SPEAR: Stagewise Preference and Expertise Alignment for Spoken Mathematical Reasoning

Anonymous ACL submission

Abstract

Large audio language models (LALMs) demonstrate fluency in general dialogue but perform poorly in spoken mathematical interactions: their responses tend to be overloaded with formulas and delivered in a rigid, written style that is unsuitable for listening comprehension. To address this issue, we propose SPEAR, a decoupled two-stage post-training framework that separately teaches a LALM how to express itself and how to solve problems. Firstly, we construct a cascaded data generation pipeline incorporating self-verification and correction to produce high-quality, style-aligned samples. These samples are then used to train the model via preference learning, establishing a colloquial, step-by-step explanation style. Secondly, we introduce a reward mechanism that jointly evaluates both answer correctness and solution process quality, reinforcing the model's reasoning ability while preserving its spoken naturalness. Experiments on four mathematical question-answering datasets show that models trained with SPEAR outperform baseline LALMs and training methods, demonstrating that SPEAR effectively and simultaneously enhances both response accuracy and spoken naturalness.

1 Introduction

In recent years, large language models (LLMs) (Brown et al., 2020; Anil et al., 2023) based on the Transformer architecture (Vaswani et al., 2017) have achieved breakthroughs in semantic understanding through the pre-training and fine-tuning paradigm. Meanwhile, advancements in multimodal technologies have further facilitated the integration of text, speech, and image information. In the speech domain, this trend has directly led to the emergence of large language models targeting speech dialogue, namely LALMs. By transferring the successful experiences of textual LLMs, existing LALMs (Xu et al., 2025b; Team et al., 2025) have demonstrated considerable capabilities in general dialogue and content generation.

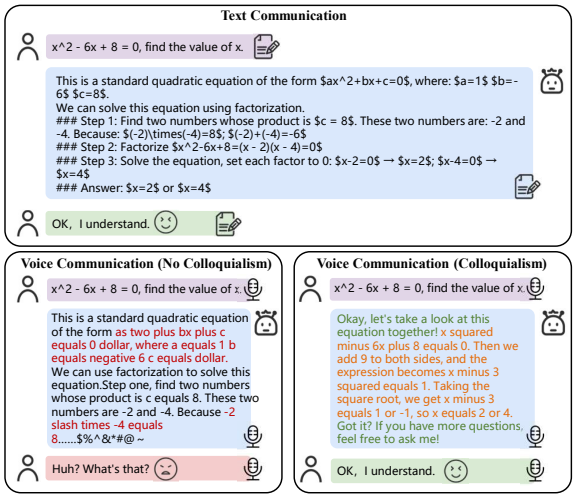

Figure 1: Differences between text and speech interactions of large models in mathematical dialogue interaction.

Mathematical reasoning tasks refer to problem-solving activities that involve logical deduction, symbolic manipulation, and abstract mathematical inference. As LLMs such as GPT(Achiam et al., 2023) and DeepSeek(Shao et al., 2024) mature via reinforcement learning post-training, they gain robust reasoning abilities. Through rigorous analysis and mathematical derivation, they deliver exceptional performance on math benchmarks, as validated on GSM8k(Cobbe et al., 2021), MATH(Hendrycks et al., 2021) and OlympiadBench(He et al., 2024). However, for LALMs that use speech for interaction, mathematical reasoning poses an additional challenge. In an audio interaction scenario, users cannot "see" complex formulas; they rely solely on hearing to comprehend the answer. Consequently, users prefer responses that are warm, concise, and colloquial, rather than lengthy, formula-heavy, and written answers. While preference alignment post-training has been used to

optimize expressive styles for voice assistants in some work Wu et al. (2025), how to systematically improve the response style of LALMs in math problem-solving without losing accuracy remains largely understudied.

In this paper, we propose Stagewise Preference and Expertise Alignment for Responses (SPEAR), a decoupled two-stage post-training framework for LALMs to speak appropriately in mathematical interactions. The core idea is to separate the optimization of how to express from how to solve. In the first stage, we establish a colloquial, step-by-step explanation style via preference learning. In the second stage, a reward mechanism that jointly evaluates answer correctness and solution process quality is introduced, reinforcing reasoning ability while preserving spoken naturalness. To support these two training stages, we design a complete pipeline for constructing preference datasets and reinforcement datasets. For preference data construction, we adopt a cascaded self-verification and revision pipeline to ensure samples' style and correctness. For reinforcement data construction, we select moderately difficult questions based on output difficulty analysis to avoid wasting resources on trivial or overly complex ones.

we conduct systematic experiments on four public math QA datasets: APE, GSM8K, SVAMP, and MAWPS, using two evaluation tasks: audio-to-text (A2T), where the model receives a spoken question and outputs a text answer; and audio-to-audio (A2A), where the model outputs a spoken answer that is then transcribed by an ASR system. Through comparative experiments and ablation studies, we validate that SPEAR effectively improves both the model's response naturalness and accuracy. Our contributions are as follows:

(1) We propose a decoupled two-stage post-training framework that separately optimizes expression style and reasoning ability for LALMs in spoken mathematical interactions.

(2) Our data construction pipeline features a cascaded self-verification and correction mechanism for preference data and a difficulty-based filtering strategy for reinforcement data.

(3) Results on four math datasets show that models trained with SPEAR outperform baseline LALMs and existing training methods in both accuracy and oral naturalness.

2 Related Work

LALMs. End-to-End LALMs aim to unify speech understanding, semantic reasoning, and speech generation within a single parameter space, improving interaction naturalness and real-time dialogue capability.

The Qwen-Audio series (Chu et al., 2023, 2024) builds cross-modal bridging with a single audio encoder for multi-task audio modeling, and its extension, Qwen-Omni series (Xu et al., 2025a,b) further realizes full-modal joint modeling via a unique Thinker-Talker architecture. SALMONN (Tang et al., 2023) activates cross-modal generalization of dual auditory encoder equipped LLMs via activation tuning; VITA (Fu et al., 2024) pioneers wake-up-free interruptible multimodal interaction, and EMOVA (Chen et al., 2025a) achieves fine-grained emotional speech control via semantically-acoustically decoupled modeling.

For unified speech understanding and generation, Baichuan-Audio (Li et al., 2025) adopts hierarchical residual vector quantization for speech discretization, GLM-4-Voice (Zeng et al., 2024) uses a three-module architecture for emotional and dialectal speech modeling, and Kimi-Audio (Ding et al., 2025) optimizes streaming speech generation with a dual-generation head design. In addition, the StepAudio series (Huang et al., 2025; Wu et al., 2025) deeply couples speech recognition and generation via end-to-end interleaved token modeling, SLAM-Omni (Chen et al., 2025b) builds foundational end-to-end speech interaction models with cross-modal reasoning, and Fun-Audio-Chat (Team et al., 2025) and Covo-Audio-Chat (Wang et al., 2026) focus on practical open-ended spoken dialogue applications.

LALMs with Post-training. Compared with LLMs, post-training research for LALMs is still in its early stages but growing rapidly in importance. StepAudio2 (Wu et al., 2025) introduces Proximal Policy Optimization (PPO) (Schulman et al., 2017) and Group Relative Policy Optimization (GRPO) (Shao et al., 2024) into speech model training: PPO controls response length to align with natural conversational pacing, while GRPO enhances audio understanding for complex inputs. Align-SLM (Lin et al., 2025) addresses the semantic coherence deficiency of textless SLMs by constructing preference data via AI feedback and fine-tuning with Direct Preference Optimization (DPO). SpeechAlign (Zhang et al., 2024) extends prefer-

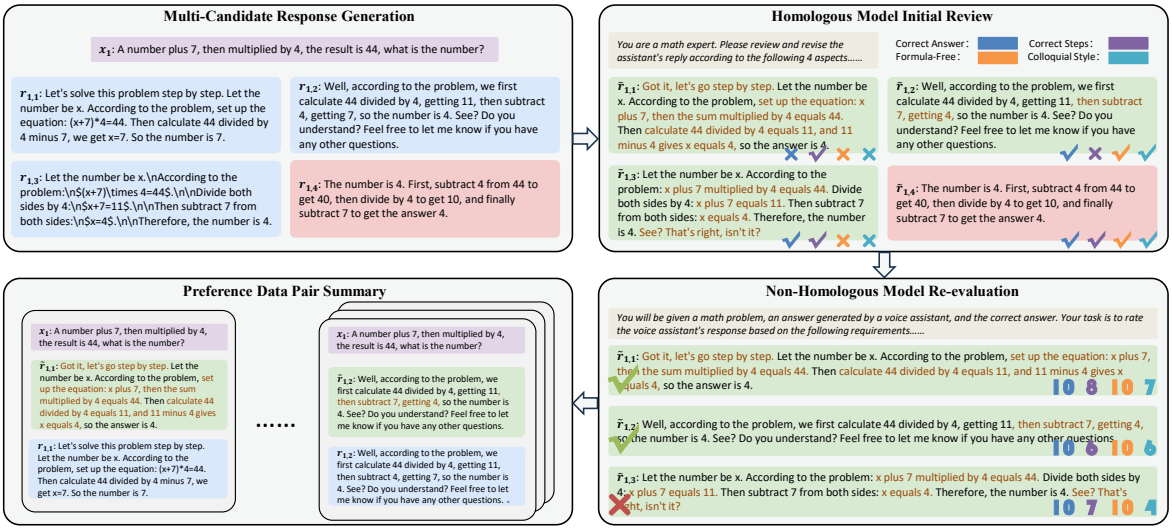

Figure 2: Construction process of a preference dataset for spoken style alignment.

ence optimization to codec-based speech generation, using iterative DPO to align outputs with human preferences for content accuracy, speaker similarity, and naturalness. Qwen3-Omni (Xu et al., 2025b) demonstrates a differentiated RL strategy for multi-module architectures: Group Sequence Policy Optimization (GSPO) (Zheng et al., 2025) strengthens the Thinker module's reasoning, while DPO improves the Talker module's generation stability and subjective quality. Yet, there is still a lack of systematic approaches to improving the response style of LALMs for mathematical problem-solving while maintaining their performance.

3 Method

4 Dataset Construction

4.1 Preference Dataset Construction

To build high-quality preference training data, we propose a cascaded "generation-scoring-correction-revalidation" pipeline that leverages the base model's generation and LLMs' evaluation capabilities to automate preference pair construction. Its overall framework is shown in Figure 2.

First, we run K independent inferences on the base LALM π_{base} with diverse decoding strategies to generate varied candidate responses $r_{i,1}, r_{i,2}, ..., r_{i,K}$. Then, a stronger LLM, denoted as the preliminary evaluation model $\pi_{\text{llm}}^{(1)}$, classifies each candidate $r_{i,k}$ as positive or negative, and records the unmet requirements $e_{i,k}$ for negative samples.

$\pi_{\text{llm}}^{(1)}$ uses a structured prompt to evaluate responses against four criteria: numerical consistency with the standard answer y_i^*, reasonable and error-free problem-solving steps, absence of LaTeX formulas or speech-incompatible special characters, and a friendly conversational tone. A response is labeled positive only if it meets all four criteria, and negative otherwise.

Next, $\pi_{\text{llm}}^{(1)}$ corrects the identified negative responses to satisfy all four criteria. We select $\pi_{\text{llm}}^{(1)}$ to be as homologous as possible to π_{base} to ensure that π_{base} can adapt more quickly to the positive class distribution during training.

The correction process takes a negative response $r_{i,k}$ and its corresponding failure reasons $e_{i,k}$ as inputs, and revises only the non-compliant parts while preserving the original core content. The corrected positive example response is denoted as $\tilde{r}_{i,k}$.

To mitigate inherent preference bias in the homologous model—which may favor base model-style responses and produce misjudgments or substandard positive examples—we introduce a second non-homologous LLM $\pi_{\text{llm}}^{(2)}$ as a validation model to re-evaluate $\pi_{\text{llm}}^{(1)}$-generated positive examples.

For each corrected positive example $\tilde{r}_{i,k}$, $\pi_{\text{llm}}^{(2)}$ applies stricter scoring across the four criteria, assigning a confidence score $s_{i,k}^{(d)} \in [0, 10]$ per dimension instead of binary labels. The final comprehensive score $S_{i,k}$ is computed as the product of the four dimension scores. We then set a threshold λ, retaining only high-quality positive examples with $S_{i,k} \geq \lambda$ and discarding both subthreshold examples and their corresponding original negative samples.

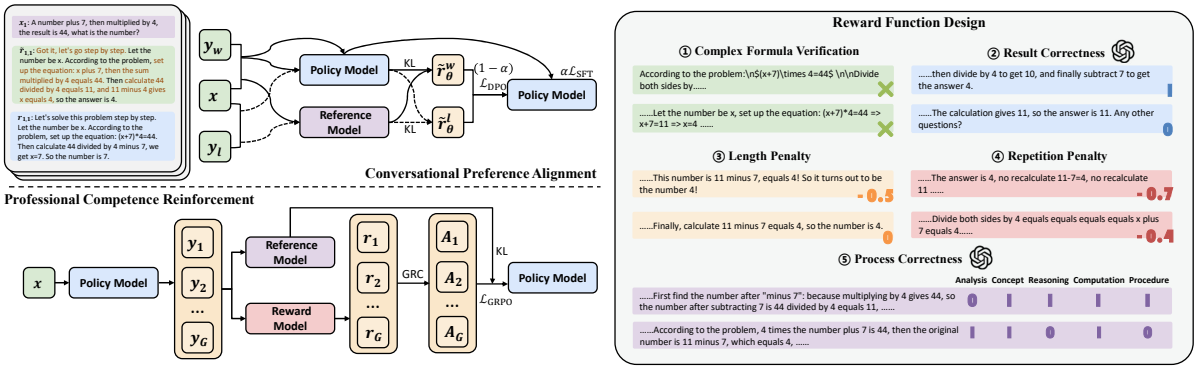

Figure 3: Specific training procedure of the two-stage preference alignment approach.

After these four steps, we retain multiple high-quality training pairs per question, each consisting of an original negative example $r_{i,k}$ and a corrected, validated positive example $\tilde{r}_{i,k}$, forming the final preference dataset $\mathcal{D}_{\text{pref}}$.

4.2 Reinforcement Learning Dataset

Next, we construct the reinforcement learning dataset. As a first step, we still generate K candidate responses per question, not to form preference pairs but to accurately assess the model's true performance on each question.

We use a third LLM $\pi_{\text{llm}}^{(3)}$ as the judge to evaluate response correctness, as string matching fails for equivalent expressions (e.g., "0.5" vs "1/2"). $\pi_{\text{llm}}^{(3)}$ takes question q_i, standard answer a_i^*, and candidate $r_{i,k}$ as inputs, outputting a binary label $y_{i,k} \in \{0, 1\}$ (1=correct, 0=incorrect), yielding K labels per question.

We compute the base model accuracy p_i for each question as the fraction of correct responses across K runs. We then filter to retain only medium-difficulty questions—those the model answers occasionally but has not fully mastered—forming the final RL dataset $\mathcal{D}_{\text{rl}}$.

5 Two-Stage Training

5.1 Conversational Preference Alignment

We perform DPO preference alignment training on $\mathcal{D}_{\text{pref}}$, with the loss function defined as:

$$\mathcal{L}_{\text{DPO}}(\pi_\theta; \pi_{\text{ref}}) = -\mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}_{\text{pref}}} \left[\log \sigma\Big(\hat{r}_\theta(x, y_w) - \hat{r}_\theta(x, y_l)\Big) \right]$$

where π_θ is the model to be optimized, π_{ref} is the reference model, y_w and y_l are the preferred ($\tilde{r}_{i,k}$) and non-preferred ($r_{i,k}$) responses from $\mathcal{D}_{\text{pref}}$, and $\hat{r}_\theta(x, y) = \beta \log \frac{\pi_\theta(y|x)}{\pi_{\text{ref}}(y|x)}$ is the implicit reward with β controlling the KL divergence penalty strength.

To enhance training stability, we blend a fraction of Supervised Fine-Tuning (SFT) loss into the DPO loss, weighted by hyperparameter $\alpha \in [0, 1]$.

5.2 Professional Competence Reinforcement

Next, based on $\mathcal{D}_{\text{rl}}$, we use GRPO to reinforce the model's professional knowledge reasoning ability, with its loss function shown below.

$$\begin{aligned}
\mathcal{L}_{\text{GRPO}}(\pi_\theta) = -\mathbb{E}_{x\sim\mathcal{D}}\mathbb{E}_{\{y_{i,t}\}_{i=1,t=1}^{G,|y|}\sim\pi_{\theta_{\text{old}}}(\cdot|x)} \Bigg[\\
\frac{1}{G\cdot|y|}\sum_{i=1}^{G}\sum_{t=1}^{|y|}\Big(\min\big(\rho_{i,t}(\theta)A_{i,t}, \\
\text{clip}\big(\rho_{i,t}(\theta), 1-\epsilon, 1+\epsilon\big)A_{i,t}\big) \\
-\beta\cdot\text{KL}\big(\pi_\theta(\cdot \mid x, \{y_{i,t'}\}_{t'<t}) \parallel \\
\pi_{\text{ref}}(\cdot \mid x, \{y_{i,t'}\}_{t'<t})\big)\Big)\Bigg]
\end{aligned}$$

where $\rho_i(\theta) = \frac{\pi_\theta(y_{i,t}|x,\{y_{i,t'}\}_{t'<t})}{\pi_{\theta_{\text{old}}}(y_{i,t}|x,\{y_{i,t'}\}_{t'<t})}$ is the importance sampling weight, $A_{i,t}$ is the intra-group advantage function normalized by group mean and variance, ϵ and β control clipping and KL penalty strength respectively, and π_{ref} is the DPO-trained reference model.

We design three reward functions: outcome reward for answer correctness, process reward for step correctness, and stabilization reward for training smoothness.

Outcome Reward The outcome reward r_{cor} evaluates only answer correctness, ignoring response quality. Given the conversational, unstructured nature of speech model outputs, we use an LLM judge

π_{cor} to assess whether the response's final answer is equivalent (not identical) to the standard answer y^*, with analysis preceding binary judgment. r_{cor} is defined as 1 for correct responses and 0 otherwise.

Process Reward To prevent correct answers from flawed reasoning, we design a checklist-style fine-grained process reward. The evaluation model π_{step} performs binary scoring across five dimensions: problem comprehension, concept understanding, logical reasoning, calculation accuracy, and step completeness. The final process reward is the sum of these five binary scores. The specific definitions of the five dimensions are as follows:

- Problem comprehension: Accurate understanding of question key information, conditions and requirements, no misunderstanding or omission of critical details.
- Concept understanding: Correct application of relevant concepts, definitions, common sense, formulas and theorems, no conceptual errors.
- Logical reasoning: Sound logical deduction with a coherent reasoning chain, no logical jumps or contradictions.
- Calculation: Accuracy in numerical calculations, algebraic operations and symbol processing, no errors.
- Step completeness: Sufficiently detailed steps for users to follow the reasoning process, no missing key steps.

Formula Filtering Before scoring both rewards, we run a rule-based formula check $f_{\text{formula}}(y)$ using regular expressions to detect LaTeX math environments and symbols. If formulas are detected, the corresponding reward is immediately set to 0, skipping LLM evaluation; otherwise, normal scoring proceeds.

Stabilization Reward Our experiments show that overly aggressive training parameters cause two typical generation degenerations as epochs increase: excessively verbose responses and frequent repetitive/meaningless sequences. These issues degrade user experience and disrupt reinforcement learning optimization. To address this, we introduce two stabilization rewards: length penalty and repetition penalty. They only activate for incorrect answers, avoiding interfering with correct response optimization.

Length penalty constrains response length to prevent excessive verbosity. Define soft upper limit $L_{\text{exp}} = L_{\text{max}} - L_{\text{cache}}$, where L_{max} is the hard maximum length and L_{cache} is the buffer length. For response token length L, excess length is $L_{\text{exceed}} = \max(0, L - L_{\text{exp}})$. The penalty is:

$$r_{\text{len}} = -\frac{L_{\text{exceed}}}{L_{\text{cache}}}$$

This non-positive reward equals 0 when $L \leq L_{\text{exp}}$, with penalty magnitude proportional to excess length.

Repetition penalty suppresses excessive n-gram repetition to avoid repetitive or meaningless sequences. Compute n-gram repetition rate $\rho = 1 - \frac{N_{\text{unique}}}{N_{\text{total}}}$. where N_{total} and N_{unique} are total and unique n-gram counts. Penalty r_{rep} uses preset maximum penalty $P_{\text{max}} < 0$ and threshold $\tau_{\text{rep}} < 0$ to avoid over-penalizing minor repetition:

$$r_{\text{rep}} = \begin{cases} \rho \cdot P_{\text{max}}, & \text{if } \rho \cdot P_{\text{max}} \leq \tau_{\text{rep}} \\ 0, & \text{otherwise} \end{cases}$$

We select either r_{len} or r_{rep} as r_{stab} based on task needs.

Comprehensive Reward The final GRPO reward combines outcome, process, and stabilization terms with r_{cor} as the dominant component:

$$r_{\text{total}}(x, y, y^*) = \alpha \cdot r_{\text{cor}} + (1 - \alpha) \cdot \frac{r_{\text{step}}}{5} + \mathbf{1}\big(r_{\text{cor}} = 0\big) \cdot r_{\text{stab}}$$

where α balances correctness and step quality, $r_{\text{step}}/5$ normalizes it to $[0, 1]$ to avoid numerical imbalance, $\mathbf{1}(\cdot)$ is the indicator function, and r_{stab} is the selected stabilization reward.

6 Experiment

6.1 Configuration

We use two 7B-parameter LALMs as baselines: Qwen2.5-Omni(Xu et al., 2025a) and Covo-Audio-Chat(Wang et al., 2026). The former has a tightly coupled audio-text design, so we only conduct A2T experiments on it. The latter has decoupled language model and TTS module, so we perform both A2T and A2A experiments on it.

Model	Audio-to-Audio				Audio-to-Text			
	APE	GSM8K	MAWPS	SVAMP	APE	GSM8K	MAWPS	SVAMP
Larger/Closed-source Models:								
Qwen3-Omni	40.18	9.45	53.52	56.00	**94.21**	**93.76**	**97.18**	91.33
GPT-4o-Audio	31.40	62.89	84.79	82.67	80.00	88.89	94.65	90.00
GPT-4o-Mini-Audio	22.81	69.93	89.58	83.00	68.07	87.69	95.21	88.00
Gemini2.5-pro	-	-	-	-	88.77	89.92	93.80	89.67
SFT:								
Qwen2.5-Omni	-	-	-	-	86.84	91.70	95.77	88.67
Covo-Audio-Chat-Chat	76.67	83.10	89.86	78.67	83.16	89.69	91.27	84.33
GRPO:								
Qwen2.5-Omni	-	-	-	-	88.60	91.13	96.34	92.33
Covo-Audio-Chat-Chat	74.91	82.93	91.55	81.33	80.53	84.19	91.27	82.67
SPEAR:								
Qwen2.5-Omni (ours)	-	-	-	-	90.88	**93.76**	96.62	**94.00**
Covo-Audio-Chat-Chat (ours)	**78.07**	**86.88**	**91.83**	**84.67**	87.89	89.81	91.83	86.67

Table 1: Comparison of mathematical capabilities between two-stage preference alignment, conventional training approaches, and larger/closed-source models. The data in bold indicates the best result on the current dataset, and the data underlined indicates the second-best result.

We select a series of state-of-the-art LALMs for comparison. These include GLM-4-Voice(Zeng et al., 2024), Kimi-Audio(Ding et al., 2025), Baichuan-Audio(Li et al., 2025), Step-Audio2-Mini(Wu et al., 2025), Mimo-Audio(Zhang et al., 2025), Fun-Audio-Chat(Team et al., 2025), GPT-4o-Audio, Gemini2.5-pro(Comanici et al., 2025), and Qwen3-Omni(Xu et al., 2025b). GPT-4o-Audio and Gemini2.5-pro are closed-source models. Fun-Audio-Chat has 8B parameters. Qwen3-Omni has 30B parameters. All remaining comparison models have 7B parameters.

For data construction and training, we deploy Qwen3-235B-A22B-Instruct(Yang et al., 2025) via vLLM for most evaluations. DeepSeek-v3(Liu et al., 2024) and GPT-4o APIs are used for Chinese and English inference result evaluation, respectively.

Table 2: Dataset for mathematical question answering experiments.

Dataset	Train	Test	Language
APE	12,820	190	Chinese
GSM8K	7,012	582	English
SVAMP	700	300	English
MAWPS	1,417	355	English

We select four publicly available mathematical question-answering datasets for experiments: APE(Zhao et al., 2020), GSM8K(Cobbe et al., 2021), SVAMP(Patel et al., 2021), and MAWPS(Koncel-Kedziorski et al., 2016). The scale and language distribution of each dataset are shown in Table 2.

The original datasets are in text form, we synthesize audio for training question texts using Cosy-Voice(Du et al., 2024) with multi-speaker random sampling for diversity. For test data, we use URO-Bench's(Yan et al., 2025) audio subsets for APE and GSM8K, while for SVAMP and MAWPS we generate test audio using the same Cosy-Voice pipeline.

We constructed preference datasets in both Chinese and English: the Chinese dataset is built from the training data of APE, and the English dataset is built from the training data of GSM8K. Additionally, we constructed a corresponding reinforcement dataset for each datasets. Details of the training set construction and model training are provided in Appendix A.1.

6.2 Comparison of Professional Competence

We compare the proposed approach against four baselines on mathematical ability: larger open-source models, closed-source models, SFT models, and GRPO-reinforced models with reward r_{cor}. Table 1 reports the evaluation results.

In the A2T mode, large-scale Qwen3-Omni achieves top scores on APE and MAWPS. Meanwhile, Qwen2.5-Omni after two-stage preference alignment matches Qwen3-Omni on GSM8K and surpasses it on SVAMP, demonstrating that a small model can match or exceed large general-purpose models in domain-specific reasoning after SPEAR

Model	Audio-to-Audio				Audio-to-Text			
	APE	GSM8K	MAWPS	SVAMP	APE	GSM8K	MAWPS	SVAMP
Larger/Closed-source Models:								
Qwen3-Omni	39.13	24.19	35.24	30.02	47.55	42.97	45.46	45.00
GPT-4o-Audio	43.61	49.11	54.28	53.38	56.68	48.63	50.89	51.40
GPT-4o-Mini-Audio	35.58	44.44	52.00	48.85	50.95	43.72	46.63	45.62
Gemini2.5-pro	-	-	-	-	57.97	46.05	41.85	45.92
Same-scale Models:								
GLM-4-Voice	58.18	47.42	50.54	46.57	57.16	49.33	52.90	49.07
Kimi-Audio	65.42	15.09	25.30	24.30	72.32	43.19	48.96	50.77
Step-Audio2-Mini	67.29	42.41	47.32	47.43	64.97	43.31	44.45	44.43
MiMo-Audio	73.61	39.89	32.99	33.82	36.95	41.89	62.59	61.45
Fun-Audio-Chat	54.87	47.79	49.37	49.02	63.79	61.13	60.31	57.82
Baseline:								
Qwen2.5-Omni	80.79	71.43	75.75	73.88	81.53	77.10	80.97	79.28
Covo-Audio-Chat-Chat	73.24	59.78	62.06	62.18	74.13	60.85	61.45	63.27
SPEAR:								
Qwen2.5-Omni (ours)	-	-	-	-	89.08	86.62	87.01	85.60
Covo-Audio-Chat-Chat (ours)	**85.89**	**82.07**	**83.24**	**80.40**	**89.16**	**87.18**	**87.86**	**86.03**

Table 3: Comparison of response styles between the baseline model after two-stage preference alignment and other models.

training.

In the A2A mode, Qwen3-Omni and closed-source models drop sharply, especially on APE and GSM8K. Beacuse their outputs tend to be formal, written-style responses with rigorous LaTeX expressions, which hinder speech synthesis and degrade performance. Additional issues include homophones, improper segmentation, and non-standard pronunciation. By contrast, our Covo-Audio-Chat not only outperforms these models by a large margin but also avoids a significant A2A-to-A2T drop, confirming that SPEAR can effectively reshape response style and enhances model's robustness to exhibit strong math reasoning ability in various interactive scenarios.

Comparing SFT and GRPO baselines, both Qwen2.5-Omni and Covo-Audio-Chat benefit more from the proposed two-stage framework than from either SFT or GRPO alone. Regardless of the baseline's pre-trained knowledge, SPEAR can further enhance model's domain knowledge and unlock its domain-specific reasoning potential .

6.3 Comparison of Response Styles

We compare response styles of four model types against our approach: larger open-source models, closed-source models, same-scale open-source models, and baseline models. For scoring, a dedicated prompt was constructed to instruct an LLM to score model's response from colloquialness and anthropomorphism.

As it shown in Table 3, the baselines of Qwen2.5-Omni and Covo-Audio-Chat already excel and outperforming other comparison models. Nevertheless, their response styles can be further improved through SPEAR training. After SPEAR training, the models not only become better at expressing solution processes in a colloquial manner but also demonstrate enhanced ability to use filler words, everyday expressions, and other friendly narrative styles.

Furthermore, some models (e.g., Qwen3-Omni and GPT-4o-Mini-Audio) show lower response style scores in A2A mode than in A2T mode. This is partly due to the same reason as the decline in mathematical ability: overly written-style responses cause the speech synthesis module to malfunction. Among 7B open-source models, some are designed for pure voice interaction, using additional A2A-style training data and a different system prompt to activate A2A capability. As a result, their A2A response style scores are actually higher than their A2T scores. However, the discrepancy between their A2A and A2T training data leads to performance imbalance across the two reasoning modes. In contrast, Covo-Audio-Chat originally maintains a balanced performance between the two modes, and this balance is preserved after SPEAR training, indicating that our training scheme does not compromise the model's inherent robustness

Model	Paradigm	Oral	Ability			APE		GSM8K	
			r_{cor}	r_{step}	r_{oral}	Acc (↑)	Sty (↑)	Acc (↑)	Sty (↑)
Qwen2.5-Omni	A2T	×	×	×	×	84.39	81.53	84.77	77.10
		×	✓	×	✓	86.32	89.24	89.35	**89.26**
		✓	×	×	×	87.37	89.63	93.13	85.98
		✓	✓	×	×	**90.88**	**89.87**	93.18	85.98
		✓	✓	✓	×	**90.88**	89.08	**93.76**	86.62
Covo-Audio-Chat-Chat	A2T	×	×	×	×	76.67	74.13	82.42	60.85
		×	✓	×	✓	80.18	74.53	82.25	62.02
		✓	×	×	×	84.21	**89.97**	86.37	86.86
		✓	✓	×	×	86.67	89.08	89.40	86.96
		✓	✓	✓	×	**87.89**	89.16	**89.81**	**87.18**
Covo-Audio-Chat-Chat	A2A	×	×	×	×	70.53	73.24	81.79	59.78
		✓	×	×	×	75.08	**86.89**	84.59	**82.07**
		✓	✓	×	×	**78.60**	85.87	85.57	81.70
		✓	✓	✓	×	78.07	85.89	**86.88**	**82.07**

Table 4: Ablation study on the APE and GSM8k datasets. r_{oral} scores from four aspects: colloquialness, emotional richness, anthropomorphism, and conciseness.

and cross-modal transferability.

6.4 Ablation Study

To validate the contribution of each training component, we conduct ablation experiments on APE and GSM8K. The experimental results are presented in Table 4. Firstly, across all settings, oral alignment training substantially improves style scores and also yields a modest accuracy gain. This indicates that oral alignment not only makes responses more natural and colloquial but also enhances performance on specific math tasks. Secondly, adding professional reinforcement training with r_{cor} after oral alignment further improves mathematical accuracy without sacrificing response style. This confirms the effectiveness of performing professional reinforcement after oral alignment, which strengthens domain-specific reasoning while preserving the adjusted style. Thirdly, when process reward r_{step} is further added, performance improves again. Fine-grained step-level rewards encourage more complete and structured reasoning, leading to more correct final answers. Although the improvement from step rewards is modest, it remains valuable given that accuracy is already high and near the model's bottleneck.

To further demonstrate the advantage of separately optimizing response style and domain-specific ability via two-stage training, we additionally report results from directly using GRPO to jointly optimize both objectives ($r_{\text{cor}} + r_{\text{oral}}$) in A2T experiments. As the results show, this joint method outperforms the baseline in both accuracy and style, but still lags significantly behind our two-stage method, especially in accuracy. The reason is that the two objectives have inconsistent optimization paths; GRPO can only achieve a trade-off rather than the optimum for each. However, our SPEAR decouples the objectives by performing oral alignment first followed by professional reinforcement sequentially, making each stage's objective clearer and optimization more sufficient, thus achieving optimal outcomes for both.

7 Conclusion

We propose SPEAR, a two-stage optimization framework that improves both response accuracy and colloquialness. The first stage uses preference alignment to make the response style more friendly, lively, and anthropomorphic. The second stage applies professional reinforcement with a dual reward mechanism to enhance reasoning ability while preserving the colloquial style. For data construction, we design a cascaded pipeline of "generation-scoring-correction-revalidation" and a difficulty-based filtering strategy. Extensive experiments on four mathematical QA datasets show that the baseline models trained with SPEAR significantly improve performance in both domain-specific reasoning and colloquial response capability, matching or even exceeding larger closed-source models. Ablation studies validate the contributions of each stage and confirm the advantage of two-stage decoupled optimization.

Limitations

There remain several promising directions for extending this work. First, while we focus on oral mathematical QA as a representative scenario, extending the decoupled two-stage framework to other STEM disciplines such as physics or chemistry reasoning tasks remains an open direction. Second, the checklist used for process reward requires further adjustment and optimization to better capture fine-grained reasoning quality in spoken interactions. Third, the prompts employed in evaluation may not fully unleash the model's potential, and different prompt designs could lead to slightly varied results, exploring more effective prompt strategies is valuable.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, and 1 others. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Rohan Anil, Andrew M Dai, Orhan Firat, Melvin Johnson, Dmitry Lepikhin, Alexandre Passos, Siamak Shakeri, Emanuel Taropa, Paige Bailey, Zhifeng Chen, and 1 others. 2023. Palm 2 technical report. *arXiv preprint arXiv:2305.10403*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, and 1 others. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Kai Chen, Yunhao Gou, Runhui Huang, Zhili Liu, Daxin Tan, Jing Xu, Chunwei Wang, Yi Zhu, Yihan Zeng, Kuo Yang, and 1 others. 2025a. Emova: Empowering language models to see, hear and speak with vivid emotions. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 5455–5466.

Wenxi Chen, Ziyang Ma, Ruiqi Yan, Yuzhe Liang, Xiquan Li, Ruiyang Xu, Zhikang Niu, Yanqiao Zhu, Yifan Yang, Zhanxun Liu, and 1 others. 2025b. Slam-omni: Timbre-controllable voice interaction system with single-stage training. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 2262–2282.

Yunfei Chu, Jin Xu, Qian Yang, Haojie Wei, Xipin Wei, Zhifang Guo, Yichong Leng, Yuanjun Lv, Jinzheng He, Junyang Lin, and 1 others. 2024. Qwen2-audio technical report. *arXiv preprint arXiv:2407.10759*.

Yunfei Chu, Jin Xu, Xiaohuan Zhou, Qian Yang, Shiliang Zhang, Zhijie Yan, Chang Zhou, and Jingren Zhou. 2023. Qwen-audio: Advancing universal audio understanding via unified large-scale audio-language models. *arXiv preprint arXiv:2311.07919*.

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*.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, and 1 others. 2025. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*.

Ding Ding, Zeqian Ju, Yichong Leng, Songxiang Liu, Tong Liu, Zeyu Shang, Kai Shen, Wei Song, Xu Tan, Heyi Tang, and 1 others. 2025. Kimi-audio technical report. *arXiv preprint arXiv:2504.18425*.

Zhihao Du, Qian Chen, Shiliang Zhang, Kai Hu, Heng Lu, Yexin Yang, Hangrui Hu, Siqi Zheng, Yue Gu, Ziyang Ma, and 1 others. 2024. Cosyvoice: A scalable multilingual zero-shot text-to-speech synthesizer based on supervised semantic tokens. *arXiv preprint arXiv:2407.05407*.

Chaoyou Fu, Haojia Lin, Zuwei Long, Yunhang Shen, Yuhang Dai, Meng Zhao, Yi-Fan Zhang, Shaoqi Dong, Yangze Li, Xiong Wang, and 1 others. 2024. Vita: Towards open-source interactive omni multimodal llm. *arXiv preprint arXiv:2408.05211*.

Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, and 1 others. 2024. Olympiadbench: A challenging benchmark for promoting agi with olympiad-level bilingual multimodal scientific problems. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3828–3850.

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*.

Ailin Huang, Boyong Wu, Bruce Wang, Chao Yan, Chen Hu, Chengli Feng, Fei Tian, Feiyu Shen, Jingbei Li, Mingrui Chen, and 1 others. 2025. Step-audio: Unified understanding and generation in intelligent speech interaction. *arXiv preprint arXiv:2502.11946*.

Rik Koncel-Kedziorski, Subhro Roy, Aida Amini, Nate Kushman, and Hannaneh Hajishirzi. 2016. Mawps: A math word problem repository. In *Proceedings of the 2016 conference of the north american chapter of the association for computational linguistics: human language technologies*, pages 1152–1157.

Tianpeng Li, Jun Liu, Tao Zhang, Yuanbo Fang, Da Pan, Mingrui Wang, Zheng Liang, Zehuan Li, Mingan Lin, Guosheng Dong, and 1 others. 2025. Baichuan-audio: A unified framework for end-to-end speech interaction. *arXiv preprint arXiv:2502.17239*.

Guan-Ting Lin, Prashanth Gurunath Shivakumar, Aditya Gourav, Yile Gu, Ankur Gandhe, Hung-yi Lee, and Ivan Bulyko. 2025. Align-slm: Textless spoken language models with reinforcement learning from ai feedback. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 20395–20411.

Aixin Liu, Bei Feng, Bin Wang, Bingxuan Wang, Bo Liu, Chenggang Zhao, Chengqi Dengr, Chong Ruan, Damai Dai, Daya Guo, and 1 others. 2024. Deepseek-v2: A strong, economical, and efficient mixture-of-experts language model. *arXiv preprint arXiv:2405.04434*.

Arkil Patel, Satwik Bhattamishra, and Navin Goyal. 2021. Are nlp models really able to solve simple math word problems? In *Proceedings of the 2021 conference of the North American chapter of the association for computational linguistics: human language technologies*, pages 2080–2094.

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

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, and 1 others. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.

Changli Tang, Wenyi Yu, Guangzhi Sun, Xianzhao Chen, Tian Tan, Wei Li, Lu Lu, Zejun Ma, and Chao Zhang. 2023. Salmonn: Towards generic hearing abilities for large language models. *arXiv preprint arXiv:2310.13289*.

Tongyi Fun Team, Qian Chen, Luyao Cheng, Chong Deng, Xiangang Li, Jiaqing Liu, Chao-Hong Tan, Wen Wang, Junhao Xu, Jieping Ye, and 1 others. 2025. Fun-audio-chat technical report. *arXiv preprint arXiv:2512.20156*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Wenfu Wang, Chenxing Li, Liqiang Zhang, Yiyang Zhao, Yuxiang Zou, Hanzhao Li, Mingyu Cui, Hao Zhang, Kun Wei, Le Xu, and 1 others. 2026. Covo-audio technical report. *arXiv preprint arXiv:2602.09823*.

Boyong Wu, Chao Yan, Chen Hu, Cheng Yi, Chengli Feng, Fei Tian, Feiyu Shen, Gang Yu, Haoyang Zhang, Jingbei Li, and 1 others. 2025. Step-audio 2 technical report. *arXiv preprint arXiv:2507.16632*.

Jin Xu, Zhifang Guo, Jinzheng He, Hangrui Hu, Ting He, Shuai Bai, Keqin Chen, Jialin Wang, Yang Fan, Kai Dang, and 1 others. 2025a. Qwen2. 5-omni technical report. *arXiv preprint arXiv:2503.20215*.

Jin Xu, Zhifang Guo, Hangrui Hu, Yunfei Chu, Xiong Wang, Jinzheng He, Yuxuan Wang, Xian Shi, Ting He, Xinfa Zhu, and 1 others. 2025b. Qwen3-omni technical report. *arXiv preprint arXiv:2509.17765*.

Ruiqi Yan, Xiquan Li, Wenxi Chen, Zhikang Niu, Chen Yang, Ziyang Ma, Kai Yu, and Xie Chen. 2025. Uro-bench: Towards comprehensive evaluation for end-to-end spoken dialogue models. *arXiv preprint arXiv:2502.17810*.

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*.

Aohan Zeng, Zhengxiao Du, Mingdao Liu, Kedong Wang, Shengmin Jiang, Lei Zhao, Yuxiao Dong, and Jie Tang. 2024. Glm-4-voice: Towards intelligent and human-like end-to-end spoken chatbot. *arXiv preprint arXiv:2412.02612*.

Dong Zhang, Zhaowei Li, Shimin Li, Xin Zhang, Pengyu Wang, Yaqian Zhou, and Xipeng Qiu. 2024. Speechalign: Aligning speech generation to human preferences. *Advances in Neural Information Processing Systems*, 37:50343–50360.

Dong Zhang, Gang Wang, Jinlong Xue, Kai Fang, Liang Zhao, Rui Ma, Shuhuai Ren, Shuo Liu, Tao Guo, Weiji Zhuang, and 1 others. 2025. Mimo-audio: Audio language models are few-shot learners. *arXiv preprint arXiv:2512.23808*.

Wei Zhao, Mingyue Shang, Yang Liu, Liang Wang, and Jingming Liu. 2020. Ape210k: A large-scale and template-rich dataset of math word problems. *arXiv preprint arXiv:2009.11506*.

Chujie Zheng, Shixuan Liu, Mingze Li, Xiong-Hui Chen, Bowen Yu, Chang Gao, Kai Dang, Yuqiong Liu, Rui Men, An Yang, and 1 others. 2025. Group sequence policy optimization. *arXiv preprint arXiv:2507.18071*.

A Appendix

A.1 Experimental details

Preference Data For each question, we use the baseline model Qwen2.5-Omni to generate $K = 20$ candidate responses (rollouts). During decoding, we employ random temperature sampling ($\tau \in \{0.7, 0.8, 0.9, 1.0\}$) and top-p sampling ($p = 0.95$) to ensure response diversity. The homologous evaluation model $\pi_{\text{llm}}^{(1)}$ uses the deployed Qwen3-235B-A22B-Instruct vLLM service to perform initial evaluation and classification of the candidate responses, labeling positive and negative examples, and correcting negative examples. For non-homologous evaluation models, we consider using the DeepSeek-v3 API or the GPT-4o API to verify and filter the corrected positive examples. The filtering threshold is $\lambda = 950$. For A2A tasks, we use Cosy-Voice to synthesize speech for the filtered positive and negative text responses, serving as audio targets for subsequent training.

Reinforcement Data For each question, we use the baseline model Qwen2.5-Omni to generate $K = 20$ candidate responses (same generation configuration as in preference dataset construction). We use Qwen3-235B-A22B-Instruct as the evaluation model $\pi_{\text{llm}}^{(3)}$ to judge the correctness of each response and calculate the correctness rate p_i for each question. We remove easy questions ($p_i \geq \tau_{\text{high}}$) and overly complex questions ($p_i \leq \tau_{\text{low}}$), retaining questions of moderate difficulty ($\tau_{\text{low}} < p_i < \tau_{\text{high}}$). The threshold settings vary by dataset; for example, for APE, we set $\tau_{\text{low}} = 0.2$ and $\tau_{\text{high}} = 0.8$.

DPO training In DPO training, the KL divergence strength coefficient β is set to 0.1, the SFT proportion coefficient α to 0.1, the learning rate to 1e-6, and the Adam optimizer is used for full fine-tuning of the LLM part of the model, with a maximum of 4000 training steps.

GRPO training In GRPO training, for the length penalty r_{len}, the maximum length limit $L_{\text{max}} = 600$ and cache length $L_{\text{cache}} = 200$ are set; the repetition penalty r_{rep} is computed based on 5-grams, with maximum penalty value $P_{\text{max}} = -1.0$ and penalty threshold $\tau_{\text{rep}} = -0.5$; the balance coefficient $\alpha = 0.5$ for the combined reward function. GRPO training also uses a KL strength coefficient $\beta = 0.1$, but the learning rate is adjusted to 5e-7, still using the Adam optimizer for full fine-tuning of the LLM part, with a maximum of 1500 training steps.

A.2 GRPO Stability Analysis

During GRPO training with both outcome reward r_{cor} and process reward r_{step}, we observed that model rollouts tend to degenerate as training steps increase. The degeneration includes extensive irregular repetitions, garbled characters, or failure to stop after the correct answer, leading to training collapse. To address this, we previously proposed two stabilizing reward mechanisms: a length penalty r_{len} and a repetition penalty r_{rep}. Here we investigate their effects along with adjusting the KL divergence weight through comparative experiments, systematically analyzing each stabilization strategy.

We use the Qwen2.5-Omni model on the APE dataset and design five sets of comparative experiments. Each set uses r_{cor} and r_{step} as base rewards, and then introduces different stabilization strategies accordingly:

- Experiment a (baseline): only r_{cor} and r_{step}, with KL weight $\beta = 0.001$.
- Experiment b: r_{cor}, r_{step}, and r_{rep}, $\beta = 0.001$.
- Experiment c: r_{cor}, r_{step}, and r_{len}, $\beta = 0.001$.
- Experiment d: only r_{cor} and r_{step}, $\beta = 0.01$.
- Experiment e: only r_{cor} and r_{step}, $\beta = 0.1$.

We use the 3-gram repetition rate as a quantitative stability metric, monitoring the repetition rate of rollouts during training and of model responses during testing. We also record the best accuracy achieved by each experimental group on APE. The results are shown in Figure 4.

From the training repetition rate trend, the stability ranking is: Experiment e > d > c > b > a. This indicates that increasing the KL weight is the most effective at suppressing garbled repetitions, while stabilizing rewards are less effective. In the testing phase, Experiments a, b, and c show increasing repetition rates over training steps, meaning that even if degeneration is suppressed during training with stabilizing rewards, models may still output garbled repetitions at test time. In addition, The best accuracy ranking is: a > c > b ≈ d > e. This shows that stabilization measures often incur performance loss. The length penalty causes relatively small loss, whereas the repetition penalty

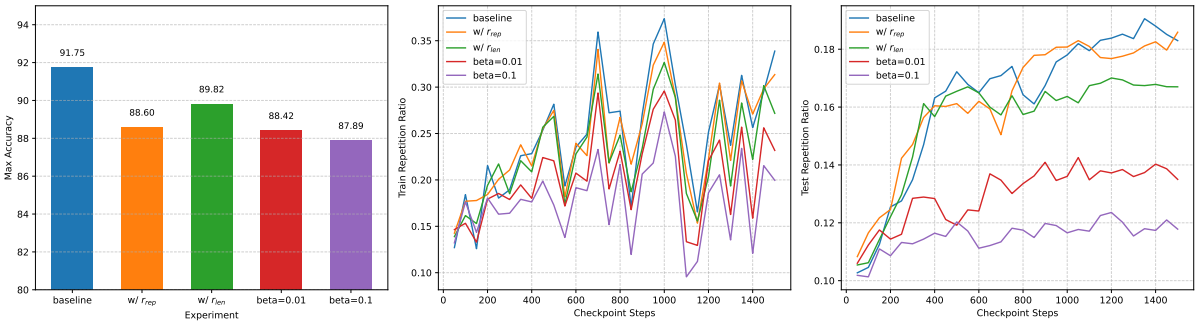

Figure 4: Comparison of GRPO training stability across different stability-preserving schemes.

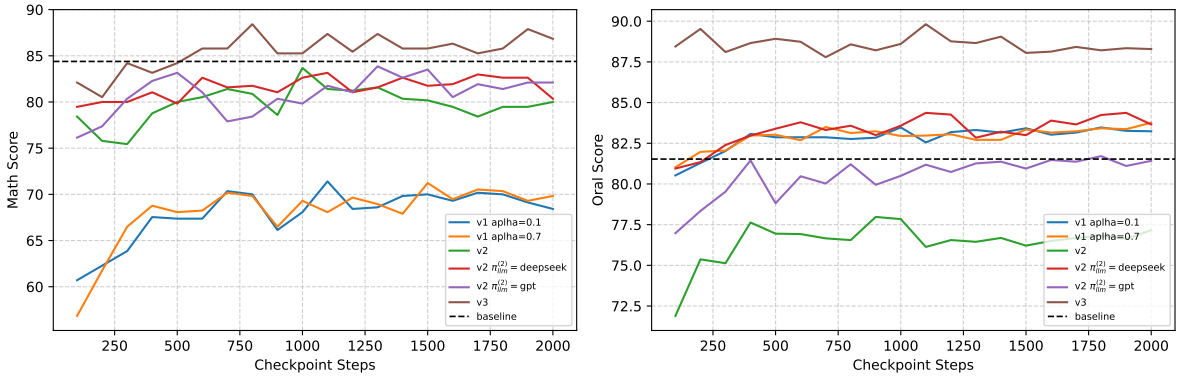

Figure 5: Comparison of preference dataset effectiveness across different processing schemes.

and increasing the KL weight lead to larger degradation.

Based on this trade-off, for mathematical QA tasks, we adopt the stabilization strategy with KL weight $\beta_{\text{KL}} = 0.1$ to ensure test-time and practical application stability, despite a considerable accuracy loss. In future work, we will explore combining a moderate KL weight with lightweight stabilizing rewards (e.g., length penalty) to better balance stability and performance.

A.3 Preference Dataset Analysis

Preference dataset quality is critical for oral alignment. Taking the APE dataset as an example, we describe the iterative construction process, which is shown in Figure 5.

The first version (v1) follows the basic pipeline: Qwen2.5-Omni generates 20 candidate responses per APE training question; Qwen3-235B-A22B-Instruct as $\pi_{\text{llm}}^{(1)}$ evaluates, labels positive/negative examples, and corrects negatives; then DeepSeek-v3 as $\pi_{\text{llm}}^{(2)}$ verifies and filters, which yields about 30,000 preference pairs. However, Qwen2.5-Omni trained on v1 performs poorly on the APE test set. Analysis shows that the corrected positive examples tend to present the answer at the beginning, followed by reasoning, which makes the trained model more prone to errors. Additionally, comparing SFT weights $\alpha = 0.7$ vs. 0.1 shows that the lower weight has a slight accuracy advantage, preserving more optimization space.

To address the above issue, we build v2: explicitly require the corrected positive examples of $\pi_{\text{llm}}^{(1)}$ to follow the format of "analyze first, give the answer at the end" and have $\pi_{\text{llm}}^{(2)}$ assign lower scores to responses that present the answer before the analysis. After v2 training, accuracy improves but still lags baseline. We also compare DeepSeek-v3 vs. GPT-4o as $\pi_{\text{llm}}^{(2)}$. It shows that DeepSeek-v3 and GPT-4o give similar accuracy, but DeepSeek-v3 achieves better style scores, indicating superior colloquialness judgment for Chinese data.

Encountering a bottleneck on APE, we turn to GSM8K. The construction strategy is simplified: use only $\pi_{\text{llm}}^{(1)}$ without secondary verification, because GSM8K questions are easier and the corrected positives are already high quality. Moreover, we find most negative examples are flagged solely for "containing written-style formulas", while correctness and colloquialness are fine. To avoid excessive focus on format, we remove most formula-related negatives, retaining only 200 samples. DPO

training on this optimized GSM8K dataset yields accuracy and style far above baseline, which guides further APE optimization.

Re-examining APE, we identify two issues: $\pi_{\text{llm}}^{(2)}$ might introduce noise, and $\pi_{\text{llm}}^{(1)}$ may also select many formula-only negatives. We design a comparative experiment omitting $\pi_{\text{llm}}^{(2)}$ and randomly selecting 30,000 samples from $\pi_{\text{llm}}^{(1)}$ outputs. The version without secondary verification shows no significant accuracy difference from the version with verification, but its style score drops notably. Thus, secondary verification helps style consistency but may be omitted when resources are constrained.

Based on these findings, we construct v3 with four key improvements:

1. Omit secondary verification entirely, simplifying the pipeline.

2. Enhance prompt diversity: randomly insert multiple colloquial output templates in $\pi_{\text{llm}}^{(1)}$ to compensate for potential style loss.

3. Reduce format-only negative examples: keep only 200 samples flagged for written-style formulas to avoid overemphasis on format.

4. Apply difficulty filtering: retain preference pairs for high-difficulty questions, making training more targeted.

The result shows that Qwen2.5-Omni trained on v3 achieves optimal performance on the APE test set and fully validates the v3 improvements.