CLARIFYACT: LEARNING TO CLARIFY WHEN NEEDED AND ACT WHEN READY

Anonymous authors
Paper under double-blind review

ABSTRACT

Language models must not only know how to solve tasks, but also determine whether a request provides sufficient information to proceed. Missing premises, unclear intent boundaries, or erroneous constraints can cause failures on otherwise solvable tasks, whereas indiscriminate clarification creates unnecessary interaction costs. We define the performance difference between complete and defective requests as the **suppression gap**. Controlled experiments across multiple models, using complete, defective, and information-restored requests, empirically demonstrate this gap and recovery after information restoration. Motivated by this recoverability, we propose View-Specific Group-Relative Optimization (**V-SGRO**), which learns protective decisions from self-generated rollouts under task-verifier feedback, without external demonstrations. V-SGRO distinguishes verification status from relative quality, combining certification, preference, and residual quality signals into a composite advantage for unified policy updates. A fixed-candidate analysis establishes conditions for joint directional progress across these objectives. Experiments show that the full configuration improves the accuracy of decisions between protection and execution while reducing premature execution. Ablations and behavioral attribution analyses further reveal distinct effects of the learning signals on challenging requests.

1 INTRODUCTION

Language models must not only solve tasks, but also decide whether a request provides sufficient information to proceed. Real-world requests may omit critical premises, leave intent underspecified, or contain constraints unsupported by available evidence. Acting too early can therefore violate user intent, while indiscriminate clarification interrupts executable tasks and increases interaction cost (Zhang et al., 2024; Zhang & Choi, 2025). Recent analyses further show that recognizing ambiguity does not reliably translate into asking for clarification (Su & Cardie, 2026). Reliable request handling requires protective action when information is insufficient and timely execution when it is sufficient.

We define the performance difference between complete and defective requests as the **suppression gap** and introduce SG-Bench to study it. Controlled comparisons among complete, defective, and information-restored requests reveal degradation and recovery across multiple models and task sources, characterizing how execution performance varies across request conditions. This motivates learning when to ask, confirm, challenge, or execute according to the current request state.

The recovery observed after information restoration suggests that such decisions can be learned from a model's own generations. However, self-generated candidates carry different forms of supervision: certified responses provide absolute success signals, non-perfect candidates reveal preference directions, and the remaining responses retain relative-quality information. Existing filtering (Zelikman et al., 2022; Singh et al., 2023), pairwise preference learning (Rafailov et al., 2023), and group-relative optimization (Shao et al., 2024) exploit only part of this structure.

We therefore propose View-Specific Group-Relative Optimization (**V-SGRO**). V-SGRO partitions self-generated candidates into certified, preference, and residual views, derives a source-specific advantage from each, and combines them into a unified centered policy update. Gradient alignment further modulates the residual contribution. Training relies only on self-generated trajectories and task-verifier feedback, without external demonstrations. We evaluate V-SGRO on controlled benchmarks,

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

source ablations, and behavioral analyses, and analyze joint directional progress in a fixed-candidate model.

Our main contributions are threefold:

- **Problem formulation and empirical evidence.** We formalize the suppression gap and introduce SG-Bench, a three-condition diagnostic benchmark. Controlled experiments across models and task sources establish performance degradation and recovery, grounding the goal of learning to clarify when needed and execute when ready.
- **Multi-view optimization from self-generated trajectories.** We propose V-SGRO to distinguish verification status from relative quality and integrate certification, preference, and residual quality signals into a composite advantage. We establish conditions for joint directional progress in a fixed-candidate model.
- **Performance advantages and behavioral attribution.** On dialogue control tasks, the full configuration achieves higher control accuracy than multiple classes of training baselines and reduces premature execution. Ablations and behavioral attribution analyses identify the distinct effects of different learning signals on challenging requests.

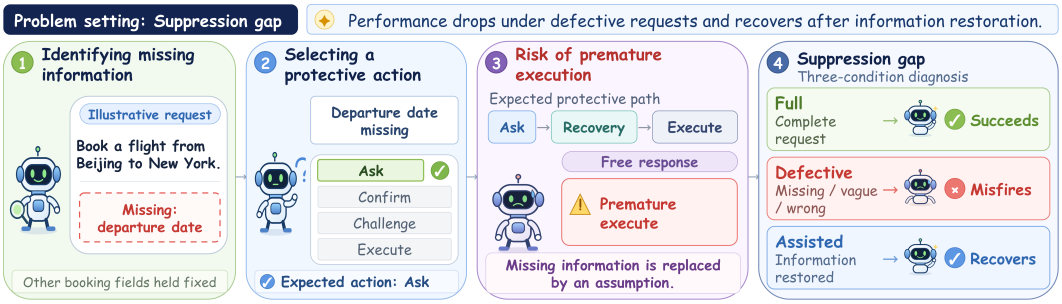

Figure 1: **Problem setting and three-condition diagnosis of the suppression gap.** (1) Identify missing task information. (2) Select a protective action—asking, confirming, or challenging. (3) Illustrate how missing information can lead to premature execution. (4) Compare execution under complete, defective/no-ask, and information-restored requests to show performance degradation and recovery for the same model.

2 Suppression Gap: Definition and Empirical Findings

We examine execution across complete, defective, and information-restored requests (Figure 1), using our self-constructed SG-Bench and a diagnostic adaptation of τ^2-Bench. Figure 2 summarizes benchmark construction, and Table 1 reports the paired evaluations.

2.1 Three-Condition Diagnosis and Problem Definition

For each underlying task t, we construct a complete request x_t^{full}, a defective request x_t^{bad}, and an information-restored request $x_t^{\text{assisted}} = x_t^{\text{bad}} \oplus a_t$, where a_t supplies missing information or repairs a constraint and $\oplus$ denotes information supplementation.

Let n be the number of diagnostic tasks and $s_t^c(m) \in \{0, 1\}$ indicate whether model m passes the corresponding execution check for task t under condition $c \in \{\text{full}, \text{bad}, \text{assisted}\}$. We define the **suppression gap (SG)** and **recovery gain (RG)** as

$$\begin{aligned} \text{SG}(m) &= \frac{1}{n} \sum_{t=1}^{n} \left[s_t^{\text{full}}(m) - s_t^{\text{bad}}(m) \right], \\ \text{RG}(m) &= \frac{1}{n} \sum_{t=1}^{n} \left[s_t^{\text{assisted}}(m) - s_t^{\text{bad}}(m) \right]. \end{aligned} \tag{1}$$

These metrics characterize observed execution performance under the diagnostic protocol: SG compares complete with defective requests, and RG measures the gain after information restoration.

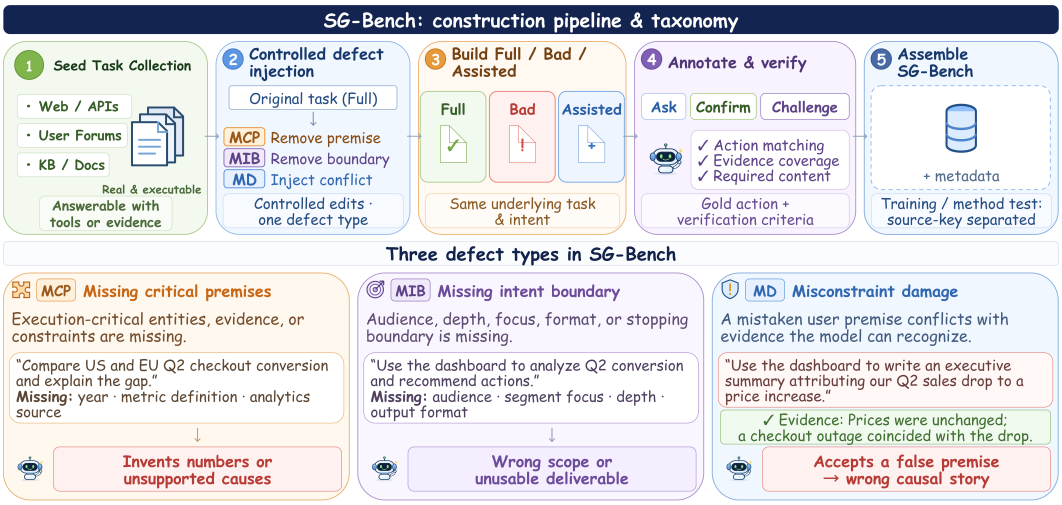

Figure 2: **SG-Bench construction pipeline and defect taxonomy.** (1) Collect public tasks and freeze their evidence. (2) Inject one controlled defect into each complete request. (3) Construct full, bad, and assisted variants of the same task. (4) Annotate protective actions, relevant dimensions, and evidence checks. (5) Assemble the benchmark with source-key metadata and data splits. The lower panels illustrate missing critical premises (MCP), missing intent boundaries (MIB), and misconstraint damage (MD).

Contextualized evaluation highlights how available context affects judgments of responses to underspecified queries (Malaviya et al., 2025). Our three-condition diagnosis measures execution differences under information defects and restoration, complementing the recognition–behavior gap studied by Su & Cardie (2026).

2.2 Self-Constructed and Adapted Benchmarks

Self-constructed SG-Bench. As shown in Figure 2, construction comprises task collection, defect injection, triplet construction, annotation and checking, and benchmark assembly. We collect public tables and documents, clean and freeze their contents, and construct evidence-grounded complete requests. We introduce missing critical premises (MCP) by removing execution-relevant entities, evidence, or constraints; missing intent boundaries (MIB) by removing scope, audience, or output requirements; and misconstraint damage (MD) by introducing premises or constraints that conflict with the evidence. Each defective request is paired with recovery information and annotated with its expected action, relevant dimensions, and evidence checks. Source identifiers support source-key-separated training and method testing. The diagnostic set contains 240 samples from 80 source keys, with 80 samples per defect type and 720 records per model. A common content checker evaluates required evidence citations, required content coverage, and prohibited items. Appendix B details the source composition and collection splits.

Adapted τ^2-Bench. Starting from existing tool-interaction tasks in τ^2-Bench (Barres et al., 2025), we construct defective requests and recovery information. Dataset construction uses Qwen3-32B (Yang et al., 2025) as a separate screening model, retaining tasks that succeed under full and assisted conditions and fail under no-ask requests. We then align one matched triplet per source task, yielding 103 unique tasks and 309 records per model. Evaluation uses strict tool-task success criteria; the eight comparison models are evaluated on the same aligned tasks.

Empirical findings. Table 1 shows positive SG and RG for all eight evaluated models on both SG-Bench and the adapted tool-task set. On the latter, the suppression gap spans 34.0–66.0 percentage points and recovery gain 38.8–54.4. Across both protocols, aggregate execution performance is lower under defective requests and improves after information restoration. These findings motivate a request-state-dependent control objective: ask, confirm, or challenge when clarification or correction is needed, and execute when the request is sufficiently specified. The next section develops V-SGRO

Table 1: **Three-condition execution diagnostics on SG-Bench and adapted τ^2-Bench.** Each condition contains 240 SG-Bench samples or 103 adapted tool tasks. F/B/A are pass counts for full, defective/no-ask, and assisted inputs; SG and RG quantify degradation and recovery in percentage points. E counts exceptions retained as failures.

	SG-Bench (self-constructed)						Adapted τ^2-Bench					
Model	**F**	**B**	**A**	**SG**	**RG**	**E**	**F**	**B**	**A**	**SG**	**RG**	**E**
Qwen3-8B	66	21	45	+18.8	+10.0	3	64	7	59	+55.3	+50.5	13
Qwen3-14B	59	17	101	+17.5	+35.0	0	76	8	64	+66.0	+54.4	0
DeepSeek-V3.2	39	2	17	+15.4	+6.2	0	53	9	51	+42.7	+40.8	60
Qwen3.6-Plus	101	18	56	+34.6	+15.8	0	60	12	59	+46.6	+45.6	3
Qwen3.5-397B-A17B	105	25	84	+33.3	+24.6	0	57	15	55	+40.8	+38.8	0
MiniMax-M2.5	95	21	68	+30.8	+19.6	0	68	13	65	+53.4	+50.5	0
DeepSeek-V4-Pro	8	3	4	+2.1	+0.4	0	40	5	46	+34.0	+39.8	0
GLM-5.2	16	7	20	+3.8	+5.4	0	47	7	51	+38.8	+42.7	55

F (Full): pass count under complete requests. **B (Defective/no-ask)**: pass count under defective requests, with clarification disallowed and direct execution required. **A (Assisted)**: pass count after supplying missing information or repairing erroneous constraints in the defective request. Each condition contains $n = 240$ SG-Bench samples or $n = 103$ adapted τ^2-Bench tasks. Passes use frozen-evidence content checks for SG-Bench and strict tool-task success criteria for adapted τ^2-Bench.

SG (Suppression gap): $100(\mathrm{F} - \mathrm{B})/n$, the full-to-defective performance drop. **RG (Recovery gain)**: $100(\mathrm{A} - \mathrm{B})/n$, the gain from information restoration. Both are in percentage points. **E (Exceptions)**: exception records across all three conditions ($3n = 720$ or 309, respectively), retained in the denominators and counted as failures.

to learn these conditional decisions from verification status and relative-quality differences among self-generated candidates.

3 Method: View-Specific Group-Relative Optimization

The preceding experiments reveal performance degradation under defective requests and recovery after information restoration. This motivates learning when to take protective action and when to execute. We learn from self-generated candidates using task-verifier feedback. Candidates differ both in relative quality and in whether they meet the verification standard: certified responses provide reinforcement targets, while non-perfect responses provide directions for improvement. We therefore propose View-Specific Group-Relative Optimization (V-SGRO), which combines certification, preference, and residual quality signals into a composite advantage for a unified policy update. Figure 3 illustrates the overall pipeline.

3.1 Learning Setup and Rollout Groups

Let x denote the visible request and its context, and let π_θ be the policy with parameters θ. The policy selects an action from asking, confirming, challenging, and executing, and generates the corresponding response. Complete, defective, and information-restored requests form separate inputs; each rollout group corresponds to one input condition.

For each x, the initial policy generates N responses:

$$\mathcal{R}(x) = \{(y_i, r_i)\}_{i=1}^{N}, \qquad r_i = V(x, y_i) \in [0, 1], \tag{2}$$

where y_i is a complete response and V is a task verifier. We use $N = 8$, keep the candidate pool fixed during training, and freeze the initial policy as the reference π_{ref}. Responses are self-generated, while supervision is supplied by task rules and annotations. Verifier scores cover action selection, missing or conflicting dimensions, evidence and required content, and output format. Group-dependent quantities below are written with their dependence on x implicit.

The full-group reward mean and population standard deviation are

$$\mu_R = \frac{1}{N}\sum_{i=1}^{N} r_i, \qquad \sigma_R = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(r_i - \mu_R)^2}. \tag{3}$$

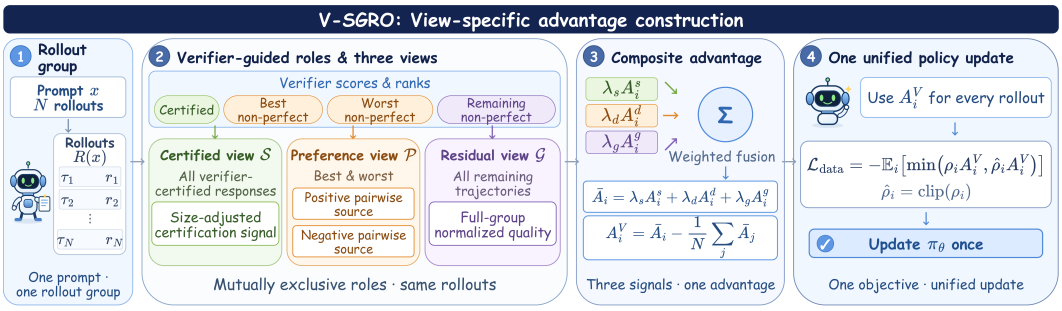

Figure 3: **V-SGRO: from a shared rollout group to one policy update.** (1) Generate and score $N = 8$ responses for each input. (2) Partition candidates into certified, preference, and residual views: all verifier-certified responses, a best–worst non-perfect pair when valid, and the remaining responses, respectively. (3) Weight and jointly center the three source advantages, with gradient alignment controlling the residual contribution. (4) Optimize one policy using the resulting composite advantage and the reference-deviation penalty.

Group-normalized rewards express relative quality (Shao et al., 2024). Their invariance to admissible positive affine transformations motivates explicitly retaining absolute verification status alongside relative quality. Appendix A.2 gives the derivation.

3.2 Verifier-Guided Candidate Roles

Let $\mathcal{I} = \{1, \ldots, N\}$ index the candidates. We first collect all candidates meeting the full-score threshold:

$$\mathcal{S} = \{i \in \mathcal{I} : r_i \geq 1 - \delta\}, \qquad s = |\mathcal{S}|, \qquad \delta = 10^{-6}. \tag{4}$$

These form the certified view. The non-perfect pool is $\mathcal{U} = \mathcal{I} \setminus \mathcal{S}$.

When $|\mathcal{U}| \geq 2$, select

$$c = \arg\max_{i \in \mathcal{U}} r_i, \qquad d = \arg\min_{i \in \mathcal{U}} r_i, \tag{5}$$

resolving ties by the smallest original candidate index. A preference pair is valid only when $r_c > r_d$. Set $\mathcal{P} = \{c, d\}$ for a valid pair and $\mathcal{P} = \varnothing$ otherwise; thus $n_d = |\mathcal{P}| \in \{0, 2\}$. The residual set is $\mathcal{G} = \mathcal{U} \setminus \mathcal{P}$. These three mutually exclusive sets cover the entire group. If no valid pair exists, all non-perfect candidates enter $\mathcal{G}$.

3.3 View-Specific Source Advantages

We assign source advantages A_i^s, A_i^d, and A_i^g to the certified, preference, and residual views. Throughout, $\mathbf{1}[\cdot]$ denotes an indicator, $\text{sg}[\cdot]$ stops gradients, and $\text{clip}(u, l, h)$ restricts u to $[l, h]$.

Certified view. The certification source is

$$A_i^s = \frac{N}{\max(1, s)} \mathbf{1}[i \in \mathcal{S}]. \tag{6}$$

For $s > 0$, this size compensation gives $N^{-1} \sum_i A_i^s = 1$, explicitly encoding verification status. An empty certified set yields a zero source.

Preference view. Let T_i be the number of supervised response tokens. Define the response-average log probability

$$\ell_i(\theta) = \frac{1}{T_i} \sum_{t=1}^{T_i} \log \pi_\theta(y_{i,t} \mid x, y_{i,<t}), \tag{7}$$

where $y_{i,t}$ is token t and $y_{i,<t}$ its response prefix. The reference uses the same reduction to obtain ℓ_i^{ref}. For a valid pair, define

$$h = \beta_{\text{p}} \left[(\ell_c - \ell_c^{\text{ref}}) - (\ell_d - \ell_d^{\text{ref}}) \right], \qquad \beta_{\text{p}} = 2, \tag{8}$$

$$\omega = \text{clip}(r_c - r_d, 0.1, 2). \tag{9}$$

With $\mathrm{sigmoid}(u) = (1 + e^{-u})^{-1}$, the preference source is

$$A_i^d = \begin{cases} \dfrac{N}{2}\,\mathrm{sg}[\omega\beta_{\mathrm{p}}\,\mathrm{sigmoid}(-h)]\,\big(\mathbf{1}[i=c] - \mathbf{1}[i=d]\big), & n_d = 2, \\ 0, & n_d = 0. \end{cases} \tag{10}$$

The endpoints receive equal-magnitude, opposite-sign coefficients. The gate decreases as the current policy favors the better candidate more strongly relative to the reference. The score-gap weight is applied after pair validity has been established.

Residual view. The remaining candidates retain full-group relative quality signals:

$$z_i = \begin{cases} 0, & \sigma_R \le \epsilon_\sigma, \\ \mathrm{clip}\left(\dfrac{r_i - \mu_R}{\sigma_R}, -M, M\right), & \sigma_R > \epsilon_\sigma, \end{cases} \qquad A_i^g = \mathbf{1}[i \in \mathcal{G}] z_i, \tag{11}$$

where $\epsilon_\sigma = 10^{-12}$ and $M = 2.5$. Residual candidates share the original full-group statistics and retain their group-relative scale. Appendix A.3 derives the source scaling and preference gate.

3.4 Composite Advantage and Unified Objective

Let $\lambda_s, \lambda_d, \lambda_g$ be the view weights. We combine the sources and center them over the complete group:

$$\bar{A}_i = \lambda_s A_i^s + \lambda_d A_i^d + \lambda_g A_i^g, \qquad A_i^V = \mathrm{sg}\left[\bar{A}_i - \frac{1}{N}\sum_{j=1}^{N} \bar{A}_j\right]. \tag{12}$$

The resulting coefficients satisfy $\sum_i A_i^V = 0$ and enter the objective directly. In vector form, the centering operator is $P = I - \mathbf{1}\mathbf{1}^\top/N$, where I is the $N \times N$ identity matrix and $\mathbf{1}$ the all-ones vector. Writing $\mathbf{A}^k = (A_1^k, \ldots, A_N^k)^\top$ gives $\mathbf{A}^V = \mathrm{sg}[P \sum_{k\in\{s,d,g\}} \lambda_k \mathbf{A}^k]$.

Define the fixed-reference ratio

$$\rho_i(\theta) = \exp\big[\mathrm{clip}(\ell_i - \ell_i^{\mathrm{ref}}, -20, 20)\big]. \tag{13}$$

For a microbatch $\mathcal{B}$ of rollout groups and groupwise advantages $\mathbf{A}$, the data objective adopts the clipped-surrogate form of PPO (Schulman et al., 2017):

$$\mathcal{L}_{\mathrm{data}}(\theta; \mathbf{A}) = -\frac{1}{|\mathcal{B}|}\sum_{x\in\mathcal{B}} \frac{1}{N}\sum_{i=1}^{N} \min\Big\{\rho_i\,\mathrm{sg}[A_i], \mathrm{clip}(\rho_i, 1-\epsilon_c, 1+\epsilon_c)\,\mathrm{sg}[A_i]\Big\}, \tag{14}$$

where $\epsilon_c = 0.2$. The unified loss adds a squared reference-deviation penalty:

$$\mathcal{L}(\theta) = \mathcal{L}_{\mathrm{data}}(\theta; \mathbf{A}^V) + \frac{\eta}{|\mathcal{B}|}\sum_{x\in\mathcal{B}} \frac{1}{N}\sum_{i=1}^{N} \frac{(\ell_i - \ell_i^{\mathrm{ref}})^2}{2}, \qquad \eta = 0.05. \tag{15}$$

Scores, routing, source coefficients, and view weights are detached for the current update. Gradients flow through policy probabilities and the reference-deviation term. Appendix A.4 specifies numerical conventions and boundary cases.

3.5 Gradient-Based Residual Weighting

Gradient similarity has been used to adapt auxiliary learning signals (Du et al., 2018). We fix the certified and preference weights and adjust the residual weight using gradient alignment. For each microbatch, compute the unweighted source data gradients

$$g_k = \nabla_\theta \mathcal{L}_{\mathrm{data}}(\theta; P\mathbf{A}^k), \qquad k \in \{s, d, g\}. \tag{16}$$

Each source is centered within its full group and evaluated with unit weight. Define

$$C(g_a, g_b) = \frac{\langle g_a, g_b\rangle}{\max\{\|g_a\|_2\|g_b\|_2, \epsilon_{\cos}\}}, \qquad \epsilon_{\cos} = 10^{-12}, \tag{17}$$

where $\langle \cdot, \cdot \rangle$ is the inner product and $\| \cdot \|_2$ the Euclidean norm. The gate is

$$m_g = \mathbf{1}\left[\min_{k \in \{s,d,g\}} \|g_k\|_2 > \tau_{\text{norm}}\right] \cdot \mathbf{1}[C(g_s, g_g) \geq \tau_{\cos}] \cdot \mathbf{1}[C(g_d, g_g) \geq \tau_{\cos}], \tag{18}$$

with $\tau_{\text{norm}} = 10^{-8}$ and $\tau_{\cos} = 0.3$. We use

$$\lambda_s = \lambda_d = 1, \qquad \lambda_g = \max\{0.05, 0.10 m_g\}. \tag{19}$$

The residual source receives weight 0.10 when the gate is active and retains weight 0.05 otherwise. Weights are shared across groups within each microbatch. After weight selection, the composite objective supplies the gradients used for parameter optimization. Appendix A.5 presents the training procedure, and Appendix A.6 analyzes when the composite advantage provides a common ascent direction for the three objectives in a fixed-candidate model.

4 Experiments and Results

We aim to answer four questions:

- **Does V-SGRO better decide when to clarify and when to execute?**
- **Do these gains transfer from SG-Bench to TAU2-SG without further training? Do these improvements carry over to new tasks?**
- **How does removing each learning signal or the residual-weight floor affect decisions? Which training components improve the model's decisions?**
- **Do preference and residual signals improve decisions on conflicting preferences and vague goals?**

4.1 Experimental Setup

Model and datasets. We use Qwen3-8B (Yang et al., 2025) throughout the main experiments. SG-Bench contains 720 three-condition control records. TAU2-SG, derived from τ^2-Bench (Barres et al., 2025), contains 121 request variants from 103 unique source tasks, yielding 363 first-turn records across full, bad, and assisted conditions. MultiWOZ2.4-derived (Budzianowski et al., 2018; Ye et al., 2022) contains 360 requests evenly split between asking and executing. The fixed self-generated pools contain 3,456 SG-Bench trajectories and 17,336 MultiWOZ trajectories, with eight candidates per group. V-SGRO uses LoRA (Hu et al., 2021) with rank 16, a learning rate of 5×10^{-7}, and seed 42.

Baselines. We compare SFT variants, DPO variants (Rafailov et al., 2023), SimPO (Meng et al., 2024), ORPO (Hong et al., 2024), and reward-only GRPO (Shao et al., 2024).

Metrics and evaluation protocols. **NeedAcc** measures the accuracy of deciding between protection and execution, **TypedAction** measures exact action-type accuracy, **PremExec** measures premature execution when execution should be withheld, and **OverAsk** measures unnecessary protective actions on executable requests.

4.2 Comparisons across Task Sources

Controlled decisions and external-source transfer. To evaluate both decisions under controlled defects and their transfer across task sources, we jointly assess SG-Bench and TAU2-SG. SG-Bench tests responses to different request states, whereas TAU2-SG evaluates the same checkpoints on external task sources without additional training.

Table 2 shows control accuracies of 87.50% and 97.25% for V-SGRO, exceeding ORPO, the strongest baseline by control accuracy, by 10.56 and 28.65 pp. Both premature execution and over-asking decrease, while exact action accuracy also improves. Thus, the complete configuration's advantage spans both sides of the decision: fewer premature executions under defective requests and fewer unnecessary protective actions when execution is appropriate.

Table 2: **Protective-control comparisons on SG-Bench and TAU2-SG (%).** TAU2-SG evaluates first-turn transfer of the same SG-Bench checkpoints without additional training. Control accuracy (NeedAcc) tests protection versus execution, whereas typed accuracy (TypedAction) tests the exact action. Premature execution is measured on protection-required requests; over-asking is measured on executable requests.

	SG-Bench				TAU2-SG			
Method	Control acc.↑	Typed acc.↑	Premature exec.↓	Over-asking↓	Control acc.↑	Typed acc.↑	Premature exec.↓	Over-asking↓
Base / Zero-shot	75.83	63.75	30.00	21.25	67.22	46.56	31.40	33.47
Standard SFT	76.53	64.86	29.17	20.62	67.77	46.83	31.40	32.64
Matched explicit SFT	76.25	64.58	30.42	20.42	68.32	47.38	30.58	32.23
Typed-DPO	75.42	63.61	30.83	21.46	67.49	46.83	32.23	32.64
SimPO	76.11	63.61	29.58	21.04	67.77	46.83	30.58	33.06
ORPO	76.94	64.31	27.08	21.04	68.60	47.38	29.75	32.23
Reward-only GRPO	76.11	63.75	30.00	20.83	68.32	47.38	30.58	32.23
V-SGRO	**87.50**	**76.94**	**15.83**	**10.83**	**97.25**	**70.25**	**1.65**	**3.31**

Table 3: **MultiWOZ2.4-derived control comparisons (%).** NeedAcc measures ask/execute accuracy on all 360 requests; PremExec and OverAsk use the 180 ask-required and 180 executable requests, respectively.

Method	NeedAcc ↑	PremExec ↓	OverAsk ↓
Base Qwen3-8B	73.61	26.11	26.67
Standard SFT	73.06	23.33	30.56
Typed-DPO	73.06	24.44	29.44
Multi-pair DPO	76.67	22.22	**24.44**
SimPO	71.11	30.56	27.22
ORPO	74.17	23.89	27.78
Margin-weighted DPO	70.83	27.22	31.11
Reward-only GRPO	72.50	27.22	27.78
V-SGRO	**79.72**	**13.33**	27.22

Independent training on dialogue tasks. To further test the learning approach under different request formulations and a different action space, we train independently on MultiWOZ-derived tasks and compare methods under identical test prompts. Its balanced ask/execute setting directly connects the two error types to overall accuracy. V-SGRO reaches 79.72% control accuracy (Table 3), exceeding Reward-only GRPO and Multi-pair DPO by 7.22 and 3.06 pp. Relative to Multi-pair DPO, it makes 16 fewer premature executions and five more unnecessary inquiries, yielding 11 additional correct decisions overall. Its dialogue-task advantage therefore comes from a larger reduction in premature execution than the increase in over-asking, complementing the controlled-defect and external-transfer evaluations.

4.3 Multi-View Signal Ablations

To examine the role of each learning signal in binary control, we hold the MultiWOZ candidate groups and reward statistics fixed and zero the certified, preference, or residual source before common centering. We additionally remove the residual-weight floor while retaining the residual source.

Table 4: **MultiWOZ source and weight-floor ablations (%).** Candidate groups and reward statistics are fixed; variants remove the indicated source before centering or the residual-weight floor. Full uses the predictions in Table 3.

Configuration	NeedAcc ↑	PremExec ↓	OverAsk ↓
Full V-SGRO	**79.72**	**13.33**	27.22
w/o certified	76.11	19.44	28.33
w/o preference	78.89	16.11	**26.11**
w/o residual	72.22	27.22	28.33
w/o residual floor	72.78	27.22	27.22

Removing the certified, preference, and residual signals reduces control accuracy by 3.61, 0.83, and 7.50 pp (Table 4). Preference removal avoids two unnecessary inquiries but introduces five premature executions, resulting in three fewer correct decisions overall. Removing the residual-weight floor introduces 25 premature executions without changing over-asking. Full V-SGRO therefore achieves the fewest total errors in this comparison. Residual quality and its positive weight floor primarily suppress premature execution, while the preference source improves the balance between premature execution and unnecessary clarification.

4.4 Behavioral Attribution on Difficult Requests

To locate the request types responsible for the aggregate differences, we analyze MultiWOZ's preference-conflict and hard-ask subsets. The former contains 45 requests with explicit conflicting preferences; the latter adds 45 vague-goal requests, all requiring an ask decision.

Table 5: **Control accuracy on difficult MultiWOZ requests.** Preference-conflict contains 45 conflicting-preference requests; hard-ask adds 45 vague-goal requests. Each row compares Full with removal of the indicated source.

Removed source	Request subset	Full (%)	After removal (%)	Difference (pp)
Preference	Preference-conflict ($n = 45$)	91.11	64.44	+26.67
Residual	Hard-ask ($n = 90$)	73.33	45.56	+27.78

Preference removal produces 12 fewer correct decisions on the conflict subset but nine more on the other requests, explaining its net loss of only three decisions overall. For residual removal, 25 of the 27 net lost correct decisions occur on hard-ask requests (Table 5). These decompositions locate the observed gains: preference effects concentrate on conflicting requests, while residual effects primarily concern difficult clarification decisions, connecting candidate-role design to specific control behaviors.

5 Related Work

5.1 Preference Optimization and Group-Relative Training

DPO (Rafailov et al., 2023) optimizes pairwise preferences; SimPO (Meng et al., 2024) simplifies the objective, while ORPO (Hong et al., 2024) combines preference learning with supervised adaptation. GRPO (Shao et al., 2024) constructs group-relative advantages; Dr. GRPO (Liu et al., 2025) and GSPO (Zheng et al., 2025) study normalization and sequence-level optimization. For clarification, Zhang et al. (2025) derive preferences from simulated future conversation outcomes.

5.2 Active Clarification and Protective Decisions

CLAMBER benchmarks ambiguity identification and clarification (Zhang et al., 2024); alignment adapts responses to model-perceived ambiguity (Kim et al., 2024). Retrieval uncertainty guides questions (Chi et al., 2024), intent uncertainty informs their utility (Zhang & Choi, 2025), and tool-parameter specification uncertainty guides questions and GRPO (Suri et al., 2026). AskBench combines interactive evaluation with rubric-guided RL from verifier rewards for intent-deficient and false-premise queries (Zhao et al., 2026); other work interleaves reasoning and clarification (Chen et al., 2026). Analyses examine ambiguity recognition–behavior gaps, post-clarification answering bottlenecks, and long-horizon clarification timing (Su & Cardie, 2026; Su & Healey, 2026; Gulati et al., 2026). Selective prediction studies error–coverage trade-offs and unnecessary abstention (Srinivasan et al., 2024; Rabanser, 2025); clarification acquires information for execution.

V-SGRO unifies certification, verifier-scored non-perfect preferences, and residual quality in one centered advantage per fixed self-generated group. Evaluation separates initial protection from execution after prescribed information restoration.

6 Conclusion

We formalized the suppression gap and introduced a three-condition diagnosis of performance degradation under defective requests and recovery after information restoration. Motivated by this phenomenon, V-SGRO learns protective decisions from self-generated candidates by combining certification, preference, and residual quality signals in a unified policy update. Evaluations across controlled requests, external task sources, and independently trained dialogue tasks show improved control accuracy for the full configuration. Source ablations and request-level analyses locate the observed gains in reduced premature execution and better decisions on difficult requests. Together, these results connect request-state diagnosis with role-aware learning from a model's own trajectories.

AI USE STATEMENT

Generative AI tools were used during manuscript preparation for language editing, organization, and LaTeX assistance. All scientific ideas, experimental design, implementation, data analysis, and conclusions were developed, checked, and approved by the authors, who take full responsibility for the content of this paper.

ETHICS STATEMENT

This work does not involve human-subject experiments or the collection of private personal data. SG-Bench is constructed from public tables and documents with frozen evidence, and the external evaluations use established benchmark sources. Because clarification policies can still over-ask or execute prematurely, the reported results should be interpreted as controlled research evidence rather than a guarantee of safe deployment. Deployment in consequential settings requires task-specific validation and appropriate human oversight.

REPRODUCIBILITY STATEMENT

We report the model backbone, dataset and split sizes, fixed-candidate construction, evaluation contracts, verifier definition, optimization objective, and key training hyperparameters in the main text and appendices. The appendices additionally provide the SG-Bench source composition and split protocol, V-SGRO training procedure and boundary cases, and the MultiWOZ generation, training, and evaluation prompts. The main experiments use a fixed random seed of 42.

REFERENCES

Victor Barres, Honghua Dong, Soham Ray, Xujie Si, and Karthik Narasimhan. τ^2-Bench: Evaluating conversational agents in a dual-control environment. *arXiv preprint arXiv:2506.07982*, 2025.

Paweł Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Iñigo Casanueva, Stefan Ultes, Osman Ramadan, and Milica Gasic. Multiwoz-a large-scale multi-domain wizard-of-oz dataset for task-oriented dialogue modelling. In *Proceedings of the 2018 conference on empirical methods in natural language processing*, pp. 5016–5026, 2018.

Xin Chen, Feng Jiang, Yiqian Zhang, Hardy Chen, Shuo Yan, Wenya Xie, Min Yang, and Shujian Huang. Reasoning while asking: Transforming reasoning large language models from passive solvers to proactive inquirers. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 35069–35090, 2026.

Yizhou Chi, Jessy Lin, Kevin Lin, and Dan Klein. Clarinet: Augmenting language models to ask clarification questions for retrieval. *arXiv preprint arXiv:2405.15784*, 2024.

Yunshu Du, Wojciech M Czarnecki, Siddhant M Jayakumar, Mehrdad Farajtabar, Razvan Pascanu, and Balaji Lakshminarayanan. Adapting auxiliary losses using gradient similarity. *arXiv preprint arXiv:1812.02224*, 2018.

Anmol Gulati, Hariom Gupta, Elias Lumer, Sahil Sen, and Vamse Kumar Subbiah. Ask early, ask late, ask right: When does clarification timing matter for long-horizon agents? *arXiv preprint arXiv:2605.07937*, 2026.

Jiwoo Hong, Noah Lee, and James Thorne. Orpo: Monolithic preference optimization without reference model. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 11170–11189, 2024.

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

Hyuhng Joon Kim, Youna Kim, Cheonbok Park, Junyeob Kim, Choonghyun Park, Kang Min Yoo, Sang-goo Lee, and Taeuk Kim. Aligning language models to explicitly handle ambiguity. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 1989–2007, 2024.

Zichen Liu, Changyu Chen, Wenjun Li, Penghui Qi, Tianyu Pang, Chao Du, Wee Sun Lee, and Min Lin. Understanding r1-zero-like training: A critical perspective. *arXiv preprint arXiv:2503.20783*, 2025.

Chaitanya Malaviya, Joseph Chee Chang, Dan Roth, Mohit Iyyer, Mark Yatskar, and Kyle Lo. Contextualized evaluations: Judging language model responses to underspecified queries. *Transactions of the Association for Computational Linguistics*, 13:878–900, 2025.

Yu Meng, Mengzhou Xia, and Danqi Chen. Simpo: Simple preference optimization with a reference-free reward. *Advances in Neural Information Processing Systems*, 37:124198–124235, 2024.

Stephan Rabanser. *Uncertainty-driven reliability: Selective prediction and trustworthy deployment in modern machine learning*. PhD thesis, University of Toronto (Canada), 2025.

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

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

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

Avi Singh, John D Co-Reyes, Rishabh Agarwal, Ankesh Anand, Piyush Patil, Xavier Garcia, Peter J Liu, James Harrison, Jaehoon Lee, Kelvin Xu, et al. Beyond human data: Scaling self-training for problem-solving with language models. *arXiv preprint arXiv:2312.06585*, 2023.

Tejas Srinivasan, Jack Hessel, Tanmay Gupta, Bill Yuchen Lin, Yejin Choi, Jesse Thomason, and Khyathi Chandu. Selective "selective prediction": Reducing unnecessary abstention in vision-language reasoning. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 12935–12948, 2024.

Jinyan Su and Claire Cardie. Knowing but not showing: Llms recognize ambiguity but rarely ask clarifying questions. *arXiv preprint arXiv:2605.25284*, 2026.

Jinyan Su and Jennifer Healey. Clarification is not enough: Post-clarification answering remains the bottleneck in multi-turn qa. *arXiv preprint arXiv:2605.25204*, 2026.

Manan Suri, Puneet Mathur, Nedim Lipka, Franck Dernoncourt, Ryan A Rossi, and Dinesh Manocha. Structured uncertainty guided clarification for llm agents. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 40811–40838, 2026.

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

Fanghua Ye, Jarana Manotumruksa, and Emine Yilmaz. Multiwoz 2.4: A multi-domain task-oriented dialogue dataset with essential annotation corrections to improve state tracking evaluation. In *Proceedings of the 23rd Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pp. 351–360, 2022.

Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah Goodman. Star: Bootstrapping reasoning with reasoning. *Advances in Neural Information Processing Systems*, 35:15476–15488, 2022.

Michael Zhang, W Bradley Knox, and Eunsol Choi. Modeling future conversation turns to teach llms to ask clarifying questions. In *International Conference on Learning Representations*, volume 2025, pp. 60722–60742, 2025.

Michael JQ Zhang and Eunsol Choi. Clarify when necessary: Resolving ambiguity through interaction with lms. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pp. 5541–5558, 2025.

Tong Zhang, Peixin Qin, Yang Deng, Chen Huang, Wenqiang Lei, Junhong Liu, Dingnan Jin, Hongru Liang, and Tat-Seng Chua. Clamber: A benchmark of identifying and clarifying ambiguous information needs in large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 10746–10766, 2024.

Jiale Zhao, Ke Fang, and Lu Cheng. When and what to ask: Askbench and rubric-guided rlvr for llm clarification. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 17120–17140, 2026.

Chujie Zheng, Shixuan Liu, Mingze Li, Xiong-Hui Chen, Bowen Yu, Chang Gao, Kai Dang, Yuqiong Liu, Rui Men, An Yang, Jingren Zhou, and Junyang Lin. Group sequence policy optimization, 2025. URL https://arxiv.org/abs/2507.18071.

A Additional Method Details and Analysis

A.1 Task Verifier

The verifier assigns

$$r_{\mathrm{SG}} = \mathrm{clip}_{[0,1]}\big(0.35a + 0.25b + 0.15e + 0.20m + 0.05f - 0.25p\big), \tag{A.1}$$

where $\mathrm{clip}_{[0,1]}$ restricts the result to $[0, 1]$. The score components are defined in Table A.1.

Table A.1: **Task-verifier components for SG-Bench responses.** The terms in Equation (A.1) assess action choice, relevant dimensions, evidence citations, required content, output format, and prohibited content. Their weighted sum is clipped to $[0, 1]$ to obtain the recorded reward used for candidate-role assignment.

Term	Definition
a	Action score: 1 for an exact action match, 0.5 for a correct protection/execution family only, and 0 otherwise.
b	F1 between predicted and annotated dimension sets under defective requests; 1 under complete or information-restored requests.
e	Indicator that all required evidence identifiers are cited.
m	Coverage rate of required textual content, taking values in $[0, 1]$.
f	Indicator that the complete output contract is successfully parsed.
p	Indicator that at least one prohibited item appears; multiple matches still give $p = 1$.

The dimension score b uses missing, boundary, and conflict dimensions for MCP, MIB, and MD, respectively. These defect types correspond to missing critical premises, unclear intent boundaries, and erroneous constraints. When the annotated action is asking, confirming, or challenging, set $e = m = 1$. The response contract comprises `ACTION`, `MISSING_DIMENSIONS`, `CONFLICT_DIMENSIONS`, `EVIDENCE`, and `ANSWER`.

The certification threshold is $r_i \geq 1 - 10^{-6}$. Scores are computed before training and fixed for candidate routing and source construction.

A.2 Absolute Certification and Group Normalization

Positive affine invariance. Let $\mathbf{r} = (r_1, \ldots, r_N)^\top$ have $\sigma_R > 0$, and define

$$z(\mathbf{r}) = \frac{\mathbf{r} - \mu_R \mathbf{1}}{\sigma_R}. \tag{A.2}$$

Consider $r'_i = ar_i + b$, where $a > 0$ and the transformed scores remain in the valid range. Their mean and standard deviation satisfy

$$\mu'_R = a\mu_R + b, \qquad \sigma'_R = a\sigma_R. \tag{A.3}$$

Consequently,

$$\frac{r'_i - \mu'_R}{\sigma'_R} = \frac{a(r_i - \mu_R)}{a\sigma_R} = \frac{r_i - \mu_R}{\sigma_R}, \qquad z(a\mathbf{r} + b\mathbf{1}) = z(\mathbf{r}). \tag{A.4}$$

Certification is determined by a fixed threshold. An admissible downward shift can change the certified set while preserving the standardized vector, distinguishing absolute certification from relative quality.

Group dependence of a full-score candidate. For a candidate $i^\star$ with $r_{i^\star} = 1$, define the mean score of the other candidates as

$$\mu_{-i^\star} = \frac{1}{N-1} \sum_{j \neq i^\star} r_j. \tag{A.5}$$

Then

$$1 - \mu_R = \frac{N-1}{N}(1 - \mu_{-i^\star}). \tag{A.6}$$

Thus, its group-relative coefficient depends on the other scores. Let $d_i = r_i - \mu_R$. Since $\sum_i d_i = 0$, the Cauchy–Schwarz inequality gives

$$\sum_{j \neq i^\star} d_j^2 \geq \frac{d_{i^\star}^2}{N-1}, \qquad N\sigma_R^2 \geq \frac{N}{N-1} d_{i^\star}^2. \tag{A.7}$$

As $d_{i^\star} \geq 0$, it follows that

$$0 \leq \frac{1-\mu_R}{\sigma_R} \leq \sqrt{N-1}. \tag{A.8}$$

V-SGRO supplements these relative statistics with a source coefficient determined by membership in the certified set.

A.3 Source Scaling and the Preference Gate

Source scaling. For a nonempty certified set,

$$\frac{1}{N}\sum_{i=1}^{N} A_i^s = \frac{1}{N}\, s\, \frac{N}{s} = 1. \tag{A.9}$$

Once the certified set is fixed, its source coefficient depends only on its size. For a valid preference pair, the factor $N/2$ gives

$$\frac{1}{N}\sum_{i=1}^{N} |A_i^d| = \omega\beta_{\mathrm{p}}\, \mathrm{sigmoid}(-h). \tag{A.10}$$

Preference gate. Consider the weighted pairwise objective based on DPO (Rafailov et al., 2023):

$$\mathcal{L}_{\mathrm{pair}} = -\omega \log \mathrm{sigmoid}(h). \tag{A.11}$$

With scores, reference probabilities, and pair identities fixed,

$$\nabla_\theta \mathcal{L}_{\mathrm{pair}} = -\omega\beta_{\mathrm{p}}\, \mathrm{sigmoid}(-h)\big(\nabla_\theta \ell_c - \nabla_\theta \ell_d\big). \tag{A.12}$$

The derivative yields equal-magnitude, opposite-sign endpoint signals. V-SGRO extracts their coefficients as the preference source, applies the group-size compensation, and combines them with the other sources. The gap h is recomputed under the current policy at each step, and the source coefficients are detached when used in the unified objective.

A.4 Implementation Details and Boundary Cases

Probability reduction and numerical settings. Supervised response positions exclude input and padding tokens and include the empty thinking-marker tokens introduced by the response template. The current and reference policies use the same response positions and reduction. The log-probability difference is clipped to $[-20, 20]$ before exponentiation, and the surrogate ratio interval is $[0.8, 1.2]$. The same minimum expression applies to positive and negative advantages. The reference penalty is $0.05(\ell_i - \ell_i^{\mathrm{ref}})^2/2$. Both loss terms are averaged within each group and then across groups.

The reference, candidates, scores, and routing remain fixed; current response probabilities and preference gates are recomputed during training. Full-group centering and residual standardization follow Equations (12) and (11).

Controller implementation. Source gradients use unit weights and the centered data objective. The admission rule in Section 3.5 uses current microbatch gradients, detached and converted to FP32. A gradient-norm exponential moving average with decay 0.95 is maintained for logging.

Source removal and boundary cases. A source-removal ablation sets its coefficients to zero before common centering while retaining candidate routing and full-group reward statistics. An empty certified set gives $\mathbf{A}^s = 0$. Without a valid preference pair, $\mathbf{A}^d = 0$ and $\mathcal{G} = \mathcal{U}$. When all candidates are certified, $s = N$ and each certification coefficient is 1, yielding $\mathbf{A}^V = 0$ after centering. When all candidates have the same sub-threshold score, certification and preference sources vanish, $\sigma_R = 0$, and the residual source is also zero. In the latter two cases, the objective retains the reference penalty.

A.5 Training Procedure

Table A.2 summarizes V-SGRO training, from fixed-pool preprocessing to the unified parameter update.

Table A.2: **V-SGRO training with fixed candidate groups and verifier scores.** Each microbatch constructs the three source advantages, uses source-gradient alignment to set the residual weight, and forms a centered composite advantage. Gradients of the unified objective, including the reference-deviation penalty, are accumulated before optimizer updates.

Step	Operation
Input	Fixed self-generated pool $\{\mathcal{R}(x)\}$ with verifier scores; initial policy π_{θ_0}; frozen reference π_{ref}; optimizer; gradient-accumulation factor K.
1	For each group, assign $\mathcal{S}$, $\mathcal{P}$, and $\mathcal{G}$; cache μ_R, σ_R, and the valid-pair endpoints and score-gap weight ω.
2	Initialize $\theta \leftarrow \theta_0$ and clear accumulated optimization gradients.
3	For each microbatch $\mathcal{B}$ of rollout groups:
4	Compute $\ell_i(\theta)$ and ℓ_i^{ref} for every response, then compute h for each valid pair.
5	Construct $\mathbf{A}^s$, $\mathbf{A}^d$, and $\mathbf{A}^g$ using Equations (6), (10), and (11).
6	Evaluate the unweighted source data gradients g_s, g_d, g_g and determine m_g and λ_g using Equations (18)–(19).
7	Combine the sources and center within each group to obtain $\mathbf{A}^V$ using Equation (12).
8	Evaluate the unified loss in Equation (15) and accumulate its parameter gradients.
9	After every K microbatches, apply one optimizer update and clear the accumulated optimization gradients.
10	End for.
Output	Trained policy π_θ.

A.6 Joint-Progress Analysis

Analytical setting. To isolate the effect of advantage construction, we analyze a fixed-candidate softmax model with $N = 8$. Candidates, scores, routing, and the reference are fixed. Independently parameterized candidate logits $\mathbf{u} = (u_1, \ldots, u_N)^\top$ define

$$\pi_i = \frac{e^{u_i}}{\sum_{j=1}^{N} e^{u_j}}. \tag{A.13}$$

In this model, $\mathcal{S}$ contains all exact full-score candidates with $r_i = 1$. Reference logits are denoted by $\mathbf{u}^{\mathrm{ref}}$. For a valid preference pair, define

$$\begin{aligned} h_u &= 2\big[(u_c - u_d) - (u_c^{\mathrm{ref}} - u_d^{\mathrm{ref}})\big], \\ q &= 8\omega\,\mathrm{sigmoid}(-h_u), \end{aligned} \tag{A.14}$$

where q is the magnitude of each preference endpoint's source coefficient. Let $\gamma \in [0.05, 0.10]$ be the residual weight. The composite advantage is

$$\mathbf{A} = P\left[\frac{8}{s}\mathbf{1}_{\mathcal{S}} + q(\mathbf{e}_c - \mathbf{e}_d) + \gamma\mathbf{A}^g\right], \tag{A.15}$$

where $\mathbf{1}_{\mathcal{S}}$ is the indicator vector of $\mathcal{S}$ and $\mathbf{e}_i$ is the ith standard basis vector. Write $L_A = \|\mathbf{A}\|_2$. We study the unit-direction flow induced by the composite advantage

$$\dot{\mathbf{u}} = \frac{\mathbf{A}}{L_A}, \tag{A.16}$$

where dots denote derivatives with respect to flow time.

Proposition 1 (Joint progress across three views)**.** *Suppose the initial and reference logits are finite, $1 \le s \le 4$, and $\sigma_R > 10^{-12}$. Assume a valid preference pair exists, residual rewards are nonconstant, and initially $q \le 4/s$. Define*

$$\begin{aligned} p_{\mathcal{S}} &= \sum_{i\in\mathcal{S}} \pi_i, \qquad p_{c|cd} = \frac{\pi_c}{\pi_c + \pi_d}, \\ Q_{\mathcal{G}} &= \sum_{j\in\mathcal{G}} p_{j|\mathcal{G}} r_j, \qquad p_{j|\mathcal{G}} = \frac{\pi_j}{\sum_{k\in\mathcal{G}} \pi_k}. \end{aligned} \tag{A.17}$$

Along the flow in Equation (A.16)*,*

$$\dot{p}_{\mathcal{S}} \geq \frac{4}{sL_A} p_{\mathcal{S}}(1 - p_{\mathcal{S}}) > 0, \tag{A.18}$$

$$\dot{p}_{c|cd} = \frac{2q}{L_A} p_{c|cd}(1 - p_{c|cd}) > 0, \tag{A.19}$$

$$\dot{Q}_{\mathcal{G}} = \frac{\gamma}{L_A \sigma_R} \operatorname{Var}_{p_{\cdot|\mathcal{G}}}(r) > 0, \tag{A.20}$$

where the variance is evaluated under the conditional distribution within $\mathcal{G}$*.*

Under these conditions, the composite advantage provides a common ascent direction for certified probability mass, endpoint preference, and residual conditional quality in the fixed-candidate model. The analysis characterizes the interaction of the three sources at the advantage level. The first-order representation and proof follow.

A.6.1 First-Order Representation

Define the stop-gradient candidate-level objective

$$J(\mathbf{u}) = \frac{1}{8} \sum_{i=1}^{8} \operatorname{sg}[A_i] \log \pi_i. \tag{A.21}$$

Using

$$\frac{\partial \log \pi_i}{\partial u_j} = \mathbf{1}[i = j] - \pi_j \tag{A.22}$$

and $\sum_i A_i = 0$ gives

$$\frac{\partial J}{\partial u_j} = \frac{1}{8} \sum_i A_i \big(\mathbf{1}[i = j] - \pi_j\big) = \frac{1}{8} A_j. \tag{A.23}$$

Thus, $\nabla_{\mathbf{u}} J = \mathbf{A}/8$, and the unit flow follows the same composite-advantage direction.

A.6.2 Residual Standardized Scores

Let $v_i = (r_i - \mu_R)/\sigma_R$. Then $\sum_i v_i = 0$ and $\sum_i v_i^2 = 8$. If at least $k < 8$ coordinates are at least $a > 0$, let their sum be $U \geq ka$. The Cauchy–Schwarz inequality yields

$$8 \geq \frac{U^2}{k} + \frac{U^2}{8 - k} \geq \frac{8k}{8 - k} a^2, \tag{A.24}$$

so $a \leq \sqrt{(8 - k)/k}$. Applying the same argument to $-\mathbf{v}$ gives the lower-tail bound.

For any $j \in \mathcal{G}$ with $v_j < 0$, both j and the lowest-scoring endpoint d are no greater than v_j, hence $v_j \geq -\sqrt{3}$. If $v_j > 0$, the s certified candidates, the highest-scoring endpoint c, and j give $s + 2$ coordinates at least as large as v_j. Therefore,

$$-\sqrt{3} \leq v_j \leq \sqrt{\frac{6 - s}{s + 2}} < 2.5. \tag{A.25}$$

Residual coordinates fall within the clipping interval, so $z_j = v_j$ and $A_j^g = (r_j - \mu_R)/\sigma_R$ for $j \in \mathcal{G}$.

A.6.3 Certified Probability Mass

Write the coefficients before centering as

$$b_i = \frac{8}{s} \mathbf{1}[i \in \mathcal{S}] + q\big(\mathbf{1}[i = c] - \mathbf{1}[i = d]\big) + \gamma \mathbf{1}[i \in \mathcal{G}] z_i. \tag{A.26}$$

Common centering preserves coordinate differences: $A_i - A_j = b_i - b_j$. A certified coordinate has coefficient $8/s$. Among non-certified candidates, the preferred endpoint has coefficient $q \leq 4/s$, the rejected endpoint has coefficient $-q$, and each residual coefficient is at most $\gamma M \leq 1/4$. Hence

$$\min_{i \in \mathcal{S},\, j \notin \mathcal{S}} (A_i - A_j) \geq \frac{4}{s}. \tag{A.27}$$

This also gives $L_A > 0$. As shown in Appendix A.6.4, q decreases along the flow, preserving its initial upper bound.

The softmax derivative of the set probability is

$$\dot{p}_{\mathcal{S}} = \frac{p_{\mathcal{S}}(1 - p_{\mathcal{S}})}{L_A} \left(\mathbb{E}_\pi[A \mid \mathcal{S}] - \mathbb{E}_\pi[A \mid \mathcal{S}^c] \right), \tag{A.28}$$

where $\mathcal{S}^c = \mathcal{I} \setminus \mathcal{S}$, and each conditional expectation is the probability-weighted mean over its set. Substituting Equation (A.27) yields Equation (A.18).

A.6.4 Preference Endpoints

The two endpoints receive their direct signals from the preference source, giving

$$A_c - A_d = 2q. \tag{A.29}$$

Differentiating their conditional softmax yields

$$\dot{p}_{c|cd} = \frac{2q}{L_A} p_{c|cd}(1 - p_{c|cd}) > 0. \tag{A.30}$$

Since reference logits are fixed,

$$\dot{h}_u = 2(\dot{u}_c - \dot{u}_d) = \frac{4q}{L_A} > 0. \tag{A.31}$$

Consequently, $q = 8\omega \,\mathrm{sigmoid}(-h_u)$ decreases with time, so the initial condition $q \leq 4/s$ is maintained.

A.6.5 Residual Conditional Quality

For $j, k \in \mathcal{G}$, common centering cancels and Appendix A.6.2 gives

$$A_j - A_k = \gamma(z_j - z_k) = \frac{\gamma}{\sigma_R}(r_j - r_k). \tag{A.32}$$

Differentiating $Q_{\mathcal{G}}$ gives

$$\dot{Q}_{\mathcal{G}} = \frac{\gamma}{L_A} \mathrm{Cov}_{p_{\cdot|\mathcal{G}}}(r, z) = \frac{\gamma}{L_A \sigma_R} \mathrm{Var}_{p_{\cdot|\mathcal{G}}}(r), \tag{A.33}$$

where covariance and variance use the conditional distribution within $\mathcal{G}$. Finite logits yield positive conditional probabilities, and nonconstant residual rewards yield positive variance. Hence $\dot{Q}_{\mathcal{G}} > 0$, completing the proof of Proposition 1. □

B SG-BENCH SOURCES AND CONSTRUCTION

B.1 PUBLIC SOURCES AND FROZEN EVIDENCE

SG-Bench is constructed from public tables and documents. The pipeline selects and retrieves public sources, cleans and freezes their contents locally, binds evidence snippets to tasks, constructs complete and defective requests, annotates recovery information and expected actions, defines verifier checks, and exports the three-condition protocols. Evaluation reads the frozen evidence associated with each request. Table B.1 reports the source composition of the diagnostic collection, `v0_240`.

Table B.1: **Public-source composition of the SG-Bench diagnostic collection.** Categories cover frozen tables and public documentation used to construct evidence-grounded tasks. The reported URL counts total 80; one MCP, one MIB, and one MD sample per source yield 240 samples before expansion into the three request conditions.

Recorded source group	URLs	Samples	Content and evaluation role
`public_csv_table_data`	30	90	Public tables and CSV data for evidence-grounded data analysis.
`devops_infra_database_docs`	13	39	Operations, database, and infrastructure documentation for engineering-rule execution.
`github_product_docs`	11	33	GitHub API, workflow, and repository-governance documentation.
`web_platform_http_docs`	8	24	MDN HTTP, CORS, and CSP documentation for web-protocol scenarios.
`language_packaging_docs`	8	24	Python, PyPA, and npm language and package-management documentation.
`security_accessibility_standards`	5	15	OWASP and W3C accessibility materials for security and standards-related constraints.
`payment_tax_policy_docs`	5	15	Stripe and IRS materials for payment and policy-related tasks.
Total	**80**	**240**	

For each source, missing critical premises (MCP) remove execution-relevant information; missing intent boundaries (MIB) remove scope or output requirements; and misconstraint damage (MD) introduces a constraint that conflicts with the frozen evidence. Their expected protective actions are asking, confirming, and challenging, respectively. Recovery information supplies the missing specification or repairs the conflicting constraint, after which the expected action is execution.

B.2 DIAGNOSTIC COLLECTION AND METHOD SPLITS

The diagnostic collection and the method test collection each contain 240 samples but serve different evaluations. The diagnostic collection, `v0_240`, contains 80 source keys, with one MCP, one MIB, and one MD sample per key; its three conditions produce 720 full/no-ask/assisted records. The method test collection, `main_large_heldout`, combines 96 samples from the original development/test portion with 144 additionally constructed samples. It also contains 80 source keys and 80 samples of each defect type, producing 720 full/bad/assisted control records. The two collections share 96 sample IDs.

The method training collection has 144 samples from 48 source keys and 432 condition groups, yielding 3,456 fixed candidates. The method test collection contains 90 `document_table` samples and 150 `frozen_web_evidence` samples. Training and method testing have disjoint sample IDs, source keys, and complete-request texts. Two source keys refer to the same Python `argparse` documentation URL, with different frozen versions and three samples each, and four defective-request strings recur across the splits. No complete saved message sequence is identical across these splits.

The diagnostic content checker removes thinking sections and checks complete coverage of required evidence IDs, at least 75% coverage of required mentions, and absence of prohibited phrases. Control evaluation separately scores the selected action family and exact action type using the metrics in Section 4.1.

Table B.2: **SG-Bench collections and their counting units.** Each sample expands into three condition records; each training record forms an eight-response rollout group, so 432 training groups yield 3,456 candidates. The diagnostic and method-test collections serve different evaluations and share 96 sample IDs.

Collection	Source keys	Samples	Condition records
Diagnostic collection (`v0_240`)	80	240	720
Method training collection	48	144	432
Method test collection (`main_large_heldout`)	80	240	720

C Prompt Records and Evaluation Protocols

This appendix presents the MultiWOZ candidate-generation, training, and evaluation prompts and the TAU2 evaluation protocols.

C.1 MultiWOZ Candidate Generation and Training

The following preference-conflict request is an original sampled example. Candidate generation and training share the system message below. The training user message removes the single sentence `Do not output hidden reasoning, JSON, bullets, or tables.` from the generation message; all other text, the query, responses, and rewards remain unchanged across the 17,336 training candidates.

MultiWOZ generation and training — shared system

```
You are a UCC-MST task-oriented dialogue assistant. Use only the visible
query. Write the monitoring ledger, then decide whether to ask or execute.
```

MultiWOZ candidate generation — original user message

```
[Query] I want a cheap five-star hotel in the centre, and do not ask me
to choose between price and stars.
[Task] Decide whether the visible request is executable or requires
clarification.
[Output Contract] Return only the assistant answer. Do not output hidden
reasoning, JSON, bullets, or tables. Use exactly these six headers in
order: [Query Evidence], [Decision Trace], [Dimension Ledger], [Missing
Dimensions], [VoI Control], [Action]. The four allowed missing-dimension
IDs are V1 Goal Specificity, V2 Constraint Completeness, V3 Scope
Boundedness, and V4 Preference Resolution. Use this decision rule: if the
 query names a supported service domain such as hotel, restaurant,
attraction, train, or taxi, V1 is present; if it names a domain but omits
 concrete constraints, mark V2 only; mark V3 only for partial or
unbounded multi-domain trip scope; mark V4 only for explicit conflicting
or unresolved preferences. [Missing Dimensions] must be exactly none or a
 comma-separated subset of V1, V2, V3, V4. [VoI Control] must be exactly
ask or execute.
```

MultiWOZ training — original user message

```
[Query] I want a cheap five-star hotel in the centre, and do not ask me
to choose between price and stars.
```

MultiWOZ training — original user message (continued)

```
[Task] Decide whether the visible request is executable or requires
clarification.
[Output Contract] Return only the assistant answer. Use exactly these six
 headers in order: [Query Evidence], [Decision Trace], [Dimension Ledger],
 [Missing Dimensions], [VoI Control], [Action]. The four allowed missing-
dimension IDs are V1 Goal Specificity, V2 Constraint Completeness, V3
Scope Boundedness, and V4 Preference Resolution. Use this decision rule:
if the query names a supported service domain such as hotel, restaurant,
attraction, train, or taxi, V1 is present; if it names a domain but omits
 concrete constraints, mark V2 only; mark V3 only for partial or
unbounded multi-domain trip scope; mark V4 only for explicit conflicting
or unresolved preferences. [Missing Dimensions] must be exactly none or a
 comma-separated subset of V1, V2, V3, V4. [VoI Control] must be exactly
ask or execute.
```

C.2 MULTIWOZ EVALUATION CONTRACT

The messages below are the first recorded test example. All reported MultiWOZ configurations use identical saved input messages on the 360 test records. The evaluation system message and user instruction are reproduced separately from the generation and training contracts.

MultiWOZ evaluation — system

```
You are a UCC-MST task-oriented dialogue assistant. Write the monitoring
ledger, then decide whether to ask or execute.
```

MultiWOZ evaluation — original user message

```
[Query] I need a hotel with moderate price range, free parking, 3 stars.
[Instruction] Output exactly these sections: [Query Evidence], [Decision
Trace], [Dimension Ledger], [Missing Dimensions], [VoI Control], [Action].
 The four allowed missing-dimension IDs are V1 Goal Specificity, V2
Constraint Completeness, V3 Scope Boundedness, and V4 Preference
Resolution. Use this decision rule: if the query names a supported
service domain such as hotel, restaurant, attraction, train, or taxi, V1
is present; if it names a domain but omits concrete constraints, mark V2
only; mark V3 only for partial or unbounded multi-domain trip scope; mark
 V4 only for explicit conflicting or unresolved preferences. [Missing
Dimensions] must be exactly none or a comma-separated subset of V1, V2,
V3, V4. [VoI Control] must be exactly ask or execute.
```

C.3 TAU2 EVALUATION PROTOCOLS

The TAU2-derived collection contains 121 request variants from 103 unique source tasks. For the execution diagnosis in Table 1, we group variants by `(domain, source_task_id)`, select the first no-ask variant in lexicographic `protocol_id` order, and match it with the corresponding assisted and full inputs, yielding 103 triplets and 309 records per model. TAU2-SG retains all 121 variants and expands each into full, bad, and assisted inputs, producing 363 first-turn control records. Its explicit response contract follows the SG-Bench structure, and the SG-Bench-trained checkpoints are evaluated without additional training, as described in Section 4.1.