this module computes the potential entropy $\mathcal{E}_{pot}$ for each sample. Based on the renovation threshold, samples with

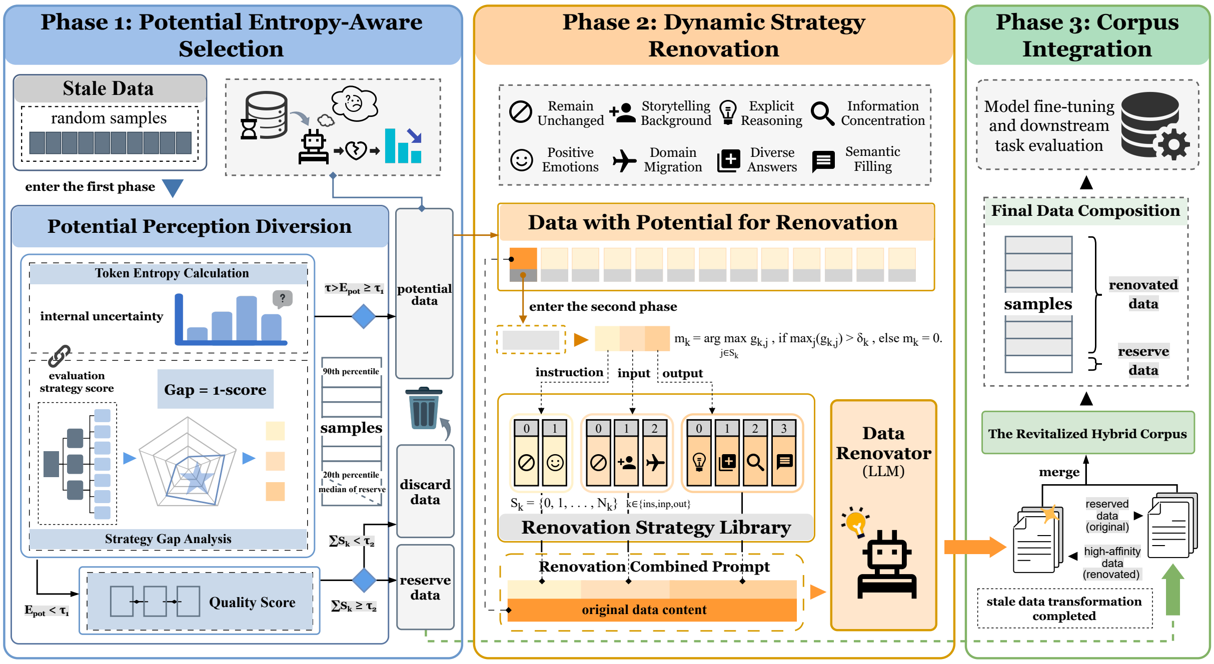

Figure 3: The overall architecture of the RSDA framework. The system employs a Potential Entropy Splitter to stratify raw data into Reserve, Discard, and Renovate streams. A Prompt Combiner then dynamically assembles rewriting instructions based on strategy marks ($\mathcal{M}$) to transform high-potential data.

low potential are routed to a quality filter, where high-fidelity instances are retained in the *Reserve* pipeline, while the remaining noise is discarded.

Dynamic Strategy Renovation. For samples identified as high-potential, the system eschews "one-size-fits-all" rewriting. Instead, it utilizes the generated mark $\mathcal{M}(x)$ as a control vector. The mark retrieves module-specific prompts from our policy repository and integrates them into a complete prompt. The data renovator then executes a targeted rewriting process, transforming the stale data into an enhanced version aligned with high-value rhetorical and logical patterns.

Corpus Integration. In the final phase, the repaired samples will be merged with the original high-fidelity data at a specific ratio to construct the final optimized dataset. This revitalized hybrid corpus ensures that the target model benefits from both the foundational correctness of the original distribution and the sophisticated reasoning structures introduced by the renovation process.

In the following sections, we proceed to detail of the two pivotal components powering RSDA.

3.2.1 Potential Entropy Modeling

Prior studies have established the efficacy of LLMs in evaluating reasoning quality (Chiang and Lee, 2023; Liu et al., 2023b; Fu et al., 2024; Yu et al., 2023) and guiding synthetic data generation (Wang et al., 2023). Building upon these capabilities, we introduce **potential entropy** $\mathcal{E}_{pot}$, a composite metric designed to quantify the latent evolutionary value of stale data that integrates internal model uncertainty signals and external semantic structure assessments. Unlike static quality scores that filter data based on current correctness, $\mathcal{E}_{pot}$ identifies samples within the model's zone of proximal development, those that represent high-value renovation targets. Our design is grounded in the hypothesis that data samples possessing both high epistemic uncertainty and significant structural deficiency offer the maximum "renovation gain", which stems from two key observations:

Internal Uncertainty $\mathcal{H}_{pred}$. High predictive entropy reflects the model's difficulty in comprehending the data distribution, indicating potential ambiguity, incompleteness, or knowledge boundaries (Kim et al., 2024).

External Strategy Deficiency $\mathcal{S}_{gap}$. Data enhancement transcends mere error correction; it involves transforming stale samples to align with high-affinity rhetorical and logical patterns.

We posit that a larger deviation between the original data structure and our optimized strategies implies a greater potential for optimization. Formally, to quantitatively evaluate the necessity of renovating a given sample x, we define the potential entropy as a weighted fusion of these two signals. Given a token sequence $x = \{t_1, t_2, ..., t_L\}$ of length L, the metric is composed of the **predictive entropy** at the token level and the **standardized strategy gap**, which we detail in the following.

Model Predictive Entropy. The first component measures the model's uncertainty regarding the data distribution. We utilize the average token entropy to quantify this. For a target model with parameters θ and a sequence $x = \{t_1, t_2, \ldots, t_L\}$, the predictive entropy is calculated as:

$$\mathcal{H}_{pred}(x) = -\frac{1}{L}\sum_{i=1}^{L}\log p_\theta(t_i|t_{<i}), \quad (1)$$

where $p_\theta(t_i|t_{<i})$ represents the probability of the i-th token given the preceding context.

A higher $\mathcal{H}_{pred}$ implies that the sample lies near the decision boundary or outside the model's current knowledge manifold, offering the highest potential information gain once their structural ambiguities are resolved through renovation.

Strategy Gap Score. Unlike standard scalar scoring methods, our model leverages the structural properties of the Alpaca format to conduct fine-grained strategy similarity assessment.

We define a strategy alignment score $s_k \in [0, 1]$, which measures the semantic similarity between the current data component k and the target renovation strategies. A high s_k indicates the data already possesses the desired traits, while a low s_k implies a significant opportunity for improvement.To quantify the renovation potential based on this distance, the model performs a strategy similarity assessment. For each component $k \in \{ins, inp, out\}$ of the data sample x, the model evaluates its semantic proximity to the corresponding strategy set $\mathbb{S}_k$:

$$s_k = \text{Sim}(x_k, \mathbb{S}_k) \in [0, 1], \quad (2)$$

where $\text{Sim}(\cdot)$ denotes the model's self-evaluated degree of congruence between the current text component x_k and the target design patterns in $\mathbb{S}_k$.

The strategy gap score ($\mathcal{S}_{gap}$) is then calculated as the weighted aggregation of the dissimilarity across components:

$$\mathcal{S}_{gap}(x) = \sum_{k\in\{ins,inp,out\}} \gamma_k \cdot \phi_k \cdot (1 - s_k), \quad (3)$$

where $(1 - s_k)$ represents the raw strategy gap. A value close to 1 indicates the data is "primitive" or "plain" relative to our advanced strategies, signaling a high priority for renovation. γ_k is a tunable hyperparameter, satisfying $\sum \gamma_k = 1$, which we calibrate to balance the contribution of each component based on its statistical distribution and empirical importance. $\phi_k \in \{0, 1\}$ serves as a validity gating indicator; it is set to 0 if the component is deemed irreparable noise, effectively masking it from the potential calculation.

Joint Calculation of Potential Entropy. Consequently, the final Potential Entropy $\mathcal{E}_{pot}(x)$ is formulated as a composite metric that balances the model's internal epistemic uncertainty with the external structural plasticity of the data:

$$\mathcal{E}_{pot}(x) = \alpha \cdot \underbrace{\mathcal{N}(\mathcal{H}_{pred}(x))}_{\text{Model Perplexity}} + \beta \cdot \underbrace{\mathcal{N}(\mathcal{S}_{gap}(x))}_{\text{Renovation Potential}}, \quad (4)$$

where $\mathcal{N}(\cdot)$ denotes the Min-Max normalization function, mapping different metric scales into a unified $[0, 1]$ range. α and β are hyperparameters regulating the trade-off between exploring knowledge boundaries and exploiting structural deficiencies.

Based on this potential estimation, we design a hierarchical decision gate $D(x)$ to stratify the data flow into three distinct pipelines: **Renovate**, **Reserve**, and **Discard**:

$$D(x) = \begin{cases} \text{Renovate}, & \tau > \mathcal{E}_{pot}(x) \geq \tau_1 \\ \text{Reserve}, & \mathcal{E}_{pot}(x) < \tau_1 \wedge \mathbf{q}(x) \geq \tau_2 \\ \text{Discard} & \text{otherwise} \end{cases}, \quad (5)$$

where τ_1 serves as the renovation activation threshold, filtering for samples with high evolutionary potential. τ acts as the **Noise Cutoff**, pruning samples dominated by extreme epistemic uncertainty or severe structural corruption that exceeds the model's renovation capability. τ_2 acts as the preservation baseline, it ensures that samples are retained in the "Reserve" subset only if their aggregate intrinsic quality meets a rigorous standard of fidelity. $\mathbf{q}(x) = \sum_{k\in\{ins,inp,out\}} s_k$ represents the cumulative raw quality, calculated as the summation of the semantic alignment scores across all data components.

Samples with high $\mathcal{E}_{pot}$ enter the data renovated, where their structural deficiencies are mapped to precise renovation strategies. This hierarchical filtering ensures that the "Reserve" subset consists solely of high-fidelity data, while the "Renovate" subset targets data with high evolutionary value.

Notably, RSDA is designed to minimize computational overhead through a **single-pass evaluation** mechanism. Specifically, the strategy alignment scores $\mathbf{s} = \{s_k\}$ are cached and subsequently reused for three distinct downstream decisions without additional inference: (1) calculating the Strategy Gap $\mathcal{S}_{gap}$; (2) deriving the cumulative quality score $q(x)$; and (3) determining the specific renovation mark $\mathcal{M}(x)$. This design ensures that the entire framework incurs a fixed cost of exactly two model calls per renovated sample (one for evaluation, one for renovation).

3.2.2 Mark-Driven Strategy Selection

However, simply identifying a sample as eligible for renovation ($\mathcal{E}_{pot}(x) \geq \tau_1$) is insufficient, as a coarse-grained binary decision fails to address the specific structural deficits of individual samples (Nguyen et al., 2025). To resolve this ambiguity and enable strategy-specific rewriting, we define the strategy mark $\mathcal{M}(x)$ as a vector of indices, guiding the system to execute surgical interventions.

Let $\mathbb{S}_k = \{0, 1, \ldots, N_k\}$ denote the set of available renovation strategy indices for component $k \in \{ins, inp, out\}$. For each candidate strategy $j \in \mathbb{S}_k$, we evaluate the specific strategy gap $g_{k,j} = 1 - s_{k,j}$. The mark m_k is assigned to the strategy exhibiting the maximum deficiency, provided it exceeds the sensitivity threshold δ_k:

$$m_k = \begin{cases} \arg\max\limits_{j \in \mathbb{S}_k} g_{k,j} & \max_j(g_{k,j}) > \delta_k \\ 0 & \text{otherwise} \end{cases}, \quad (6)$$

where m_k denotes the final renovation strategy index. While $m_k = 0$ typically implies an *identity mapping*, we implement a crucial exception for the *output* component ($k = out$): here, a value of 0 defaults to the baseline *explicit reasoning* strategy, ensuring that logical enhancement is applied even when specific deficiencies are absent. Additionally, $\mathbb{S}_k$ represents the candidate strategy set, and δ_k acts as the sensitivity threshold. The metric $g_{k,j} \in [0, 1]$ quantifies the **strategy gap**, representing the semantic deficiency of component k relative to strategy j.

Ultimately, the renovation mark is represented as $\mathcal{M}(x) = [m_{ins}, m_{inp}, m_{out}]$, with $m_k \in \{0, 1, \ldots, N_k\}$. This mark-based mechanism allows the prompt combiner to retrieve the exact template corresponding to the identified deficiency, thereby enabling the LLM to perform surgical rewriting rather than generic blind paraphrasing.

In conclusion, RSDA enables a precise stratification mechanism that safeguards high-fidelity knowledge via the Reserve pipeline while unlocking the latent value of stale data through the Renovate pipeline, transforming traditional augmentation into a surgical, strategy-aligned intervention.

4 Experiment

In this section, we conduct a comprehensive empirical evaluation of the RSDA framework to validate its effectiveness in enhancing fine-tuning instruction data utility, focusing on whether the proposed potential entropy metric and renovation strategy enable foundation models to achieve superior performance with lower data consumption and higher data affinity.

Table 1: Benchmark Datasets for Evaluation.

Benchmark	Metric
Gsm8k (Cobbe et al., 2021)	exact_match
HumanEval (Chen, 2021)	pass@1
MBPP (Austin et al., 2021)	pass@1
HellaSwag (Zellers et al., 2019)	acc_norm
MMLU (Hendrycks et al., 2020)	Accuracy
ARC-Challenge (Clark et al., 2018)	acc_norm
MATH (Hendrycks et al., 2021)	math_verify
IFeval (Zhou et al., 2023b)	prompt_strict_acc

4.1 Experimental Setup

4.1.1 Datasets and Evaluation Metrics

Seed Datasets. We primarily utilize two distinct datasets, the synthetic *Stanford Alpaca-52k* (Taori et al., 2023) and the human-authored *Databricks-Dolly-15k* (Conover et al., 2023), as seed corpora for renovation to verify RSDA's effectiveness in refining heterogeneous stale data sources.

Evaluation Benchmarks. To rigorously assess the capabilities of the fine-tuned models, we employ a diverse suite of benchmarks detailed in Table 1. Following standard protocols, we report accuracy (%) or pass@1 for downstream tasks.

Backbone Models. To verify the generalization of our framework across different architectures, we adopt two representative open-source foundation models as backbones: *Meta-Llama-3-8B* (Dubey et al., 2024) and *Qwen2.5-7B* (Yang et al., 2025).

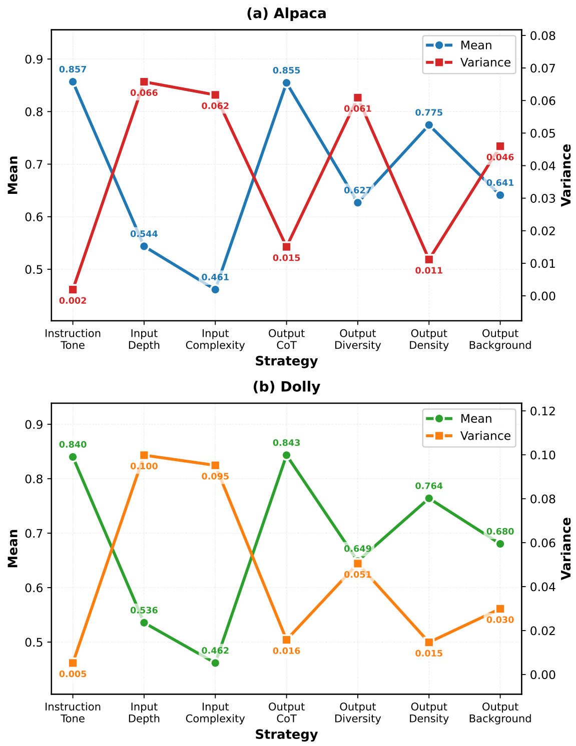

Figure 4: Statistical analysis of strategy evaluation scores across different dimensions. The solid lines with diamond markers denote the mean values, while lines marked with circles indicate the variance. The plots depict results for (a) Alpaca and (b) Dolly, calculated based on 1,000 randomly sampled instances.

Baselines. RSDA against three categories of data strategies to ensure a holistic evaluation:

- **Standard:** Fine-tuning on the full dataset and a randomly sampled subset.
- **Selection-based Methods:** Methods that filter high-quality subsets, including score-based filtering and curated quality, such as AlpaGasus (Chen et al., 2023), LIMA (Zhou et al., 2023a), and Deita (Liu et al., 2023a).
- **Evolution-based Methods:** Methods that rewrite data for complexity or quality, such as CoachLM (Liu et al., 2024), WizardLM (Xu et al., 2023), and Magpie (Xu et al., 2024c).

For a fair comparison, we re-implemented the selection and renovation pipelines of these baselines using the same backbone models and training hyperparameters as RSDA. All experiments are conducted on NVIDIA A100 Tensor Core GPUs. We employ QLoRA (Dettmers et al., 2023) for efficient fine-tuning to reduce memory usage while maintaining performance.

4.1.2 Asymmetric Policy Configuration

A notable feature of our framework is the asymmetric design of policy index 0, as shown in the Appendix D. For Instruction and Input, index 0 represents an identity mapping, reflecting our observation that structural preservation is sometimes optimal for these components. However, for the Output component, we set index 0 to explicit reasoning rather than identity. This design is grounded in our ablation studies and the consensus that explicit reasoning universally enhances the learnability of instruction-tuning data. The renovation process in RSDA is powered by GPT-4o (OpenAI, 2024) to execute the mark-driven rewriting directives. For the specific definitions of the 8 renovation strategies in our strategy tree and the corresponding prompt templates, please refer to Appendix H.

4.2 Hyperparameter Configuration

Guided by the statistical distribution of strategy scores across 1,000 random samples from the Alpaca and Dolly datasets in Figure 4(see Appendix B.2 for details) and extensive ablation studies (Appendix A.2), we calibrated the optimal configuration for the RSDA framework. To prioritize the high-impact output component while mitigating instruction saturation, we established the component weights as $\gamma_{ins} = 0.15$, $\gamma_{inp} = 0.35$, and $\gamma_{out} = 0.50$. For potential estimation, we balanced the trade-off by setting fusion coefficients to $\alpha = 0.4$ and $\beta = 0.6$. The renovation scope is dynamically stratified using percentile-based thresholds: the noise cutoff τ is set to the 90th percentile, and the renovation floor τ_1 to the 20th percentile, ensuring the middle 70% of samples enter the renovation pipeline. Finally, regarding sensitivity thresholds, we set $\delta_{ins/out} = 0.1$. For the input component, we excluded zero-score samples to mitigate the impact of empty values and set $\delta_{inp} = 0.12$ based on the mean of the valid data, ensuring the threshold targets valid semantic content. Comprehensive experimental results are detailed in Appendix B.

4.3 Evaluation on RSDA

Table 2 presents the comprehensive evaluation results on the Alpaca-52k dataset. Experimental results indicate that while RSDA utilizes an exceptionally minimal data volume of approximately 1.5% of the full dataset, it consistently surpasses both the full-data fine-tuning and baseline methods

Table 2: Comparison with state-of-the-art data strategies on Alpaca-52k across different backbones. We re-implemented all baselines to ensure fair comparison. The text **marked in red** indicates the original data used for renovated, while the **text in grey** represents the renovated data generated by our method.

Method	Data Size	General Ability	Reasoning				Common Sense	
		IFEval	ARC-Challenge	GSM8K	HumanEval	MBPP	MMLU	Hellaswag
Qwen2.5-7B								
NULL	-	29.39	63.41	81.58	77.4	75.9	74.21	79.68
Alpaca-52k (Full)	∼52,000	**40.30**	64.16	76.35	70.7	75.9	72.92	78.92
Random Samples	1,000	31.79	63.23	80.49	75.6	**79.1**	72.18	79.38
AlpaGasus	9,000	31.05	63.74	81.80	74.4	**79.1**	74.24	79.37
LIMA	1,000	36.30	63.39	82.52	77.4	**79.1**	74.25	79.75
Deita	6,000	34.75	63.99	81.96	76.2	78.3	74.08	79.47
CoachLM	1,000	31.05	63.57	82.49	76.2	78.6	74.12	79.23
WizardLM	1,000	28.65	64.16	82.79	77.6	76.3	74.21	79.76
Magpie	1,000	29.59	63.82	83.61	78.0	75.6	74.11	79.82
RSDA	∼800	30.76	**64.37**	**83.85**	**78.0**	78.9	**74.31**	**80.38**
Meta-Llama-3-8B								
NULL	-	9.98	58.19	50.27	37.2	61.9	65.43	81.44
Alpaca-52K (Full)	∼52,000	**26.06**	**63.14**	43.29	34.8	47.9	63.15	82.13
Random Samples	1,000	15.71	60.67	48.11	34.8	55.6	63.29	81.14
AlpaGasus (*selection*)	9,000	23.84	61.97	50.80	35.4	55.6	64.21	82.00
LIMA (*selection*)	1,000	12.57	58.87	50.57	35.3	54.0	65.35	81.94
Deita (*selection*)	6,000	21.44	62.12	50.19	34.1	57.1	64.96	81.98
CoachLM (*evolution*)	1,000	11.09	61.09	50.19	34.8	54.2	65.19	81.14
WizardLM (*evolution*)	1,000	13.06	61.75	52.12	37.8	45.5	64.69	82.29
Magpie (*evolution*)	1,000	11.90	60.56	51.74	**38.7**	55.0	64.75	81.91
RSDA	∼800	11.28	62.32	**52.62**	**38.7**	**57.4**	**65.51**	**82.58**

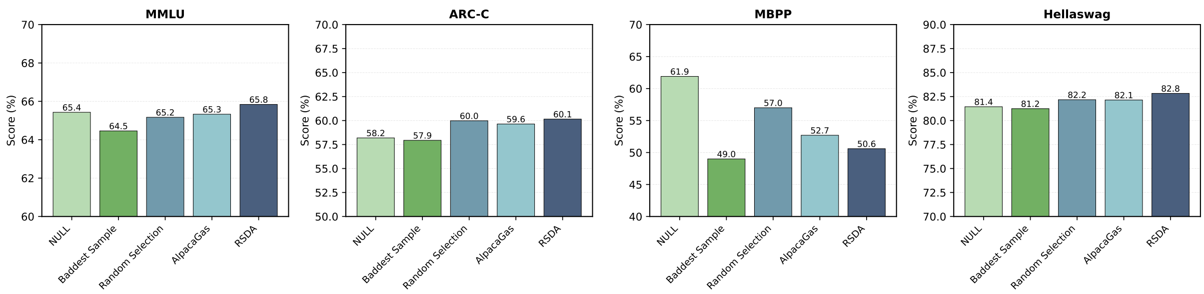

Figure 5: Results on the Databricks-Dolly dataset validating the reusability of low-quality data.

across the majority of benchmarks. The most substantial gains are observed in complex reasoning capabilities. Furthermore, it maintains top-tier performance in general knowledge and common-sense tasks such as MMLU and Hellaswag.

To validate the latent value of reusing low-quality data, Figure 5 compares the performance of models fine-tuned on different subsets of the Databricks-Dolly dataset (the complete evaluation results are detailed in Table 9). We establish "NULL" and "Random Samples" as baselines. The study specifically focuses on the "Lowest-Scoring Sample", defined as the 1,000 instances with the lowest AlpaGasus scores, which are conventionally filtered out as noise. RSDA is applied directly to this specific low-quality subset to execute data renovation. Remarkably, despite its smaller size, the RSDA-renovated data surpasses the performance of the AlpaGasus baseline, which utilizes 9,000 high-quality samples explicitly selected. This underscores that the utility of stale data is far from exhausted and holds substantial promise for reuse.

Furthermore, training and evaluation results on Qwen2.5-32B-Instruct demonstrate the superiority of RSDA, exhibiting an across-the-board advantage in various downstream benchmarks. For additional detailed analyses, including tests on larger-scale models, the complete evaluation on the Databricks-

Dolly dataset, and extended discussions of our results refer to Appendix E.

5 Conclusion

In this work, we introduced **RSDA**, a framework that shifts the focus from passive **data cleaning** to active **data renovation**. Anchored in the theory of **data affinity**, RSDA leverages **potential entropy** to precisely pinpoint stale samples with high renovation potential, while our **mark-driven strategy selection** mechanism executes targeted, interpretable interventions to address instance-specific component deficiencies. Empirical results demonstrate that stale data discarded as "low-quality" noise possess latent value and, once renovated, they can become highly effective training data. By renovating these stale resources, RSDA effectively mitigates data affinity issues, achieveing superior performance across multiple benchmarks.

Limitations

Despite the promising results demonstrated by the RSDA framework, we acknowledge several limitations in our current implementation that warrant further investigation:

Dependency on Backbone Model Capabilities. The efficacy of our framework relies heavily on the LLM. If the model acts as an imperfect judge, the calculated s_k may be inaccurate, leading to misclassification of data into the wrong pipeline. Although our second-order primitive mass gate $\mathbf{q}$ alleviates this problem to some extent, we still need to pay attention to the illusion problem that is inherent to the use of LLM.

Finite Scope of the Renovation Strategy Tree. The current Renovation Strategy Tree is manually curated based on general-purpose instruction-following tasks. While effective for general domains, these strategies may not be exhaustive or optimal for highly specialized domains. Extending RSDA to other domains would require domain experts to redesign specific strategy trees, which is also an area for future improvement.

Computational Overhead of Dynamic Evaluation. Unlike static rule-based filtering, RSDA requires calculating the potential entropy for every sample. Consequently, the data selection and renovation phase incurs a significantly higher computational cost compared to traditional methods. While we mitigate this overhead through a **single-pass evaluation** architecture and limit the computational cost to a fixed maximum of two inference passes per renovated sample, scaling RSDA to billion-token corpora may require optimization techniques such as distillation or sparse evaluation strategies.

Ethical considerations

The core renovation engine of RSDA relies on advanced foundation models, there is a risk that the renovation process may inadvertently amplify specific stylistic or social biases inherent in the teacher model. We therefore strongly advise employing fairness audits when deploying renovated data in sensitive domains. Furthermore, regarding data compliance, the experiments in this paper utilize open-source datasets (Alpaca-52k, Databricks-Dolly-15k) and adhere strictly to their respective licenses. Researchers adopting RSDA must similarly ensure that their source data permits modification and the creation of derivative works.

Acknowledgment

We acknowledge the support of the "Pioneer" and "Leading Goose" R&D Program of Zhejiang (No. 2026C02A1236) and the National Natural Science Foundation of China under Grant (No. 62577050).

References

Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, and 1 others. 2021. Program synthesis with large language models. *arXiv preprint arXiv:2108.07732*.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, and 1 others. 2023. Qwen technical report. *arXiv preprint arXiv:2309.16609*.

Stefan Becker, Jens Bayer, Ronny Hug, Wolfgang Huebner, and Michael Arens. 2024. Utilizing dataset affinity prediction in object detection to assess training data. In *International Conference on Robotics, Computer Vision and Intelligent Systems*, pages 260–277. Springer.

Ali Behrouz, Meisam Razaviyayn, Peilin Zhong, and Vahab Mirrokni. 2025. Nested learning: The illusion of deep learning architectures. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

Hannan Cao, Liping Yuan, Yuchen Zhang, and Hwee Tou Ng. 2023. Unsupervised grammatical error correction rivaling supervised methods. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 3072–3088.

Lichang Chen, Shiyang Li, Jun Yan, Hai Wang, Kalpa Gunaratna, Vikas Yadav, Zheng Tang, Vijay Srinivasan, Tianyi Zhou, Heng Huang, and 1 others. 2023. Alpagasus: Training a better alpaca with fewer data. *arXiv preprint arXiv:2307.08701*.

Mark Chen. 2021. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*.

Shuguang Chen, Leonardo Neves, and Thamar Solorio. 2022. Style transfer as data augmentation: A case study on named entity recognition. *arXiv preprint arXiv:2210.07916*.

Cheng-Han Chiang and Hung-yi Lee. 2023. Can large language models be an alternative to human evaluations? *arXiv preprint arXiv:2305.01937*.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, and 1 others. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Mike Conover, Matt Hayes, Ankit Mathur, Jianwei Xie, Jun Wan, Sam Shah, Ali Ghodsi, Patrick Wendell, Matei Zaharia, and Reynold Xin. 2023. Free dolly: Introducing the world's first truly open instruction-tuned llm. Databricks Blog.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. Qlora: Efficient finetuning of quantized llms. *Advances in neural information processing systems*, 36:10088–10115.

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

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, and 1 others. 2024. The llama 3 herd of models. *arXiv e-prints*, pages arXiv–2407.

Jinlan Fu, See Kiong Ng, Zhengbao Jiang, and Pengfei Liu. 2024. Gptscore: Evaluate as you desire. 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 6556–6576.

Yuan Ge, Yilun Liu, Chi Hu, Weibin Meng, Shimin Tao, Xiaofeng Zhao, Mahong Xia, Zhang Li, Boxing Chen, Hao Yang, and 1 others. 2024. Clustering and ranking: Diversity-preserved instruction selection through expert-aligned quality estimation. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 464–478.

Olga Golovneva, Moya Chen, Spencer Poff, Martin Corredor, Luke Zettlemoyer, Maryam Fazel-Zarandi, and Asli Celikyilmaz. 2022. Roscoe: A suite of metrics for scoring step-by-step reasoning. *arXiv preprint arXiv:2212.07919*.

Hanghui Guo, Jia Zhu, Shimin Di, Weijie Shi, Zhangze Chen, and Jiajie Xu. 2025. Dior: Adaptive cognitive detection and contextual retrieval optimization for dynamic retrieval-augmented generation. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2953–2975.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2020. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, and 1 others. 2022. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*.

Jiyeon Kim, Hyunji Lee, Hyowon Cho, Joel Jang, Hyeonbin Hwang, Seungpil Won, Youbin Ahn, Dohaeng Lee, and Minjoon Seo. 2024. Knowledge entropy decay during language model pretraining hinders new knowledge acquisition. *arXiv preprint arXiv:2410.01380*.

Ming Li, Yong Zhang, Zhitao Li, Jiuhai Chen, Lichang Chen, Ning Cheng, Jianzong Wang, Tianyi Zhou, and Jing Xiao. 2024. 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.

Yang Li, Mingxuan Luo, Yeyun Gong, Chen Lin, Jian Jiao, Yi Liu, and Kaili Huang. 2025. Deepthink:

Aligning language models with domain-specific user intents. *arXiv preprint arXiv:2502.05497*.

Wei Liu, Weihao Zeng, Keqing He, Yong Jiang, and Junxian He. 2023a. What makes good data for alignment? a comprehensive study of automatic data selection in instruction tuning. *arXiv preprint arXiv:2312.15685*.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. 2023b. G-eval: Nlg evaluation using gpt-4 with better human alignment. *arXiv preprint arXiv:2303.16634*.

Yilun Liu, Shimin Tao, Xiaofeng Zhao, Ming Zhu, Wenbing Ma, Junhao Zhu, Chang Su, Yutai Hou, Miao Zhang, Min Zhang, and 1 others. 2024. Coachlm: Automatic instruction revisions improve the data quality in llm instruction tuning. In *2024 IEEE 40th International Conference on Data Engineering (ICDE)*, pages 5184–5197. IEEE.

Ziche Liu, Rui Ke, Yajiao Liu, Feng Jiang, and Haizhou Li. 2025. Take the essence and discard the dross: A rethinking on data selection for fine-tuning large language models. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 6595–6611.

Thao Nguyen, Yang Li, Olga Golovneva, Luke Zettlemoyer, Sewoong Oh, Ludwig Schmidt, and Xian Li. 2025. Recycling the web: A method to enhance pre-training data quality and quantity for language models. *arXiv preprint arXiv:2506.04689*.

OpenAI. 2024. Gpt-4o system card. *Preprint*, arXiv:2410.21276.

Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, and 25 others. 2025. Qwen2.5 technical report. *Preprint*, arXiv:2412.15115.

Minju Seo, Jinheon Baek, James Thorne, and Sung Ju Hwang. 2024. Retrieval-augmented data augmentation for low-resource domain tasks. *arXiv preprint arXiv:2402.13482*.

Amrith Setlur, Saurabh Garg, Xinyang Geng, Naman Garg, Virginia Smith, and Aviral Kumar. 2024. Rl on incorrect synthetic data scales the efficiency of llm math reasoning by eight-fold. *Advances in Neural Information Processing Systems*, 37:43000–43031.

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.

Qwen Team and 1 others. 2024. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*, 2(3).

Pablo Villalobos, Jaime Sevilla, Lennart Heim, Tamay Besiroglu, Marius Hobbhahn, and Anson Ho. 2022. Will we run out of data? an analysis of the limits of scaling datasets in machine learning. *arXiv preprint arXiv:2211.04325*, 1:1.

Bo Wang, Qinyuan Cheng, Runyu Peng, Rong Bao, Peiji Li, Qipeng Guo, Linyang Li, Zhiyuan Zeng, Yunhua Zhou, and Xipeng Qiu. 2025a. Implicit reward as the bridge: A unified view of sft and dpo connections. *arXiv preprint arXiv:2507.00018*.

Xinyi Wang, Jinyi Han, Zishang Jiang, Tingyun Li, Jiaqing Liang, Sihang Jiang, Zhaoqian Dai, Shuguang Ma, Fei Yu, and Yanghua Xiao. 2025b. Hint: Helping ineffective rollouts navigate towards effectiveness. *arXiv preprint arXiv:2510.09388*.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2023. Self-instruct: Aligning language models with self-generated instructions. In *Proceedings of the 61st annual meeting of the association for computational linguistics (volume 1: long papers)*, pages 13484–13508.

Yufei Wang, Can Xu, Qingfeng Sun, Huang Hu, Chongyang Tao, Xiubo Geng, and Daxin Jiang. 2022. Promda: Prompt-based data augmentation for low-resource nlu tasks. *arXiv preprint arXiv:2202.12499*.

Yongliang Wu, Yizhou Zhou, Zhou Ziheng, Yingzhe Peng, Xinyu Ye, Xinting Hu, Wenbo Zhu, Lu Qi, Ming-Hsuan Yang, and Xu Yang. 2025. On the generalization of sft: A reinforcement learning perspective with reward rectification. *arXiv preprint arXiv:2508.05629*.

Can Xu, Qingfeng Sun, Kai Zheng, Xiubo Geng, Pu Zhao, Jiazhan Feng, Chongyang Tao, and Daxin Jiang. 2023. Wizardlm: Empowering large language models to follow complex instructions. *arXiv preprint arXiv:2304.12244*.

Can Xu, Qingfeng Sun, Kai Zheng, Xiubo Geng, Pu Zhao, Jiazhan Feng, Chongyang Tao, Qingwei Lin, and Daxin Jiang. 2024a. Wizardlm: Empowering large pre-trained language models to follow complex instructions. In *The Twelfth International Conference on Learning Representations*.

Ning Xu, Zhaoyang Zhang, Lei Qi, Wensuo Wang, Chao Zhang, Zihao Ren, Huaiyuan Zhang, Xin Cheng, Yanqi Zhang, Zhichao Liu, and 1 others. 2024b. Chipexpert: The open-source integrated-circuit-design-specific large language model. *arXiv preprint arXiv:2408.00804*.

Zhangchen Xu, Fengqing Jiang, Luyao Niu, Yuntian Deng, Radha Poovendran, Yejin Choi, and Bill Yuchen Lin. 2024c. Magpie: Alignment data synthesis from scratch by prompting aligned llms with nothing. *arXiv preprint arXiv:2406.08464*.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang

Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Weihao Yu, Zhengyuan Yang, Linjie Li, Jianfeng Wang, Kevin Lin, Zicheng Liu, Xinchao Wang, and Lijuan Wang. 2023. Mm-vet: Evaluating large multimodal models for integrated capabilities. *arXiv preprint arXiv:2308.02490*.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. Hellaswag: Can a machine really finish your sentence? *arXiv preprint arXiv:1905.07830*.

Wenxuan Zhang, Yue Deng, Bing Liu, Sinno Pan, and Lidong Bing. 2024. Sentiment analysis in the era of large language models: A reality check. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 3881–3906.

Chunting Zhou, Pengfei Liu, Puxin Xu, Srinivasan Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, Lili Yu, and 1 others. 2023a. Lima: Less is more for alignment. *Advances in Neural Information Processing Systems*, 36:55006–55021.

Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. 2023b. Instruction-following evaluation for large language models. *arXiv preprint arXiv:2311.07911*.

Wenqiao Zhu, Ji Liu, Rongjunchen Zhang, Haipang Wu, and Yulun Zhang. 2025. Carft: Boosting llm reasoning via contrastive learning with annotated chain-of-thought-based reinforced fine-tuning. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 5933–5948.

Wenhao Zhuang, Dawa Cairen, and Yuan Sun. 2024. Tifd: Tibetan instruction-following dataset for large language models supervised fine-tuning. *Data Intelligence*.

A Renovation Strategy Design and Pilot Experiments

To scientifically determine the optimal renovation strategies for the RSDA framework and to validate the concept of **Data Affinity**, we conducted a pilot study. We hypothesized that a model's responsiveness to data modifications (**Instruction**, **Input**, **Output**) is not uniform but depends on its intrinsic capabilities.

A.1 Renovation Case Studies

Table 3: Hyperparameter settings for QLoRA fine-tuning on QwenX-7B.

Hyperparameter	**Value**
Backbone Model	QwenX-7B
Quantization	4-bit
PEFT Method	LoRA
LoRA Rank (r)	16
LoRA Target Modules	All Linear Layers
Optimizer	AdamW
Learning Rate	3×10^{-5}
LR Scheduler	Cosine
Warmup Ratio	0.15
Weight Decay	0.05
Epochs	6
Per-device Batch Size	2
Gradient Accumulation Steps	32
Effective Batch Size	32
Max Gradient Norm	0.5

Figure 6 visually demonstrates the transformation of a single raw data sample under distinct component-specific renovation strategies. This visualization substantiates that our interventions induce significant semantic and structural shifts, extending beyond superficial paraphrasing. To isolate the impact of each strategy during this pilot phase, we adopted a modular approach: specific components were selectively modified while keeping the remaining context invariant, allowing for a precise assessment of each strategy's contribution.

A.2 Experimental Setup for Pilot Study

We randomly sampled 1,000 instances from the **GSM8K** training set as the seed corpus. We applied 8 distinct rewriting strategies to this seed data. To rigorously observe the evolution of model preferences, we selected the Qwen family as our testbed, specifically employing **Qwen1.5-7B**, **Qwen2-7B**, and **Qwen2.5-7B**. We fixed the parameter scale at **7B** across all three versions. This design eliminates model capacity as a confounding variable, ensuring that any observed divergence in data preference is attributable solely to their distinct stages of **alignment maturity** rather than differences in computational power.

Table 3 details the hyperparameters used for fine-tuning the Qwen models during the pilot study and the main experiments. We denote model as QwenX-7B, where X represents the version number.

To robustly measure mathematical reasoning ability and minimize variance from zero-shot generation during the evaluation phase, we adopted a **5-shot** setting for all inferences. This configuration ensures that performance differences reflect the model's absorption of updated training patterns rather than sensitivity to prompt variations.

A.3 Results and the Data Affinity Hypothesis

Table 4: Data renovation strategies. It is crucial to note that **Domain Migration** fundamentally alter the problem context. Consequently, to preserve semantic consistency, rewriting necessitates synchronous modifications across both the instruction and output components. The gray-painting strategy involves renovation.

Category	Method	Strategy Description
Raw	–	No training
Unmodified	D0	Full dataset
	D1	1K raw samples
Instruction	D2	Positive emotions
	D3	Negative emotions
Input	D4	Storytelling background
	D5	Logical background
	D6	Domain migration
Output	D7	Diverse answers
	D8	Information concentration
	D9	Semantic filling
	D10	Explicit reasoning

Table 5 presents the comparative evaluation of different renovation strategies across the Qwen family, with the strategy information corresponding to the "method" column in Table 4. The empirical results reveal distinct preference patterns that strongly validate our **Data Affinity** hypothesis.

All experiments are conducted on an NVIDIA A6000 GPU.

Sensitivity to Input Contextualization. We observe a negative correlation between model capability and the benefit derived from Input renovation. For the less mature **Qwen1.5-7B**, Strategy D6 yields a notable improvement, boosting accuracy from the baseline 0.5671 to 0.5792. This suggests that weaker models struggle with sparse inputs and benefit significantly from semantic completion. In contrast, the **Qwen2.5-7B** shows negligible gain from the same strategy ($0.8158 \to 0.8172$). This indicates that advanced models possess higher robustness to input ambiguity, rendering input-level renovation less critical for them.

Divergence in Output Strategy Preference. While Output renovation (Strategies D7-D10) universally enhances performance across all models—confirming that explicit reasoning traces are the dominant factor for mathematical tasks—the specific *optimal strategy* varies.**Qwen1.5-7B** achieves its peak performance with Strategy **D10** (0.6302), whereas the more capable **Qwen2.5-7B** favors Strategy **D9**, reaching a state-of-the-art accuracy of 0.8605. This divergence implies that "high-quality data" is not a static standard; rather, the optimal logical structure (e.g., D9 vs D10) must align with the model's specific cognitive manifold.

Inefficacy of Superficial Instruction Tuning. Interestingly, strategies targeting solely the Instruction component (D2, D3) generally failed to outperform the raw baseline, and in some cases (e.g., Qwen2.5 with D2), even caused performance regression ($0.8158 \to 0.7900$). This highlights that for reasoning-intensive tasks like GSM8K, superficial changes to tone or role-prompting are insufficient and may even introduce semantic noise compared to substantive logic renovation.

These phenomena indicate that "high-quality data" is not static; it depends on the model's stage. We refer to this dynamic relationship as **data affinity**. RSDA aims to maximize this affinity by selecting strategies that match the data's potential.

B Hyperparameter Sensitivity Analysis Experiment

B.1 Ablation on Fusion Coefficients

The Potential Entropy $\mathcal{E}_{pot}$ is a composite metric derived from internal epistemic uncertainty (α) and external structural deficiency (β). To determine the optimal trade-off between exploring knowledge boundaries and exploiting structural gaps, we performed a grid search on the coefficients under the constraint $\alpha + \beta = 1$. We evaluated four configurations: $\alpha/\beta \in \{0.3/0.7, 0.4/0.6, 0.5/0.5, 0.6/0.4\}$.

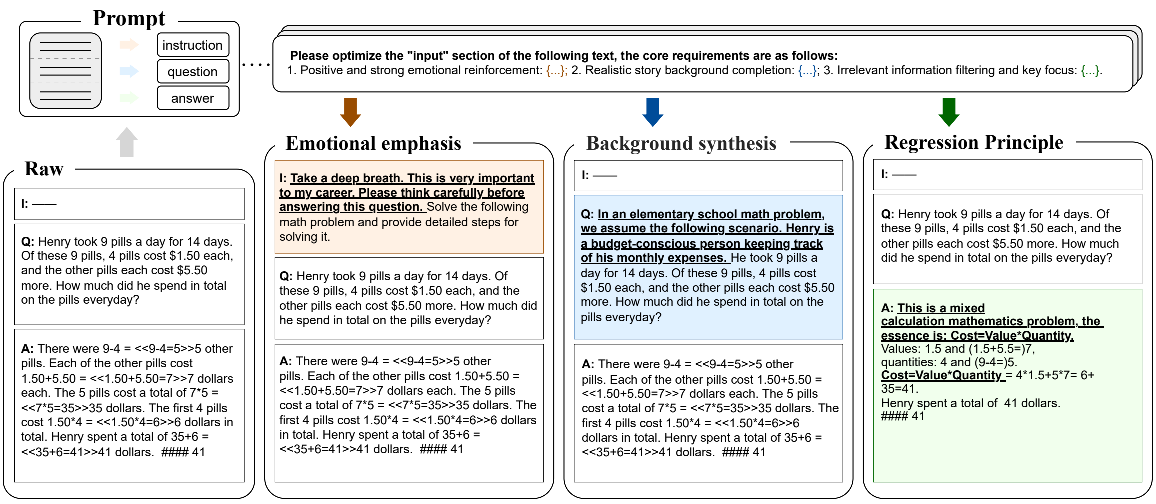

Figure 6: Component-specific renovation strategies applied to a raw seed sample. The diagram illustrates three distinct intervention paths targeting different data modules: (1) **Emotional Emphasis** modifies the Instruction to enhance engagement; (2) **Background Synthesis** enriches the Input/Question with realistic context; and (3) **Regression Principle** restructures the Output/Answer to clarify logical steps. **Note:** Only the targeted module is renovated in each path to isolate strategy effects.

All ablation studies utilized the Meta-Llama-3-8B backbone, fine-tuned on on datasets reconstructed using specific fusion coefficients applied to a 1k sample subset randomly drawn from Alpaca. We assessed performance across five key dimensions: mathematical reasoning GSM8K, MATH and ARC-Challenge, comprehensive knowledge MMLU, code generation HumanEval, and instruction following IFEval. Table 6 summarizes the results across five diverse benchmarks. We observe that:

Dominance of Structural Signal ($\beta > \alpha$)**:** Configurations where $\beta \geq 0.5$ consistently outperform those dominated by α. This suggests that for data renovation, identifying *structural flaws* (e.g., lack of reasoning steps or poor formatting) is a more reliable indicator of "renovation worthiness" than raw model uncertainty, which can sometimes be confounded by irreducible noise.

The Optimal Balance (0.4 / 0.6): The configuration $\alpha = 0.4, \beta = 0.6$ achieves the highest average score (Avg. **41.28**). While the purely structure-heavy setting (0.3/0.7) performs well on instruction following (IFEval), it suffers a slight regression in reasoning tasks (GSM8K). This indicates that retaining a significant weight for epistemic uncertainty ($\alpha = 0.4$) is crucial for capturing "hard samples" that are structurally sound but conceptually challenging for the model.

Diminishing Returns of Uncertainty: As α increases to 0.6, we observe a notable drop in HumanEval and IFEval scores. This implies that prioritizing high-entropy samples without considering their structural quality may introduce noise or difficult-but-unfixable data into the renovation pipeline.

Based on these findings, we selected $\alpha = 0.4$ and $\beta = 0.6$ as the default hyperparameters for the RSDA framework. Moreover, for the calculation of Model Predictive Entropy, we employed a locally deployed **Qwen3-0.6B** model to compute token-wise probability distributions.

B.2 Model Strategy Score Distribution

To determine our component-specific threshold settings, we visualized the distribution of strategy preference scores across two distinct instruction-tuning datasets: **Alpaca-52k** (synthetic) and **Databricks-Dolly-15k** (human-authored). We randomly sampled 1K instances from each dataset and plotted the policy scores generated by the GPT-4o model for these data points, as shown in Figure 7 8.

Both datasets exhibit a characteristic "low mean, high variance" pattern in the Input component. This is primarily due to many samples have empty inputs (Null), resulting in zero-scores that drag down the mean, while complex inputs score highly. This bimodal distribution necessitates the stricter sensitivity threshold we need to avoid false positives on empty inputs. Additionally, across both datasets, instruction scores consistently clustered at higher

Table 5: Performance comparison across different methods and model versions. Raw denotes the baseline without any modifications. D0-D10 represent different modification strategies categorized by their target components. The strategy yielding the optimal outcome is **marked in gray**, with the data **marked in red** represents the baseline for modification.

Category	Method	Qwen1.5-7B			Qwen2-7B			Qwen2.5-7B		
		Acc	stderr	Loss	Acc	stderr	Loss	Acc	stderr	Loss
Raw	–	0.5671	0.0136	–	0.7786	0.0114	–	0.8158	0.0107	–
Unmodified	D0	0.4958	0.0138	0.0518	0.7368	0.0124	0.0711	0.7672	0.0116	0.2312
	D1	0.5057	0.0138	0.1703	0.7400	0.0121	0.1926	0.7839	0.0113	0.4095
Instruction	**D2**	**0.5295**	**0.0137**	**0.1846**	**0.7407**	**0.0121**	**0.2308**	**0.7900**	**0.0112**	**0.4271**
	D3	0.4488	0.0137	0.1194	0.7248	0.0123	0.1438	0.7748	0.0115	0.3095
Input	D4	0.4943	0.0138	0.1504	0.7513	0.0119	0.1791	0.8023	0.0112	0.3459
	D5	0.3548	0.0132	0.1334	0.7202	0.0124	0.1578	0.7453	0.0120	0.3327
	D6	**0.5792**	**0.0136**	**0.3313**	**0.7642**	**0.0117**	**0.3357**	**0.8172**	**0.0106**	**0.4672**
Output	D7	0.5557	0.0137	0.6882	0.7801	0.0114	0.6396	0.8173	0.0106	0.7248
	D8	0.5830	0.0136	0.5758	0.7810	0.0116	0.5442	0.8234	0.0105	0.6335
	D9	0.6133	0.0134	0.3811	0.8014	0.0110	0.3540	0.8605	0.0095	0.4207
	D10	**0.6302**	**0.0133**	**0.2512**	**0.8036**	**0.0109**	**0.2412**	**0.8560**	**0.0097**	**0.3095**

Table 6: Ablation results of fusion coefficients α (Internal Potential) and β (External Potential) on Meta-Llama-3-8B. The setting $\alpha = 0.4, \beta = 0.6$ yields the best comprehensive performance. To reduce costs, we utilized the GPT-4o model when renovating these data. For all evaluation steps, inference was conducted with a batch size of 1.

Coefficients		Benchmarks						
α (internal)	β (external)	**GSM8K**	**MMLU**	**HumanEval**	**IFEval**	**ARC-challenge**	**MATH**	**Avg.**
0.3	0.7	51.40	**65.09**	38.4	9.24	55.89	20.24	40.04
0.4	**0.6**	**52.58**	65.06	38.2	**10.75**	**59.86**	**21.22**	**41.28**
0.5	0.5	50.57	64.98	36.6	9.61	56.14	20.42	38.72
0.6	0.4	51.78	64.30	**39.6**	9.06	55.89	20.5	40.18

levels, while output scores exhibited the most dispersed distribution. This confirms that response quality remains the primary bottleneck and the area most worthy of improvement.

B.3 Potential Entropy Parameter Settings

B.3.1 Impact of Component Weights

By analyzing the results of different strategies in the GSM8K ablation experiments and examining the distribution of model strategy preference scores across 1K random data points on two datasets, we determined the parameter settings for this section. To mitigate the saturation effect of the instruction component and prioritize the high-impact output component, we set the weights to $\gamma_{ins} = 0.15, \gamma_{inp} = 0.35$, and $\gamma_{out} = 0.5$.

B.3.2 Renovation Threshold Sensitivity

These thresholds stratify the corpus into noise, renovation candidates, and reserve data, balancing the trade-off between renovation cost and performance gain. To filter out unrecoverable noise, we set the upper bound τ to the **90th percentile** of the distribution, directly discarding the top 10% of samples with excessive entropy. Conversely, regarding the renovation floor, we set τ_1 to the **20th percentile**. Samples falling below τ_1 undergo a secondary quality evaluation, where only the top 50% are preserved to maintain the foundational distribution, while the rest are discarded. Consequently, this configuration ensures that the middle 70% of samples enter the renovation pipeline.

B.3.3 Balancing Internal and External Potential

To regulate the contribution of each factor within a normalized scope, we impose the constraint $\alpha + \beta = 1$. Our goal is to prioritize the identification of tangible structural flaws over pure epistemic uncertainty. Following ablation studies, we

Table 7: Hyperparameter settings used in the calculation of Potential Entropy and Renovation Decisions.

Category	Hyperparameter	Value
Metric Fusion	Internal Potential Coeff. (α)	0.4
	External Potential Coeff. (β)	0.6
Component Weights	Instruction Weight (γ_{ins})	0.15
	Input Weight (γ_{inp})	0.35
	Output Weight (γ_{out})	0.50
Thresholds	Noise Cutoff (τ)	90th Percentile
	Renovation Floor (τ_1)	20th Percentile
	Reserve Quality Filter (τ_2)	Median of Retain
	Sensitivity Threshold (δ_{ins})	0.10
	Sensitivity Threshold (δ_{inp})	0.12
	Sensitivity Threshold (δ_{out})	0.10

Table 8: The RSDA Renovation Strategy Library. Each index corresponds to a specific semantic or structural transformation directive used by the Data Renovator.

Component	Index	Strategy Name	Description
$\mathbb{S}_{ins}$	0	Identity	Retain the original instruction to preserve intent.
	1	Positive Sentiment Injection	Rewrite with affirmative and encouraging tones to enhance user engagement.
$\mathbb{S}_{inp}$	0	Identity	Retain the original context.
	1	Narrative Contextualization	Transform abstract inputs into story-driven, real-world scenarios.
	2	Domain Transfer	Migrate the problem context to a different domain (e.g., medical/legal) to improve generalization.
$\mathbb{S}_{out}$	0	Explicit Reasoning[†]	Rewrite to include Chain-of-Thought (CoT) processes.
	1	Solution Diversification	Generate multiple valid perspectives or solution paths.
	2	Info. Density Concentration	Condense the response to maximize information entropy per token.
	3	Semantic Expansion	Enrich the output with detailed explanations and background knowledge.

[†] *Note: Explicit Reasoning serves as the default baseline strategy for the Output component when no other specific deficiency is detected.*

established the fusion coefficients as $\alpha = 0.4$ and $\beta = 0.6$, thereby assigning a higher priority to identifying data with explicit strategic weaknesses.

B.3.4 Sensitivity Threshold

Finally, to prevent over-editing, we established the sensitivity thresholds. For the ***input*** component, which exhibits high variance, we apply a stricter threshold to avoid false positives derived from noise. In contrast, for the saturated ***instruction*** and ***output*** components, we adopt a standard threshold to capture fine-grained improvements. For the input component, the prevalence of empty instances results in skewed statistics (low mean and high variance). To address this, we calibrated $\delta_{inp} = 0.12$ by calculating the average score of exclusively non-zero samples, ensuring the threshold targets valid semantic content. In contrast, for the more stable *instruction* and *output* components, we set $\delta_{ins/out} = 0.1$.

To facilitate future work, table 7 lists all hyperparameters used in the RSDA framework, including metric fusion, selection thresholds, etc.

C The Reusability Value of Low-quality Data

To confirm the reuse value of outdated data, we conducted a renovation experiment using the Dolly dataset and the Meta-Llama3-8B model, and the results are shown in the Table 9. The experimental results demonstrate that even the 1,000 data points initially rated as the worst quality can still generate value after renovation, even surpassing the capabilities of the 9,000 selected top-performing data.

Table 9: The reusability value of low-quality data

dataset	Size	MMLU	ARC-C	MBPP	Hellaswag	Gsm8k	IFeval	Humaneval
NULL	-	65.43	58.19	61.9	81.44	50.27	9.98	37.2
Baddest Sample	1,000	64.46	57.94	49.0	81.24	50.42	10.34	39.0
Random Selection	1,000	65.17	59.98	57.0	82.17	50.87	10.54	42.7
AlpaGasus	9,000	65.33	59.64	52.7	82.14	51.62	10.91	43.7
RSDA	$\sim$800	65.84	60.15	50.6	82.83	52.33	10.28	41.6

D Mark Strategy Library

The RSDA framework employs a deterministic mapping between integer strategy indices and specific rewriting directives.

Table 8 enumerates the complete library of 8 renovation strategies. As detailed in Section 4.1, the indexing is asymmetric: for **Instruction** and **Input**, Index 0 denotes **Identity**, whereas for **Output**, Index 0 defaults to **Explicit Reasoning**, reflecting the universal benefit of logical depth.

E Detailed Analysis of Main Results on Alpaca-52k

This section presents supplementary details and evaluation results regarding the fine-tuning of the larger Qwen2.5-32B-Instruct model. Additionally, we provide a granular analysis of the experimental results presented in Table 2, comparing RSDA against the data selection and evolution strategies across two distinct backbone models: Qwen2.5-7B and Meta-Llama-3-8B.

E.1 Superiority in Reasoning and Common Sense

The most significant observation is RSDA's dominance in tasks requiring deep reasoning and knowledge application.

- Mathematical Reasoning (GSM8K): On Qwen2.5-7B, RSDA achieves an accuracy of **83.85**, surpassing the full dataset baseline (76.35) by a significant margin of +7.5%, and outperforming the strong filtering baseline AlpaGasus (81.80). Similarly, on Meta-Llama-3-8B, RSDA achieves **52.62**, effectively reversing the negative transfer observed in the full dataset (43.29).

- Code Generation (HumanEval & MBPP): RSDA demonstrates robust performance in coding tasks. Notably, on HumanEval with Qwen2.5-7B, RSDA reaches **78.0**, exceeding both LIMA (77.4) and Deita (76.2).

- Common Sense (MMLU & Hellaswag): Our method consistently achieves the highest scores across both backbones. For instance, on MMLU (Meta-Llama-3-8B), RSDA scores **65.51**, surpassing the closest competitor LIMA (65.35) and significantly outperforming the Full dataset (63.15).

E.2 Data Efficiency and the LIMA Paradigm

RSDA achieves these results using only $\sim$800 samples, which constitutes approximately 1.5% of the full Alpaca-52k dataset and less than 10% of the data used by AlpaGasus (9,000 samples). This extreme data efficiency validates our core hypothesis: the bottleneck in current instruction tuning is not the quantity of data, but the *affinity* and *potential* of the samples. By surgically renovating only the high-potential instances, RSDA maximizes the information density per sample.

E.3 Mitigating Negative Transfer in Stale Data

A critical insight from Table 2 is the phenomenon of negative transfer in the "Alpaca-52k" baseline.

- On Meta-Llama-3-8B, training on the full stale dataset causes a performance regression in GSM8K ($50.27 \rightarrow 43.29$) and MBPP ($61.9 \rightarrow 47.9$) compared to the base model (NULL).

- In contrast, RSDA not only prevents this degradation but induces positive transfer ($50.27 \rightarrow 52.62$ in GSM8K). This confirms that our Potential Entropy metric effectively filters out the "toxic" noise that confuses modern LLMs, while the renovation process restores the utility of the remaining data.

E.4 Instruction Following (IFEval) Trade-offs

We observe that RSDA scores lower on IFEval compared to the Full dataset. This is an expected trade-off stemming from data scale: IFEval rigorously tests adherence to rigid formatting constraints (e.g., "do not use capital letters"). Acquiring this superficial compliance relies heavily on

Table 10: Diagnostic dimensions and corresponding renovation strategies.

Diagnostic Dimension	Instruction Tone	Input Depth	Input Complexity	Output CoT	Output Diversity	Output Density	Output Background
Renovation Strategy	Positive Sentiment Injection	Narrative Contextualization	Domain Transfer	Explicit Reasoning	Solution Diversification	Info. Density Concentration	Semantic Expansion

Table 11: Comparison of different data synthesis methods fine-tuned on **Qwen2.5-32B-Instruct** across multiple benchmarks. GSM8K reports 5-shot strict exact match. Best results in each column are **bold**.

Method	**Size**	**ARC-C (%)**	**HumanEval (pass@1)**	**MMLU (%)**	**HellaSwag (%)**	**GSM8K (%)**
No fine-tuning	–	72.27	–	83.19	85.16	75.28
Random Sample	1000	72.61	70.1	83.30	85.06	79.30
Alpagasus (selection)	9000	72.53	65.5	83.21	84.57	**84.22**
CoachLM (evolution)	1000	72.78	70.7	83.65	84.81	78.70
RSDA (ours)	∼800	**72.93**	**71.5**	**84.39**	**85.33**	80.48

exposing the model to a massive, diverse set of format-constrained samples during fine-tuning. By drastically pruning the dataset to ∼800 samples, we inevitably sacrifice the long-tail diversity of these formatting patterns.

Notably, this performance drop is not unique to RSDA. CoachLM, another typical LLM-based augmentation method, suffers a similarly severe decline on IFEval after data reduction (dropping to 11.09% on Llama-3-8B, slightly lower than RSDA's 11.28%). We argue this trade-off is highly acceptable for researchers prioritizing core model capabilities over strict formatting compliance.

F Diagnostic Dimension to Renovation Strategy Mapping

The RSDA framework operates through a two-layer pipeline in which diagnostic dimensions and renovation strategies are closely coupled. As clarified in the main article, the diagnostic dimensions serve as the basis for the model-as-judge to identify lateral defects in the original seed data, while the renovation strategies represent the concrete surgical actions executed by the second layer to improve data affinity.

Table 10 presents the complete mapping between each diagnostic dimension and its corresponding renovation strategy. Specifically, when the model detects a defect along a given diagnostic dimension, it triggers the associated renovation strategy to address that defect. For example, a low score on **Instruction Tone** triggers **Positive Sentiment Injection** to rewrite instructions with a more constructive and engaging tone; a deficiency in **Input Depth** triggers **Narrative Contextualization** to enrich the input with background context; and a weakness in **Output CoT** triggers **Explicit Reasoning** to elicit step-by-step chain-of-thought responses.

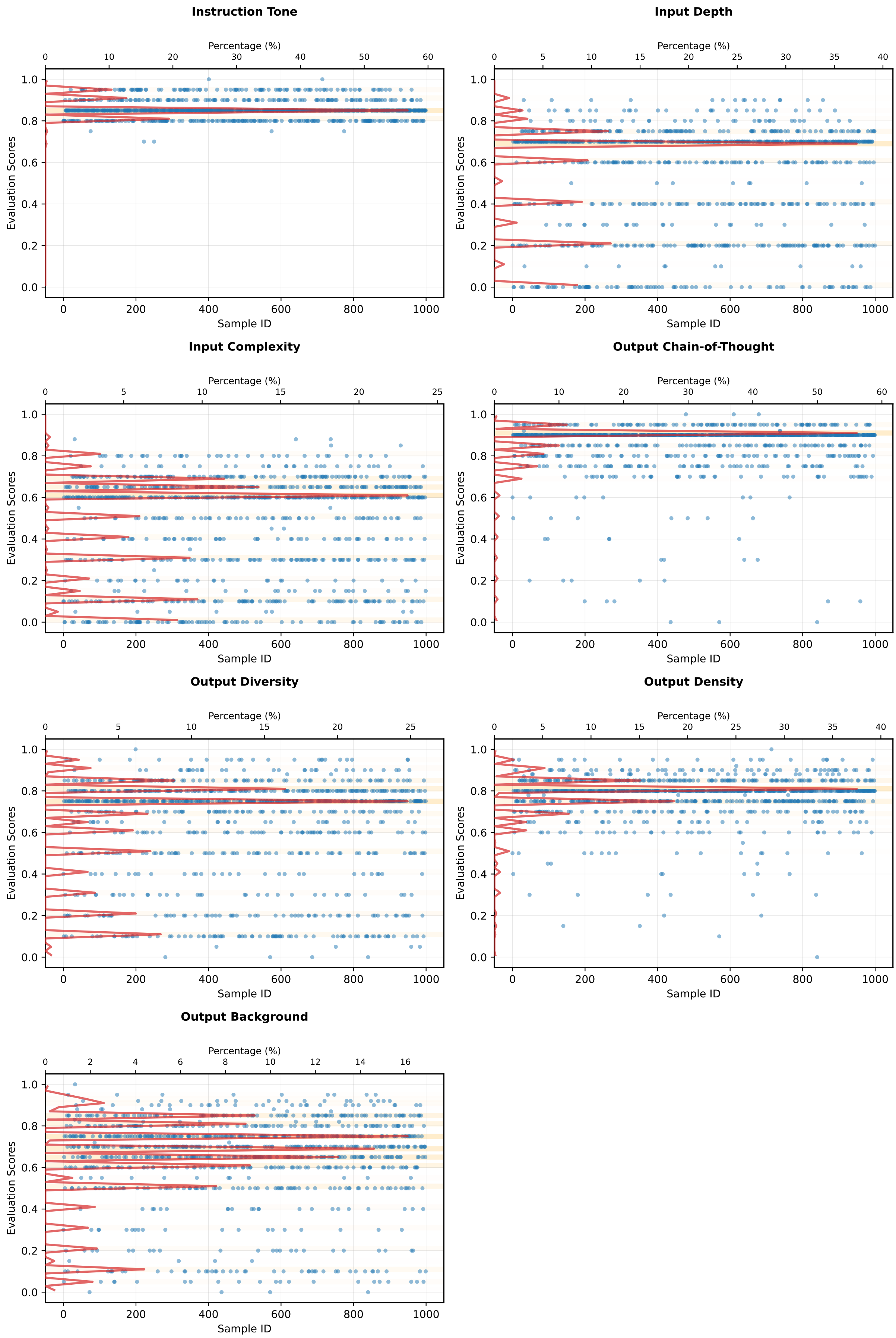

Figure 7: Scatter plots of strategy evaluation scores across **1,000 random Alpaca samples**. The X-axis represents sample ID, the Y-axis represents strategy score, points indicate individual data scores, and the line chart shows the volume of data at each score.

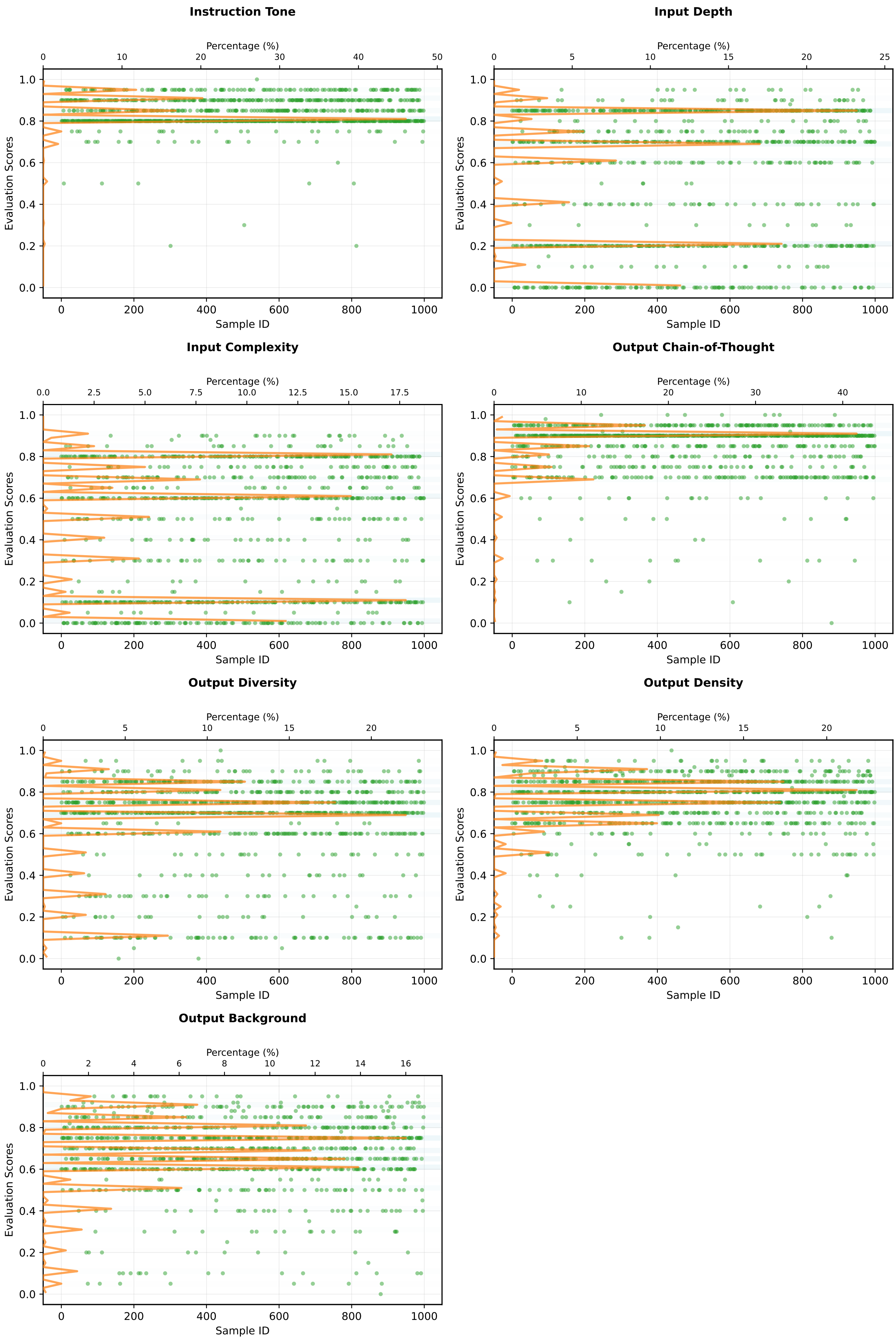

Figure 8: Scatter plots of strategy evaluation scores across **1,000 random Dolly samples**. The X-axis represents sample ID, the Y-axis represents strategy score, points indicate individual data scores, and the line chart shows the volume of data at each score.

G Algorithm Pseudocode

In this section, we provide the formal procedural descriptions of the RSDA framework. Algorithm 1 outlines the global data evolution pipeline, detailing the stratification of the raw corpus into **Renovate**, **Reserve**, and **Discard** subsets based on Potential Entropy $\mathcal{E}_{pot}$ and Quality Scores $\mathbf{q}$. Algorithm 2 elaborates on the **Mark-Driven Renovation** mechanism, demonstrating how specific renovation strategies are triggered by component-wise deficiency gaps.

G.1 RSDA Data Evolution Pipeline

Algorithm 1: RSDA Evolution Pipeline

Input: Raw dataset $\mathcal{D} = \{x_i\}_{i=1}^{N}$, model $\mathcal{M}$, parameters α, β, δ
Output: Evolved dataset $\mathcal{D}_{\text{final}}$

`// Step 1: Metric Calculation across Corpus`
$\mathcal{D}_{\text{final}} \leftarrow \emptyset$
$\mathcal{S} \leftarrow [\,]$ `// score array`
foreach $x_i \in \mathcal{D}$ **do**
  $\mathcal{H}_{\text{pred}}^{(i)} \leftarrow \text{COMPUTEUNCERTAINTY}(\mathcal{M}, x_i)$
  $\mathcal{S}_{\text{gap}}^{(i)} \leftarrow \text{COMPUTESTRATEGYGAP}(\mathcal{M}, x_i)$
  $\mathcal{E}_{\text{pot}}^{(i)} \leftarrow \alpha \cdot \mathcal{N}(\mathcal{H}_{\text{pred}}^{(i)}) + \beta \cdot \mathcal{N}(\mathcal{S}_{\text{gap}}^{(i)})$
  Append $\mathcal{E}_{\text{pot}}^{(i)}$ to $\mathcal{S}$
end
`// Step 2: Threshold Determination`
$\tau_{\max} \leftarrow \text{PERCENTILE}(\mathcal{S}, 90)$ `// noise cutoff`
$\tau_1 \leftarrow \text{PERCENTILE}(\mathcal{S}, 20)$ `// renovation floor`
`// Step 3: Stratification and Processing`
foreach $x_i \in \mathcal{D}$ **do**
  if $\tau_1 \leq \mathcal{E}_{pot}^{(i)} < \tau_{\max}$ **then**
    `// Zone of Proximal Development → Renovate`
    $x_{\text{new}} \leftarrow \text{MARKDRIVENRENOVATION}(x_i, \mathcal{S}_{\text{gap}}^{(i)}, \delta)$
    Add x_{new} to $\mathcal{D}_{\text{final}}$
  else if $\mathcal{E}_{pot}^{(i)} < \tau_1$ **then**
    `// Low Potential → Check Quality for Reserve`
    $q_i \leftarrow \sum \text{COMPONENTSCORES}(x_i)$
    $\tau_2 \leftarrow \text{MEDIAN}(\{q_j : \mathcal{E}_{\text{pot}}^{(j)} < \tau_1\})$
    if $q_i \geq \tau_2$ **then**
      Add x_i to $\mathcal{D}_{\text{final}}$
    end
  else
    `// High Entropy Noise or Low Quality → Discard`
    continue
  end
end
return $\mathcal{D}_{final}$

The algorithm outlines the global data stratification workflow. It first computes the Potential Entropy $\mathcal{E}_{pot}$ for the entire corpus to quantify evolutionary value. Based on dynamic thresholds derived from the score distribution, the data is partitioned into three streams: samples within the "Zone of Proximal Development" enter the renovation pipeline; high-fidelity samples with low potential are routed to the Reserve set subject to a secondary quality filter; and the remaining noise is discarded.

G.2 Renovation Strategy Mark Generation

The algorithm details the execution of the Mark-Driven Renovation. It iterates through the sample components, triggering targeted rewriting strategies only when a component's deficiency gap g_k exceeds the sensitivity threshold δ_k. To preserve semantic coherence, the algorithm includes a logical dependency check: if the *Input* context is modified, the *Output* is mandatorily recomputed to align with the new problem formulation.

Algorithm 2: Strategy Mark Generation

Require: Sample $x = \{x_{\text{ins}}, x_{\text{inp}}, x_{\text{out}}\}$; Strategy sets $\mathbb{S} = \{\mathbb{S}_{\text{ins}}, \mathbb{S}_{\text{inp}}, \mathbb{S}_{\text{out}}\}$; Thresholds $\delta = \{\delta_{\text{ins}}, \delta_{\text{inp}}, \delta_{\text{out}}\}$
Ensure: Mark vector $\mathcal{M}(x) = [m_{\text{ins}}, m_{\text{inp}}, m_{\text{out}}]$

$\mathcal{M} \leftarrow [\,]$
foreach $k \in \{ins, inp, out\}$ **do**
  $g_{\max} \leftarrow 0, \quad j_{\text{best}} \leftarrow 0$
  `// Evaluate active strategies`
  foreach $j \in \mathbb{S}_k \setminus \{0\}$ **do**
    $s_{k,j} \leftarrow \text{LLM-JUDGE}(x_k, \text{Prompt}_j)$
    $g_{k,j} \leftarrow 1 - s_{k,j}$
    if $g_{k,j} > g_{\max}$ **then**
      $g_{\max} \leftarrow g_{k,j}, \quad j_{\text{best}} \leftarrow j$
    end
  end
  `// Apply threshold`
  if $g_{\max} > \delta_k$ **then**
    $m_k \leftarrow j_{\text{best}}$
  else
    $m_k \leftarrow 0$ `// baseline strategy`
  end
  Append m_k to $\mathcal{M}$
end
return $\mathcal{M}$

H Renovation Prompt Templates

The following section demonstrates the meta-prompts we designed.

Prompt1 for Positive Emotion Injection

 # Role
You are a data augmentation expert proficient in Natural Language Processing and Affective Computing. Your expertise lies in transforming neutral or objective declarative text into text with "positive strong emotion" or "positive pragmatic presuppositions", while strictly preserving the original logic, numerical values, and data structure.
 # Task
Please read the input JSON data and rewrite the "instruction" field. The goal of the rewriting is to introduce a strong emotional tone of positive, optimistic, affirmative, or emphasizing "cost-effectiveness/efficiency" by adding modifiers, adverbs, or adjusting sentence structure, while maintaining the absolute accuracy of the original mathematical problem or logical description.
 # Guidelines
In this task, "positive strong emotion" includes, but is not limited to:
1. Positive Presuppositions: Use words like "only", "just", "must" to emphasize low cost or minimal impact.
2. Positive Adjectives/Adverbs: Add "successfully", "efficiently", "luckily", "excitingly".
3. Affirmative Tone: Change passive statements to descriptions of active control, highlighting the protagonist's agency.
 # Constraints
1. Absolutely No Modification of Numerical Values: All numbers and units must be completely consistent with the original text.
2. Maintain Original Logic: The core relationships of the problem must not change.
3. Maintain JSON Format: The output must be in valid JSON format; do not change the key names.
4. Language Consistency: If the original text is in English, please rewrite it in English; if it is in Chinese, rewrite it in Chinese.
 # Note
If the original data does not have an instruction section, fill it in according to the input section.

Prompt2 for Negative Emotion Injection

 # Role

You are an expert specializing in instruction compliance and adversarial data augmentation. Your expertise lies in transforming ordinary instruction text into text imbued with "negative emotion" or "strictly defensive expressions." You excel at using negative constraints to strengthen the binding force of instructions.

 # Task

Read the input JSON data and rewrite the "instruction" field.
The goal of the rewrite is to create a sense of urgency and seriousness, "must be done," "absolutely cannot be wrong," and "failure is strictly forbidden" by adding strong negative modifiers, warnings, or prohibitions before or after the instruction, while preserving the core intent of the original instruction.

 # Guidelines

In this task, "negative emotion" does not refer to pessimistic emotions, but rather to strong instructions based on negation. Please refer to the following keywords and sentence structures:
1. Core Vocabulary: "cannot fail to", "never miss", "avoid", "do not forget", "no excuse", "prohibit".
2. Sentence Structure:
- "Under no circumstances should you..."
- "It is strictly forbidden to..."
- "You can't make mistakes."
3. Tone: Serious, warning, high-pressure, and unquestionable.

 # Constraints

1. Core Instructions Retained: The original task required by the instruction (e.g., "solve a math problem", "translate a sentence") must be retained and cannot be lost due to the addition of modifiers.
2. Data Format Preserved: The output must be in valid JSON format, and the key names must not be modified.
3. Value Preservation: If the original instruction contains specific values, they must be strictly preserved.
4. Language Consistency: If the original text is in English, please rewrite it in English; if it is in Chinese, please rewrite it in Chinese.

 # Note

If the original data does not have an instruction section, fill it in according to the input section.

Prompt3 for Storytelling Background

 # Role

You are a skilled storyteller and educator, as well as a keen observer of real life. Your expertise lies in imbuing dry mathematical or logic problems with rich "real-world context" and "character motivations". You possess the keen ability to capture the real-world value behind numbers, such as budget control, time management, health monitoring, and business decisions, and transform it into a compelling backstory.

 # Task

Read the input JSON data and rewrite the "input" field.
The goal of the rewrite is to add a short, introductory story at the beginning, while keeping the original problem description completely unchanged. This backstory must:
1. Introduce Real-World Value: Explain why this calculation is important, for example: to save money, to save time, or to ensure the success of the experiment.
2. Shape Characters/Context: Introduce a specific protagonist or scenario, such as: "a thrifty housewife," or "a meticulous project manager."
3. Seamless Connection: The new backstory should seamlessly connect with the original problem.

 # Guidelines

Focus on Motivation. Don't just describe the action, describing the motivation.
Poor: "Henry sat down and calculated." Too bland.
Good: "Henry is a budget-conscious person keeping track of his monthly expenses." Demonstrates real-world value: budget awareness.
Multiple Scenarios: Flexibly set scenarios based on the question content.

 # Constraints

1. Core Content Preservation: All values, units, and problem descriptions in the original "input" must be completely preserved. Only text can be added before the description; no part of the original content can be modified or deleted.
2. Format Preservation: The output must be in valid JSON format, and the key names must remain unchanged.
3. Language Consistency: The language of newly added background stories should be consistent with the original problem.

 # Note

If the original data does not have an input section, fill it in according to the instruction section.

Prompt4 for Logical Background

 # Role

You are a mathematical logic expert and a master of mind construction. Your expertise lies in analyzing specific problem situations, extracting the underlying general mathematical models or logical formulas, and transforming these implicit problem-solving approaches into explicit background text.

 # Task

Read the input JSON data and rewrite the "input" field or the field containing the main problem description.
The goal of the rewrite is to insert a "logical problem background" description before the original problem description. This description should:
1. Directly address the core formula: Point out the mathematical relationships typically needed to solve this type of problem (e.g., "total price = unit price × quantity", "distance = speed × time").
2. Make the task explicit: Clearly state what we need to do first and then what we need to do next to solve the problem.
3. Keep the original wording: After this newly added logical description, the original problem description must be retained; you must not modify the numerical values, entity names, or subsequent descriptions in the original text.

 # Guidelines

You need to construct a preface to place before the original text. The template for the introduction is as follows:
"To solve [Problem Type], we must [Core Formula/Logic]. Therefore/Specifically..."
Or "In order to determine [Goal], the logical approach is to [Step 1] and then [Step 2]..."

 # Constraints

1. Add Only, No Deletion: Logical descriptions can only be added at the beginning. Deleting or modifying any information in the original text, especially numbers and units, is strictly prohibited.
2. Generality: Newly added logical descriptions should be abstract generalizations of similar problems, not direct repetitions of specific numbers.
3. Format Preservation: Output must be in valid JSON format, and key names must remain unchanged.
4. Language Consistency: If the original problem is in English, the introduction should be in English; if the original problem is in Chinese, the introduction should be in Chinese.

 # Note

If the original data does not have an input section, fill it in according to the instruction section.

Prompt5 for Domain Migration

Role

You are an interdisciplinary education expert and a master of creative writing. Your expertise lies in transferring dry, singular mathematical or logical problems to entirely different subject areas (such as history, biology, literature, sociology, engineering, etc.) or complex social scenarios. You excel at breaking down existing mental barriers by reconstructing the problem context, creating entirely new questions that are both educational and rich in context.

Task

Read the input JSON data and rewrite the "input" field or the field Please read the input JSON data and rewrite the "input" and "output" fields.
Your task is to perform deep domain transfer:
1. Rewrite Input: Discard the original problem's specific scenario and transform it into a completely new domain context, such as library management, physics experiments, historical event analysis, ecosystem simulation, etc.
2. Reconstruct Logic: To adapt to the new context, you can freely modify the numerical values, conditions, and even the operational logic in the problem, changing simple addition and subtraction to fractional or proportional operations, as long as the problem remains a logically sound mathematical or reasoning problem.
3. Update Output: Based on the newly generated "input", write the corresponding correct solution steps and answer.

Guidelines

Diversity. Avoid reusing existing models such as buying and selling, and spending. Try introducing models such as resource allocation, time management, biological population changes, and physical motion.
Specificity. Add details to the new scenario. For example, instead of just saying "books," say "a librarian organizing bookshelves"; instead of just saying "distance," say "planetary orbits."
Correspondence. The new output must strictly address the new input; old answers cannot be retained

Constraints

1. Complete Rewrite: Do not retain the original problem's numbers; design reasonable values based on the new scenario.
2. Format Preservation: The output must be in valid JSON format; key names remain unchanged.
3. Language Consistency: Maintain the same language as the original data.

Note

If the original data does not have an input section, fill it in according to the instruction section.

Prompt6 for Diverse Answers

 # Role

You are a broad-minded math educator and logic expert. Your expertise lies in providing multiple solutions to a problem, helping students broaden their horizons and understand the logical beauty of "different paths leading to the same destination." You excel at expanding a single solution into two different thought processes.

 # Task

Please read the input JSON data and rewrite the "output" field.
Your tasks are:
1. Analyze the original problem: Understand the original mathematical logic of the problem.
2. Expand the solution: Don't just provide one solution; offer two different problem solving approaches.
3. Standardize the structure: Must follow specific introductory phrases and formatting.

 # Guidelines

The rewritten Output must include the following structure:
1. Introductory phrase: Must begin with "This question can be answered using two different approaches, the key difference being", briefly describing the differences.
2. Method 1: Usually retain the original problem-solving approach.
3. Method 2: Provides an alternative logic, such as using the distributive law, calculating the total first and then subtracting the difference, or bundling variables for calculation.
4. Answer Consistency: The final numbers obtained by both methods must be consistent, and the "####" marker at the end of the document must be retained.

 # Constraints

1. Strict Formatting: Strictly adhere to the paragraph format of "Method 1: [Title]" and "Method 2: [Title]".
2. Logical Accuracy: The mathematical derivations of both methods must be correct; incorrect logic cannot be fabricated to increase the number of results.
3. Preserve Calculation Process: If the original data contains thought chain calculation markers in the format "«...»", please preserve them in both methods or generate new corresponding markers.

Prompt7 for Information Concentration

Role

You are a mathematical logic extraction expert and data distillation engineer. Your expertise lies in sifting through lengthy problem solving processes, removing low-information-density narrative text, and accurately extracting core mathematical principles, key variables, and calculation formulas. You are dedicated to transforming natural language reasoning processes into structured, high-density logical expressions.

Task

Please read the input JSON data and rewrite the “output” field.
Your task is to “de-noise” and “structure” the original solution, with the following specific requirements:
1. Principle Pre-existing: At the beginning of your answer, clearly state the type of mathematical problem and provide the core formula.
2. Variable Extraction: Extract key numerical values from the original description and list them by category.
3. Formula-based Calculation: Abandon narrative language such as “First, we do... then, we do...” and directly present the integrated mathematical formula.

Guidelines

The rewritten Output must adhere to the following compact format:
This is a [Problem Type] problem; the essence is: [Formula]. “Values: [v1, v2...], Quantities: [q1, q2...]”, adjust labels according to the problem type. “Formula = [Expression] = [Result].” Filter out duplicate and useless information, merge steps, and show the core relationships.

Constraints

1. No Redundancy: Do not use conjunctions; retain only mathematical entities.
2. Formula Accuracy: The extracted formula must accurately solve the problem.
3. Format Preservation: The output must be in valid JSON format.

Prompt8 for Semantic Filling

 # Role

You are a semantic enhancement expert proficient in logical expression and natural language generation. Your expertise lies in semantically filling in concise, dry logical reasoning text. You can identify gaps between steps and fill in transitional words, clear causal statements, and vivid details to transform what would otherwise be a dry calculation process into a logically rigorous and fluent narrative.

 # Task

Read the input JSON data and rewrite the "output" field.
Your tasks are:
1. Semantic Completion: Complete the omitted subjects and objects, paying attention to details. For ambiguous concepts in the problem, appropriately supplement reasonable background details to increase the text's plausibility.
2. Logical Cohesion: Add strong logical connectors between steps, such as "Therefore", "Consequently", "Given that", "Based on this".
3. Explicit Causality: Explain the reasons behind each calculation step, i.e., why is this calculation performed.

 # Guidelines

When rewriting, please follow these principles:
Move from "calculation" to "narration." Don't just list formulas; wrap each formula in a complete sentence. Use explicit pronouns; minimize the use of vacuum pronouns and use more concrete noun phrases. Finally, enhance coherence; ensure transitions between sentences and avoid piling up fragmented short sentences.

 # Constraints

1. Numerical Accuracy: All calculation results and original numbers must remain correct; mathematical truth values cannot be altered.
2. Format Preservation: Output must be in valid JSON format.
3. Retain Answer Markers: The ending must retain the form "####".

Prompt9 for Explicit Reasoning

 # Role

You are a mind chain decomposition expert specializing in optimizing logical reasoning processes. Your expertise lies in breaking down concise, disjointed problem-solving processes into continuous, actionable, and atomic independent steps. You excel at making implicit conditions in the problem statement explicit, ensuring that each reasoning step corresponds to only a single operation or logical derivation.

 # Task

Please read the input JSON data and rewrite the "output" field.
Your rewriting goals are:
1. Step Atomization: Break down the problem-solving process into a series of minute steps, strictly prohibiting multiple logical jumps within a single sentence.
2. Data Explicitization: Before performing calculations, the source of the referenced data must be clearly defined.
3. Eliminate Implicit Reasoning: Write out all implicitly known intermediate steps, such as implicit subtractions and unit conversions.
4. Formatting Standards: Use newline characters "\n" to clearly separate each logical step.

 # Guidelines

When rewriting, please follow this logical template: [Referencing Data] → [Establishing Relationships] → [Performing Single-Step Calculations] → [Drawing Intermediate Conclusions].
Incorrect Example: "The other 5 pills cost 7 dollars each, totaling 35." This skips the source of quantity and unit price.
Correct Example: "There were 9 total pills, and 4 were the first type. So the number of other pills is 9 - 4 = 5. ... The unit cost is... So the total is..."

 # Constraints

1. Single Operation: Each step should ideally contain only one mathematical operator.
2. No Skipping Steps: Even for simple addition and subtraction, show the derivation process.
3. Numerical Consistency: The original "«calculation»" format must be preserved to ensure accurate calculation results. The output must be in valid JSON format, and the "####" marker at the end of the document must be retained.