

## Article

# SocraGate: Inference-Time Admission Control for Reducing Answer Leakage in Scaffolded LLM Tutoring

Wanfang Li and Keyun Zhao \*

School of Media and Communication, Qufu Normal University, Rizhao 276800, China; liwanfang26@163.com

\* Correspondence: zhaokeyun0907@163.com

## Abstract

Large language models can solve mathematics problems accurately yet compromise tutoring by revealing answers before learners complete the reasoning. Prompt- and reward-based alignment changes the response distribution but does not screen the particular utterance delivered to a learner. We therefore formulate answer leakage as an output-admission problem and introduce SocraGate, a two-stage framework that combines scaffold-oriented policy optimization with inference-time screening. A five-level scaffold-spectrum reward shapes candidate generation. Deterministic answer-equivalence rules and a reference-conditioned semantic classifier then screen complete candidates against a fixed leakage threshold. An empty admitted set triggers re-screened regeneration and, if necessary, a fixed probe. Across five MathDial training seeds, SocraGate reduces utterance-level telling rate (TR) from 0.168 for the fixed-simulator GRPO baseline to 0.032, while simulator-based Success@10 rises from 0.482 to 0.546. Under the same simulator-based protocol and without parameter updates, TR also falls on QATD-2k (0.159 to 0.047) and CoMTA (0.150 to 0.053). In a blind evaluation of 160 dialogues, SocraGate receives the highest mean rating on all four dimensions, including 4.78 out of 5 for non-leakage. The cumulative analysis shows the largest single-stage reduction in TR after deterministic screening, whereas guarded regeneration raises Success@10 from 0.521 to 0.546. These results support output admission as a practical complement to policy alignment for scaffolded LLM tutoring.

**Keywords:** answer leakage; controlled generation; dialogue tutoring; large language models; pedagogical alignment; selective generation

## 1. Introduction

A model can solve a mathematics problem correctly and still fail as a tutor. In dialogue tutoring, revealing the answer too early reduces opportunities to diagnose errors, attempt the next step, and consolidate a strategy. We call this premature disclosure *answer leakage*. It conflicts with scaffolding and the assistance dilemma because support should advance the learner without completing the learner's work [1–5]. The effectiveness of intelligent tutoring systems likewise depends on alignment between instructional support and learning outcomes [6]. Recent LLM tutoring research exposes the same tension. MathDial quantifies a trade-off between learner success and answer disclosure [7]. Pedagogical alignment and steering favor guided multi-turn support [8,9], while multi-turn reinforcement learning penalizes answer leakage [10].

Prompting, preference optimization, and reinforcement learning can reduce average leakage by reshaping the tutor policy. However, these approaches do not directly screen the sampled utterance before it reaches a learner. A policy may perform well on average



Academic Editor: Christos Bouras

Received: 7 September 2026

Revised: 12 October 2026

Accepted: 21 October 2026

Published: 25 October 2026

**Copyright:** © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

yet reveal an answer under stochastic sampling, unfamiliar problem formats, or persistent requests. Adversarial evaluation further shows that learners can increase leakage by actively attempting to extract solutions [11]. This gap raises our research question: can response-level admission reduce answer leakage while preserving useful scaffolding? Our objective, *Per-Step Alignment*, is to check each response before delivery, complementing the distributional preference learned during training.

SocraGate implements this objective through a training–inference decomposition. A scaffold-spectrum reward shapes the policy toward pedagogically useful candidates, while inference-time admission determines which complete utterances may be emitted. Each candidate passes through deterministic answer-equivalence rules and a reference-conditioned semantic leakage classifier. If the admitted set is empty, guarded regeneration produces new candidates that undergo the same checks before the system uses a fixed probe. This design adapts the screening and rejection principle of selective prediction to generated tutoring responses [12]. Here, rejection initiates a screened recovery process over complete utterances instead of terminating with abstention.

The paper makes three contributions.

- **Complete-response admission formulation.** We formulate answer leakage as an admission decision over complete candidate responses. Candidate-level classifier diagnostics are separated from end-to-end measures of emitted tutoring responses.
- **Training–inference control architecture.** SocraGate combines scaffold-oriented group-relative policy optimization (GRPO) with deterministic and semantic admission plus guarded regeneration. The same checks govern initial and regenerated candidates.
- **Multi-corpus and human evaluation.** We evaluate SocraGate on MathDial and transfer it without parameter updates to QATD-2k and CoMTA. The evaluation includes blind human ratings, component analysis, classifier sensitivity, residual failures, and inference latency.

## 2. Related Work

**Pedagogical alignment and tutor evaluation.** General-purpose LLMs are optimized to answer requests, whereas effective tutoring often requires eliciting the learner’s next step without disclosing the solution. The AI Teacher Test and BEA 2023 shared task provide contextual evaluation of generated tutor responses [13,14]. MathDial makes the answer–guidance tension measurable through multi-turn dialogues, pedagogical-move annotations, and reference solution steps [7]. Recent systems shape tutoring behavior through preference-based alignment, explicit pedagogical plans, and outcome-driven reinforcement learning [8–10,15]. Direct preference optimization supplies a general preference-learning mechanism rather than a tutor-specific objective [16]. Evaluation has also expanded beyond answer accuracy and lexical similarity. MRBench organizes pedagogical quality into learning-science dimensions, and the BEA 2025 shared task evaluates error diagnosis and guidance [17,18]. MathTutorBench covers scaffolding, Socratic questioning, and problem solving, while EducationQ evaluates teaching through simulated multi-agent dialogues [19,20].

Scaffolding provides a pedagogical basis for this emphasis on guidance: it involves contingent support, gradual fading, and transfer of responsibility [21]; autonomy support likewise emphasizes the learner’s active role [22]. SocraGate applies this principle specifically to answer disclosure: scaffold-oriented rewards favor focus and probing moves, while admission checks whether a completed response reveals the solution. Learner-state estimation and the timing of fading remain functions of the surrounding tutoring system.

**Student simulation and evaluation validity.** Outcome-driven tutor training often relies on a student simulator that reacts to candidate guidance and supplies a proxy out-

come. Bayesian knowledge tracing estimates latent mastery from observed attempts, while dialogue knowledge tracing extends student modeling to tutor–student interactions [23,24]. GPTEach instead uses simulated students for interactive teaching practice [25]. LLM simulators can condition on cognitive and non-cognitive profiles, although fidelity varies across traits and settings [26]. Evaluation on authentic dialogues also finds substantial discrepancies between prompted simulators and real learners [27]. Fine-tuning and preference optimization reduce, but do not eliminate, this gap. Simulator-based success therefore characterizes performance within a controlled interaction model. Authentic corpora provide complementary evidence across dialogue styles. QATD-2k contains question-anchored Eedi tutoring sessions, while CoMTA contains 188 anonymized Khanmigo mathematics-tutoring conversations [28,29]. Their different annotation schemas support transfer assessment while preserving corpus-specific metric definitions.

**Controlled generation and tutor-specific admission.** Controlled generation can intervene during decoding, after generation, or through policy learning. NeuroLogic enforces lexical and logical constraints during autoregressive decoding [30]. FUDGE reweights next-token probabilities using predicted future attributes, whereas DExperts and GeDi contrast conditional distributions [31–33]. Constitutional AI, Sparrow, Self-Refine, and constrained policy optimization use feedback, critique, preferences, or policy constraints to improve adherence [34–37]. Selective prediction formalizes rejection through coverage–risk trade-offs, confidence thresholds, and learned reject options [12,38,39]. Sequence generation typically requires sequence-level confidence or an auxiliary evaluator, while calibrated risk guarantees require additional held-out assumptions [40,41].

Tutor-specific systems move closer to complete-utterance control. MWPTutor combines finite-state pedagogy with automated checks and repeated generation [42]. Kadir mediates completed proposals through deterministic and optional semantic checks, while Pisan uses a solution-code detector and an LLM judge [43,44]. TRAVER combines knowledge tracing with turn-by-turn verification for coding tutors [45]. The UnMask project restricts retrieval of answer-bearing chunks according to learner mastery and also checks response grounding [46]. HeuristicEdu shapes Socratic guidance through supervised warm-up and GRPO [47]. Across these approaches, control targets learner adaptation, information access, policy shaping, or output checking. SocraGate couples scaffold-oriented policy shaping with hybrid, reference-conditioned admission and repeated screening after regeneration. Table 1 compares selected tutor-specific output controls while keeping their tasks and evaluation protocols distinct.

**Table 1.** Mechanism-level comparison of tutor-specific output controls. The table identifies where control enters each pipeline and how each system handles rejection.

| System        | Policy or Contract                            | Screened Object and External Check                                         | Rejection Path                               | Reported Evaluation Emphasis                           |
|---------------|-----------------------------------------------|----------------------------------------------------------------------------|----------------------------------------------|--------------------------------------------------------|
| MWPTutor [42] | Expert-defined finite-state pedagogy          | State-conditioned utterances with task-specific automated checks           | Repeated generation until checks pass        | Human ratings on two mathematics word-problem datasets |
| Kadir [43]    | Authorization-aware disclosure contract       | Completed proposal with deterministic and optional semantic checks         | Action-specific deterministic fallback       | Fixed proposals and prospective attack sequences       |
| Pisan [44]    | Per-turn ceiling on an eight-rung help ladder | Risky reply with a solution-code detector and an LLM judge                 | Revision-oriented handling and escalation    | Automated scripted personas and model-based auditing   |
| SocraGate     | Scaffold-spectrum reward with GRPO            | Complete candidate with answer-equivalence rules and a semantic classifier | Re-screened regeneration, then a fixed probe | Multi-corpus transfer and blind human ratings          |

### 3. Method

SocraGate separates candidate-distribution shaping from response admission. During training, a scaffold-spectrum reward favors candidates that advance the learner’s reasoning without disclosing the answer. During inference, deterministic rules and a semantic classifier screen each complete candidate before selection. An empty admitted set initiates scaffold-restricted regeneration, and every regenerated candidate undergoes the same checks. Figure 1 summarizes this training–inference decomposition.

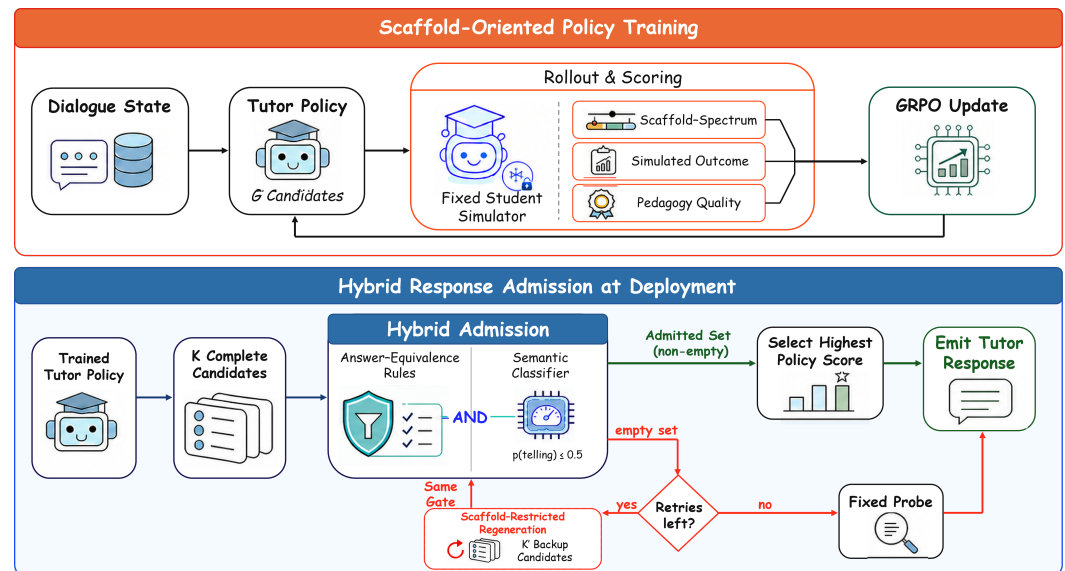


**Figure 1.** SocraGate training and inference workflow. During training, scaffold-spectrum, simulated-outcome, and pedagogical-quality rewards shape the tutor’s candidate distribution through GRPO. At deployment, deterministic answer-equivalence rules and a semantic leakage classifier jointly screen complete candidates. Empty admitted sets trigger scaffold-restricted regeneration through the same gate; the system uses a fixed probe only when no retry remains. The output labeled  $p(\text{telling})$  in the schematic is the sigmoid leakage score  $s_\psi$ . Orange and blue panels denote training and deployment, respectively. Green arrows indicate selection from a non-empty admitted set; red arrows indicate regeneration or fixed-probe fallback.

#### 3.1. Problem Formulation and Evaluation Events

Consider a corpus of  $N$  tutoring dialogues. Dialogue  $i$  contains  $M_i$  tutor turns and is represented as

$$\tau_i = (q_i, u_{i,0}, t_{i,1}, u_{i,1}, \dots, t_{i,M_i}, u_{i,M_i}), \quad (1)$$

where  $q_i$  is the problem and  $u_{i,0}$  is the learner’s initial utterance. The variables  $t_{i,m}$  and  $u_{i,m}$  denote the tutor and subsequent learner utterances at turn  $m$ , respectively. Reference information  $r_i$  contains the final answer and any available worked solution steps.

Let  $\ell(t; q, r) \in \{0, 1\}$  indicate whether response  $t$  explicitly states or semantically reveals the reference answer. For each emitted tutor utterance, the evaluation protocol assigns  $y_{i,m} = \ell(t_{i,m}; q_i, r_i)$  separately from the admission decision. The utterance-level telling rate (TR) is

$$\text{TR} = \frac{\sum_{i=1}^N \sum_{m=1}^{M_i} y_{i,m}}{\sum_{i=1}^N M_i}. \quad (2)$$

Let  $c_{i,m} \in \{0, 1\}$  indicate whether the learner response  $u_{i,m}$  correctly solves  $q_i$ , and let  $T_i = \min(T, M_i)$ . Leakage-free success within  $T$  tutor turns is

$$\text{Success@ } T = \frac{1}{N} \sum_{i=1}^N \mathbb{I} \left[ \exists m \in \{1, \dots, T_i\} : c_{i,m} = 1 \wedge \sum_{j=1}^m y_{i,j} = 0 \right]. \quad (3)$$

A dialogue counts as successful if the learner gives a correct solution at some evaluated turn and no tutor response up to and including that turn reveals the answer. The leakage condition includes turn  $m$  because  $t_{i,m}$  precedes  $u_{i,m}$ .

### 3.2. Training-Stage Candidate Shaping

For dialogue  $i$ , the history before tutor turn  $m$  is

$$h_{i,m} = (q_i, u_{i,0}, t_{i,1}, u_{i,1}, \dots, t_{i,m-1}, u_{i,m-1}).$$

We omit the dialogue index below when the context is unambiguous. At each history, the old tutor policy samples a group of  $G$  complete candidates  $t_m^{(k)} \sim \pi_{\Phi_{\text{old}}}(\cdot \mid h_m)$ . Each candidate initializes a rollout in which the fixed student simulator  $\pi_S$  supplies subsequent learner utterances. The tutor policy continues the dialogue until the simulator solves the problem or the turn limit is reached. The simulator response model and decoding configuration remain fixed across tutor policies.

The pedagogical-move evaluator  $\rho$  maps each candidate to one of the five MathDial moves. We convert these labels into a scaffold-spectrum reward,

$$r_{\text{scaf}}(t) = \begin{cases} 1.0, & \rho(t) = \text{focus}, \\ 0.7, & \rho(t) = \text{probing}, \\ 0.3, & \rho(t) = \text{telling\_strategy}, \\ 0.0, & \rho(t) = \text{generic}, \\ -1.0, & \rho(t) = \text{telling\_answer}. \end{cases} \quad (4)$$

Focus and probing responses receive the largest positive rewards because they preserve learner participation. Telling-strategy responses retain partial credit, whereas direct answer disclosure receives a negative reward. This weighting encodes a preference for guided participation over solution delivery.

For candidate  $t_m^{(k)}$ , we set  $r_{\text{scaf}}^{(k)} = r_{\text{scaf}}(t_m^{(k)})$ . Two additional terms evaluate the rollout that this candidate initiates. The binary outcome reward  $r_{\text{out}}^{(k)}$  equals one when the simulator reaches a correct solution before the turn limit. The pedagogical score  $r_{\text{ped}}^{(k)} \in [0, 1]$  averages seven normalized rubric dimensions. These dimensions are dialogue consistency, relevance, personalization, scaffold use, error diagnosis, support for next-step progress, and fluency. Their weighted sum yields one sequence-level reward for the initiating candidate,

$$R^{(k)} = w_s r_{\text{scaf}}^{(k)} + w_o r_{\text{out}}^{(k)} + w_p r_{\text{ped}}^{(k)} \quad (5)$$

with  $(w_s, w_o, w_p) = (1.0, 0.5, 0.5)$ . For each group, GRPO forms the standardized advantage

$$A^{(k)} = \frac{R^{(k)} - \bar{R}}{\text{sd}(R) + \epsilon_A}, \quad (6)$$

where  $\bar{R}$  and  $\text{sd}(R)$  are the group mean and standard deviation. The positive constant  $\epsilon_A$  prevents division by zero.

Let  $L_k$  be the token length of candidate  $k$ . For token  $\ell$ , the importance ratio is

$$\eta_{k,\ell}(\Phi) = \frac{\pi_{\Phi}(t_{m,\ell}^{(k)} \mid h_m, t_{m,<\ell}^{(k)})}{\pi_{\Phi_{\text{old}}}(t_{m,\ell}^{(k)} \mid h_m, t_{m,<\ell}^{(k)})}. \quad (7)$$

For a sampled candidate group at history  $h_m$ , the empirical token-averaged GRPO objective is

$$\mathcal{J}_{\text{GRPO}}(\Phi) = \frac{1}{G} \sum_{k=1}^G \frac{1}{L_k} \sum_{\ell=1}^{L_k} \left[ \min(\eta_{k,\ell} A^{(k)}, \text{clip}(\eta_{k,\ell}, 1 - \epsilon_c, 1 + \epsilon_c) A^{(k)}) - \beta \hat{D}_{\text{KL}}^{(k,\ell)}(\pi_{\Phi} \parallel \pi_{\text{ref}}) \right], \quad (8)$$

where  $\epsilon_c$  is the clipping range and  $\beta$  controls regularization toward the reference policy  $\pi_{\text{ref}}$  [48]. The rollout-level advantage  $A^{(k)}$  is applied to every token in candidate  $k$ . The admission gate does not contribute gradients to this objective; it operates after the policy generates complete candidate utterances.

### 3.3. Hybrid Candidate Admission

Policy shaping changes the response distribution but does not verify an individual sample. SocraGate therefore evaluates each complete candidate using deterministic and semantic checks. Let  $B_{\mathcal{R}}(a, q, r) \in \{0, 1\}$  indicate whether candidate  $a$  triggers any rule in the collection  $\mathcal{R}$ . The collection contains 18 answer-equivalence checks: eight canonical answer constructions, seven numerical equivalence rules, and three symbolic tests. The numerical rules cover identity, fraction–decimal, percentage, sign, rounding, unit, and scientific-notation equivalence. The symbolic tests cover simplified-expression equality, polynomial-coefficient equality, and equation-side reversal.

Numeric parsing uses `decimal.Decimal`. We set comparison tolerances of  $10^{-3}$  for fraction–decimal equivalence and  $5 \times 10^{-4}$  for rounding. A parse failure produces no rule-based decision, so the candidate remains subject to semantic classification. The reference-conditioned semantic classifier assigns

$$s_{\psi}(a; q, r) = \sigma(\text{MLP}_{\psi}(\text{Encoder}([a; q; r]))), \quad (9)$$

where the encoder jointly represents the candidate, question, and reference, and the MLP maps this representation to a sigmoid score. We use this output as a leakage score rather than a calibrated probability, given the class weighting and sampled class proportions used in training. With  $z = 1$  denoting a telling candidate, the classifier is trained using class-weighted binary cross-entropy,

$$\mathcal{L}_{\text{tell}} = -\mathbb{E}[w_+ z \log s_{\psi} + w_- (1 - z) \log (1 - s_{\psi})], \quad (10)$$

where  $s_{\psi}$  abbreviates  $s_{\psi}(a; q, r)$ , the expectation is over labeled classifier-training examples, and  $w_+$  and  $w_-$  are the class weights. Training settings are given in Section 4.2.

For a threshold  $\theta$ , scores above  $\theta$  are classified as telling. The admission indicator is

$$G_{\theta}(a; q, r) = \mathbb{I}[B_{\mathcal{R}}(a, q, r) = 0 \wedge s_{\psi}(a; q, r) \leq \theta]. \quad (11)$$

Thus, a score equal to  $\theta$  remains admissible when no deterministic rule fires. For a candidate set  $C$ , the admitted set is  $S_{\theta}(C) = \{a \in C : G_{\theta}(a; q, r) = 1\}$ . We fix the operational threshold at  $\theta = 0.5$ . If  $S_{\theta}(C)$  is non-empty, the selected candidate maximizes the length-normalized policy score

$$g_{\Phi}(a \mid h_m, c) = \frac{1}{|a|} \log \pi_{\Phi}(a \mid h_m, c), \quad (12)$$

where  $|a|$  is the number of generated tokens and  $c$  is the generation constraint. The initial candidate group uses the empty constraint  $c = \emptyset$ .

### 3.4. Guarded Regeneration and Emission

An empty admitted set indicates that the initial samples fail the operational admission criterion. SocraGate then applies a scaffold-restricted instruction  $c_{\text{scaf}}$  and samples  $K'$

backup candidates. The instruction requests the learner's next reasoning step without supplying the final answer. Each backup set passes through the same rule collection, classifier, and threshold. The procedure stops at the first non-empty admitted set or after  $K_{\max}$  failed backup rounds.

Candidate ranking uses the same conditioning context that generated the corresponding set. If every backup round remains empty, SocraGate emits a fixed problem-independent probe  $t_{\text{probe}}$ . This fallback asks the learner to identify the next step and contains no problem-specific answer content. Algorithm 1 specifies the complete inference procedure. Each turn evaluates at most  $K + K_{\max}K'$  candidates.

---

**Algorithm 1** SocraGate inference
 

---

**Require:** policy  $\pi_{\Phi}$ ; history  $h_m$ ; question  $q$ ; reference  $r$ ; rules  $\mathcal{R}$ ; classifier  $s_{\psi}$ ; threshold  $\theta$ ; counts  $K, K'$ ; retries  $K_{\max}$ ; constraint  $c_{\text{scaf}}$ ; probe  $t_{\text{probe}}$

**Ensure:** tutor response  $t_m$

```

1: function ADMIT( $C; q, r, \theta$ )
2:   return  $\{a \in C : G_{\theta}(a; q, r) = 1\}$ 
3: end function
4:  $c \leftarrow \emptyset$ 
5:  $C \leftarrow \text{Sample}(\pi_{\Phi}(\cdot | h_m, c), K)$ 
6:  $S \leftarrow \text{Admit}(C; q, r, \theta)$ 
7:  $b \leftarrow 0$ 
8: while  $S = \emptyset$  and  $b < K_{\max}$  do
9:    $b \leftarrow b + 1$ 
10:   $c \leftarrow c_{\text{scaf}}$ 
11:   $C \leftarrow \text{Sample}(\pi_{\Phi}(\cdot | h_m, c), K')$ 
12:   $S \leftarrow \text{Admit}(C; q, r, \theta)$ 
13: end while
14: if  $S = \emptyset$  then
15:   $t_m \leftarrow t_{\text{probe}}$ 
16: else
17:   $t_m \leftarrow \arg \max_{a \in S} g_{\Phi}(a | h_m, c)$ 
18: end if
19: return  $t_m$ 

```

---

## 4. Experiments

The experiments examine four aspects of SocraGate. First, we test whether output admission reduces answer leakage while preserving simulator-based learner success. Second, we evaluate whether the observed ordering transfers to two external tutoring corpora. Third, we quantify the cumulative effects of deterministic rules, semantic admission, and guarded regeneration. Finally, we characterize classifier sensitivity, residual leakage, human ratings, and inference cost.

### 4.1. Evaluation Protocol and Datasets

**MathDial.** MathDial contains approximately 3000 teacher–student mathematics tutoring dialogues grounded in multi-step reasoning problems and annotated with pedagogical moves [7]. We use an 80/10/10 dialogue-level training, validation, and test split. The partition evaluates generalization to held-out dialogues under this split.

**External corpora.** QATD-2k contains authentic Eedi tutoring sessions anchored to the question attempted by each learner [28]. CoMTA contains anonymized student–Khanmigo mathematics tutoring conversations [29]. We apply the MathDial-trained tutor and gate without parameter updates. For gate inputs, each corpus's question field maps to  $q$ , and  $r$  contains only corpus-provided answer or solution information. Across all three corpora, Success@10 follows Equation (3) under the same simulator-based evaluation protocol.

#### 4.2. Baselines, Metrics, and Implementation

**Baselines.** All learned methods use Qwen2.5-7B-Instruct [49]. Reward-side comparisons include a pedagogical prompt [25], fixed-simulator DPO [15,16], and fixed-simulator GRPO [10,48]. Inference-time comparisons include NeuroLogic, FUDGE, and GeDi adapted to answer-leakage control [30,31,33]. Self-revise implements a critique-and-rewrite loop based on Self-Refine and Constitutional AI [34,36]. These inference-time methods operate on the same GRPO-trained tutor policy. SocraGate uses the scaffold-spectrum policy trained with Equation (5). Table 2 therefore compares complete systems, while the component analysis evaluates cumulative admission stages on the SocraGate policy.

**Table 2.** MathDial tutoring results. Values are mean  $\pm$  standard deviation over five seeds. The best mean in each column is bold. Success@10 is simulator-based. Pedagogy is the author-defined seven-dimension automatic score.

| Method           | BERTScore $\uparrow$                | Move-Match $\uparrow$               | Sim. Succ.@10 $\uparrow$            | TR $\downarrow$                     | Pedagogy $\uparrow$                 |
|------------------|-------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|
| Markel prompt    | 0.601 $\pm$ 0.009                   | 0.612 $\pm$ 0.003                   | 0.398 $\pm$ 0.002                   | 0.246 $\pm$ 0.010                   | 0.641 $\pm$ 0.004                   |
| Fixed-sim. DPO   | 0.642 $\pm$ 0.007                   | 0.667 $\pm$ 0.008                   | 0.451 $\pm$ 0.006                   | 0.193 $\pm$ 0.008                   | 0.679 $\pm$ 0.007                   |
| Fixed-sim. GRPO  | 0.661 $\pm$ 0.006                   | 0.689 $\pm$ 0.001                   | 0.482 $\pm$ 0.009                   | 0.168 $\pm$ 0.007                   | 0.704 $\pm$ 0.006                   |
| NeuroLogic       | 0.658 $\pm$ 0.003                   | 0.691 $\pm$ 0.007                   | 0.489 $\pm$ 0.003                   | 0.094 $\pm$ 0.006                   | 0.708 $\pm$ 0.002                   |
| FUDGE            | 0.663 $\pm$ 0.005                   | 0.695 $\pm$ 0.001                   | 0.497 $\pm$ 0.008                   | 0.078 $\pm$ 0.005                   | 0.715 $\pm$ 0.005                   |
| GeDi             | 0.660 $\pm$ 0.006                   | 0.693 $\pm$ 0.004                   | 0.493 $\pm$ 0.002                   | 0.085 $\pm$ 0.006                   | 0.711 $\pm$ 0.006                   |
| Self-revise      | 0.665 $\pm$ 0.004                   | 0.701 $\pm$ 0.006                   | 0.502 $\pm$ 0.008                   | 0.069 $\pm$ 0.005                   | 0.722 $\pm$ 0.007                   |
| <b>SocraGate</b> | <b>0.687 <math>\pm</math> 0.005</b> | <b>0.731 <math>\pm</math> 0.001</b> | <b>0.546 <math>\pm</math> 0.007</b> | <b>0.032 <math>\pm</math> 0.004</b> | <b>0.798 <math>\pm</math> 0.003</b> |

$\uparrow$ : higher is better;  $\downarrow$ : lower is better. Bold numerical entries indicate the best mean in each column.

**Metrics.** BERTScore measures semantic similarity to a reference tutor response [50]. Move-match is the fraction of turns whose predicted pedagogical-move label equals the reference move. The auxiliary pedagogy score averages seven five-point rubric dimensions mapped linearly to  $[0, 1]$ : dialogue consistency, relevance, personalization, scaffold use, error diagnosis, next-step support, and fluency. We interpret pedagogy and move-match as alignment diagnostics because their rubric dimensions and move categories also define training objectives. Blind human ratings provide a complementary assessment of response quality. Simulator-based Success@10 follows Equation (3), and utterance-level telling rate (TR) follows Equation (2). TR summarizes leakage labels on emitted responses under the evaluation protocol, after deterministic rejection, guarded regeneration, score-based selection, and fixed-probe fallback; classifier precision and recall instead characterize candidate-level screening. Automatic results are reported as mean and standard deviation over five training seeds unless noted otherwise. These standard deviations quantify run-to-run variation; between-method comparisons describe observed mean differences.

**Configuration.** GRPO uses learning rate  $10^{-6}$ , batch size 32, and group size  $G = 8$ . The classifier MLP has hidden width 256 and is trained with AdamW for five epochs. Its learning rate is  $2 \times 10^{-4}$ , batch size is 64, and class-weight ratio is  $w_+/w_- = 2$ . Positive examples are *telling\_answer* turns. Negatives are sampled from the other four MathDial moves at a 1:2 positive-to-negative ratio. Approximately 7% of the classifier set consists of ambiguous boundary cases that underwent auxiliary relabeling with GPT-4o. Training uses eight A100 80-GB GPUs. At inference, SocraGate samples  $K = 8$  initial candidates and  $K' = 4$  candidates per backup round, with  $K_{\max} = 3$  retries. Generation uses temperature 0.7, nucleus probability 0.95, and a maximum output length of 256 tokens. We fixed the operational threshold at 0.5 before test evaluation and used it for all reported results.

#### 4.3. MathDial Results

SocraGate achieves the lowest mean TR and the highest mean on each reported utility metric in Table 2. Figure 2 visualizes the three metrics closest to the central claim. Relative to fixed-simulator GRPO, SocraGate reduces TR from 0.168 to 0.032. Simulator-based Success@10 increases from 0.482 to 0.546, while the pedagogy score increases from 0.704 to 0.798.

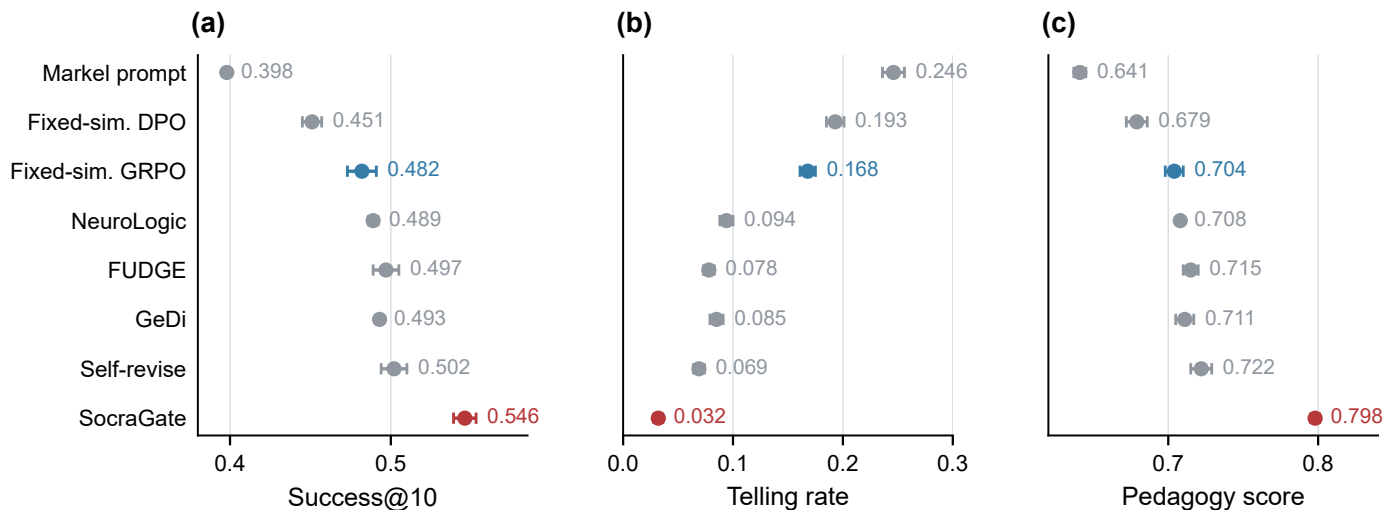


**Figure 2.** Main MathDial results. (a) simulator-based Success@10; (b) utterance-level telling rate; (c) author-defined pedagogy score. Points show means over five seeds, and horizontal error bars show standard deviations. SocraGate is red, fixed-simulator GRPO is blue, and other baselines are gray.

Self-revise has the best aggregate means among the inference-time comparators. SocraGate lowers TR from 0.069 to 0.032 and raises simulator-based Success@10 from 0.502 to 0.546. BERTScore increases from 0.665 to 0.687, move-match from 0.701 to 0.731, and pedagogy from 0.722 to 0.798. Thus, all reported aggregate means improve under the shared automatic evaluation protocol. Blind non-leakage ratings (Section 4.5) assess perceived response quality, while residual-error analysis (Section 4.7) characterizes failure types among identified leaks; these complement automatic TR without providing an independent binary estimate of it.

The distinction between guidance and disclosure is also relevant to multistep problems. For illustration, solving  $3(2x - 5) = 21$  requires two divisions and an addition. A prompt asking which operation to undo first leaves these steps to the learner, whereas stating  $x = (21/3 + 5)/2$  supplies an answer-equivalent expression. Such cases motivate combining equivalence rules with semantic screening. The aggregate results do not distinguish performance by reasoning-step count, operation count, or expression depth.

#### 4.4. Cross-Corpus Evaluation

The MathDial ordering is preserved on both external corpora without parameter updates (Figure 3a,b). On QATD-2k, SocraGate reduces TR from 0.159 to 0.047 relative to fixed-simulator GRPO. Its simulator-based Success@10 increases from 0.461 to 0.512, and pedagogy increases from 0.668 to 0.741. Relative to FUDGE, the corresponding changes are  $-0.054$ ,  $+0.026$ , and  $+0.039$ .

On CoMTA, SocraGate reduces TR from 0.150 to 0.053 relative to fixed-simulator GRPO. Simulator-based Success@10 increases from 0.447 to 0.490, and pedagogy increases from 0.651 to 0.717. Table 3 reports all values. The consistent ordering supports transfer across corpus-specific inputs under the shared simulator-based protocol.

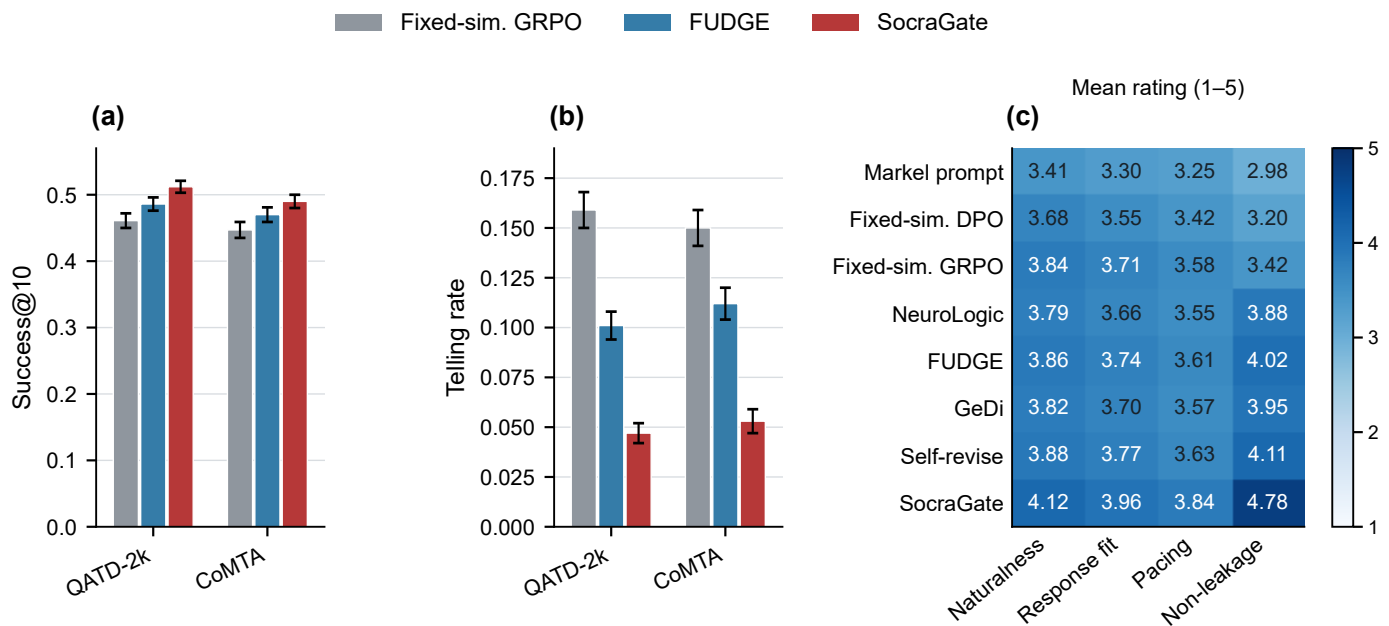


**Figure 3.** Cross-corpus results and blind ratings. (a) simulator-based Success@10 on QATD-2k and CoMTA; (b) utterance-level TR on the same corpora. Bars show means, and error bars show standard deviations over five seeds. (c) mean blind ratings for 20 dialogues per method, with six raters per dialogue.

**Table 3.** Cross-corpus evaluation without parameter updates. Values are mean  $\pm$  standard deviation over five seeds. Success@10 is simulator-based.

| QATD-2k          |                                     |                                     |                                     |
|------------------|-------------------------------------|-------------------------------------|-------------------------------------|
| Method           | Sim. Success@10 $\uparrow$          | TR $\downarrow$                     | Pedagogy $\uparrow$                 |
| Fixed-sim. GRPO  | 0.461 $\pm$ 0.011                   | 0.159 $\pm$ 0.009                   | 0.668 $\pm$ 0.008                   |
| FUDGE            | 0.486 $\pm$ 0.010                   | 0.101 $\pm$ 0.007                   | 0.702 $\pm$ 0.007                   |
| <b>SocraGate</b> | <b>0.512 <math>\pm</math> 0.009</b> | <b>0.047 <math>\pm</math> 0.005</b> | <b>0.741 <math>\pm</math> 0.006</b> |
| CoMTA            |                                     |                                     |                                     |
| Method           | Sim. Success@10 $\uparrow$          | TR $\downarrow$                     | Pedagogy $\uparrow$                 |
| Fixed-sim. GRPO  | 0.447 $\pm$ 0.012                   | 0.150 $\pm$ 0.009                   | 0.651 $\pm$ 0.008                   |
| FUDGE            | 0.470 $\pm$ 0.011                   | 0.112 $\pm$ 0.008                   | 0.682 $\pm$ 0.007                   |
| <b>SocraGate</b> | <b>0.490 <math>\pm</math> 0.010</b> | <b>0.053 <math>\pm</math> 0.006</b> | <b>0.717 <math>\pm</math> 0.007</b> |

$\uparrow$ : higher is better;  $\downarrow$ : lower is better. Bold numerical entries indicate the best mean for each metric within each corpus. Bold labels distinguish corpus headings, column headings, and SocraGate.

#### 4.5. Blind Human Evaluation

The blind evaluation sampled 160 MathDial test dialogues, with 20 dialogues from each method. Six graduate raters had at least six months of mathematics-tutoring experience. Each rater scored every dialogue for guidance naturalness, response fit, pacing, and non-leakage credibility. Method identities were masked, and dialogue order was randomized independently for each rater.

A 20-dialogue calibration preceded formal scoring. Mean pairwise Cohen's  $\kappa$  was 0.71 during calibration, with a range of 0.62–0.79. Mean pairwise  $\kappa$  after calibration was 0.74, with a range of 0.66–0.81. Five-point anchors defined poor, intermediate, and strong performance for each dimension. The anchors distinguished generic from context-specific guidance, generic from specific response fit, and implausible from attainable pacing. For non-leakage, they distinguished explicit answer disclosure from preservation of essential learner reasoning.

Table 4 and Figure 3c report method means and 1000-resample dialogue-bootstrap standard errors. SocraGate has the highest mean on all four dimensions. Compared with Self-revise, it gains 0.24 in naturalness, 0.19 in response fit, 0.21 in pacing, and 0.67 in non-leakage. These differences describe the 20 sampled dialogues per method; the dialogue-bootstrap standard errors summarize uncertainty in each method mean, not in a paired contrast between methods.

**Table 4.** Blind human evaluation. Values are mean  $\pm$  dialogue-bootstrap standard error over 20 dialogues per method. All six raters score every dialogue on a 1–5 scale.

| Method           | Naturalness $\uparrow$            | Response Fit $\uparrow$           | Pacing $\uparrow$                 | Non-Leakage $\uparrow$            |
|------------------|-----------------------------------|-----------------------------------|-----------------------------------|-----------------------------------|
| Markel prompt    | 3.41 $\pm$ 0.08                   | 3.30 $\pm$ 0.09                   | 3.25 $\pm$ 0.08                   | 2.98 $\pm$ 0.10                   |
| Fixed-sim. DPO   | 3.68 $\pm$ 0.07                   | 3.55 $\pm$ 0.08                   | 3.42 $\pm$ 0.07                   | 3.20 $\pm$ 0.09                   |
| Fixed-sim. GRPO  | 3.84 $\pm$ 0.06                   | 3.71 $\pm$ 0.07                   | 3.58 $\pm$ 0.07                   | 3.42 $\pm$ 0.08                   |
| NeuroLogic       | 3.79 $\pm$ 0.07                   | 3.66 $\pm$ 0.07                   | 3.55 $\pm$ 0.07                   | 3.88 $\pm$ 0.07                   |
| FUDGE            | 3.86 $\pm$ 0.06                   | 3.74 $\pm$ 0.07                   | 3.61 $\pm$ 0.06                   | 4.02 $\pm$ 0.06                   |
| GeDi             | 3.82 $\pm$ 0.06                   | 3.70 $\pm$ 0.07                   | 3.57 $\pm$ 0.07                   | 3.95 $\pm$ 0.07                   |
| Self-revise      | 3.88 $\pm$ 0.06                   | 3.77 $\pm$ 0.06                   | 3.63 $\pm$ 0.06                   | 4.11 $\pm$ 0.06                   |
| <b>SocraGate</b> | <b>4.12 <math>\pm</math> 0.05</b> | <b>3.96 <math>\pm</math> 0.06</b> | <b>3.84 <math>\pm</math> 0.06</b> | <b>4.78 <math>\pm</math> 0.05</b> |

$\uparrow$ : higher is better. Bold numerical entries indicate the highest mean rating in each column. The SocraGate label is bold to identify the proposed method.

#### 4.6. Component Analysis

Before admission, the scaffold-spectrum policy reaches simulator-based Success@10 of 0.501, compared with 0.482 for fixed-simulator GRPO, while both have TR of 0.168 (Tables 2 and 5). The cumulative analysis then holds the scaffold-spectrum policy fixed and adds admission stages (Figure 4a). Deterministic rules produce the largest single-stage reduction in TR, from 0.168 to 0.071, while simulator-based Success@10 increases from 0.501 to 0.518. Adding the semantic classifier further reduces TR to 0.039, with Success@10 reaching 0.521. Adding the regeneration further reduces TR to 0.032, with Success@10 reaching 0.546.

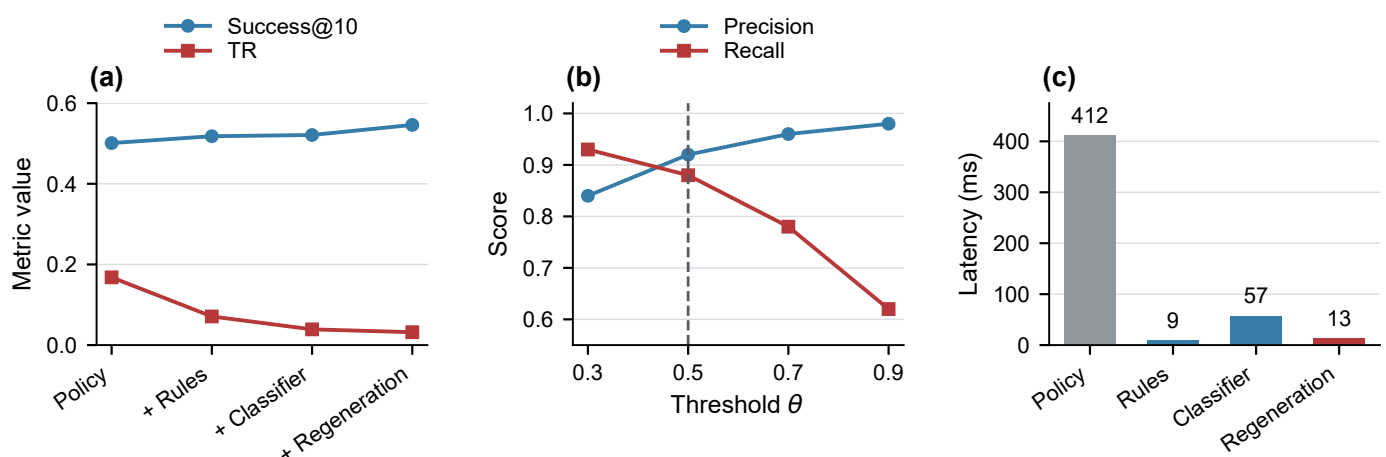


**Figure 4.** Component, classifier, and latency diagnostics. (a) cumulative changes in simulator-based Success@10 and TR. (b) telling-class precision and recall across classifier thresholds; the dashed line marks the fixed threshold 0.5. (c) mean per-turn latency at  $K = 8$ . Threshold results are post hoc test-set diagnostics, not model-selection evidence.

Guarded regeneration produces the largest incremental gain in simulator-based Success@10, from 0.521 to 0.546, while pedagogy increases from 0.782 to 0.798. TR decreases further from 0.039 to 0.032. The rows form a cumulative sequence and are interpreted as pipeline increments rather than factorial component effects.

**Table 5.** Incremental component analysis on MathDial. Rows cumulatively add inference components to the scaffold-spectrum policy. Values are means over five seeds.

| Configuration            | Sim. Success@10 ↑ | TR ↓         | Pedagogy ↑   |
|--------------------------|-------------------|--------------|--------------|
| Scaffold-spectrum policy | 0.501             | 0.168        | 0.732        |
| + Rule filter            | 0.518             | 0.071        | 0.758        |
| + Semantic classifier    | 0.521             | 0.039        | 0.782        |
| + Guarded regeneration   | <b>0.546</b>      | <b>0.032</b> | <b>0.798</b> |

↑: higher is better; ↓: lower is better. Bold entries indicate the best mean in each column among the cumulative configurations.

#### 4.7. Classifier Sensitivity, Residual Leakage, and Latency

**Classifier sensitivity.** Table 6 and Figure 4b report post hoc test-set sensitivity of the semantic leakage classifier. The operational threshold was fixed at  $\theta = 0.5$  before test evaluation; the remaining thresholds characterize sensitivity. At  $\theta = 0.5$ , telling-class precision is 0.92, recall is 0.88, and F1 is 0.90. Lower thresholds increase recall, whereas higher thresholds increase precision. This diagnostic varies the classifier threshold; reward weights and sampling budgets remain fixed.

**Table 6.** Post hoc classifier sensitivity on the MathDial test set. Precision, recall, and F1 measure detection of the *telling\_answer* class. The fixed operational threshold is bold.

| Threshold  | Precision ↑ | Recall ↑    | F1 ↑        |
|------------|-------------|-------------|-------------|
| 0.3        | 0.84        | 0.93        | 0.88        |
| <b>0.5</b> | <b>0.92</b> | <b>0.88</b> | <b>0.90</b> |
| 0.7        | 0.96        | 0.78        | 0.86        |
| 0.9        | 0.98        | 0.62        | 0.76        |

↑: higher is better. The bold row identifies the fixed operational threshold  $\theta = 0.5$  and its associated results; bold does not indicate the best value in every column.

**Residual leakage.** Two annotators manually categorized 47 residual emitted leaks in the MathDial audit. Implicit semantic paraphrases accounted for 24 cases, or 51.1%. Intermediate values equivalent to the answer accounted for 15 cases. The remaining eight cases used sentence forms not covered by the deterministic rules. Table 7 reports the composition of these 47 identified leaks; the percentages do not estimate leakage frequency across all emitted responses.

**Table 7.** Residual leakage taxonomy. Counts sum to the 47 emitted responses identified as answer revealing in the MathDial audit.

| Failure Mode                            | Count | Share  |
|-----------------------------------------|-------|--------|
| Implicit semantic paraphrase            | 24    | 51.1%  |
| Intermediate value equivalent to answer | 15    | 31.9%  |
| Rule-uncovered sentence form            | 8     | 17.0%  |
| Total                                   | 47    | 100.0% |

**Latency.** Figure 4c reports mean per-turn latency at  $K = 8$ . Policy sampling contributes 412 ms, deterministic rules add 9 ms, the classifier adds 57 ms, and guarded regeneration adds 13 ms averaged across all tutor turns. Total mean latency is 491 ms per tutor turn, 19.2% above policy sampling alone. The measured mean does not characterize the worst-case retry path, which can evaluate up to  $K + K_{\max}K' = 20$  candidates per turn.

## 5. Discussion and Limitations

The component results support treating candidate quality and response admission as complementary design concerns in LLM tutoring. The policy favors scaffolded guidance, while admission checks each proposed response for disclosure before delivery.

The present evidence concerns mathematics tasks with available reference solutions, assessed through a fixed learner simulator and a small blind-rating study. These evaluations characterize tutoring behavior rather than real-student learning gains. Extension to other domains will require suitable reference information and verification rules, with learning outcomes, admission coverage, and inference cost evaluated in the intended setting.

## 6. Conclusions

SocraGate formulates answer-leakage control in scaffolded LLM tutoring as an admission decision over complete responses, combining scaffold-oriented policy optimization with deterministic and semantic screening and guarded regeneration. Across five MathDial seeds, SocraGate reduces utterance-level telling rate from 0.168 to 0.032 relative to fixed-simulator GRPO, while simulator-based Success@10 rises from 0.482 to 0.546. The same ordering appears on QATD-2k and CoMTA without parameter updates under the shared simulator-based protocol. SocraGate also receives the highest mean rating on all four dimensions in the blind evaluation. Cumulative analysis identifies deterministic screening as the largest single-stage reduction in leakage and guarded regeneration as the largest incremental gain in simulator-based success. Within these evaluated mathematical tasks, the results support treating answer disclosure as an explicit constraint on response delivery alongside pedagogical objectives.

**Author Contributions:** Conceptualization, W.L. and K.Z.; methodology, W.L.; software, W.L.; investigation, W.L.; data curation, W.L.; writing—original draft preparation, W.L.; writing—review and editing, K.Z.; visualization, W.L.; supervision, K.Z.; project administration, K.Z. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Institutional Review Board Statement:** Ethical review and approval were waived for this study because the research evaluated an AI tutoring system using existing tutoring datasets and simulated learners, with human involvement limited to rating and annotating model-generated tutoring responses. No instructional intervention was conducted with real learners. The exemption was reviewed and issued by the School of Media, Qufu Normal University, as documented in the institutional materials provided to the journal.

**Informed Consent Statement:** Written informed consent was obtained from the individuals involved in the human evaluation.

**Data Availability Statement:** The MathDial, QATD-2k, and CoMTA datasets are available from their original sources [7,28,29], subject to the respective access and licensing terms. The dialogues generated in this study and the human evaluation records are available from the corresponding author upon reasonable request.

**Conflicts of Interest:** The authors declare no conflicts of interest.

## References

1. Vygotsky, L.S.; Cole, M. *Mind in Society: Development of Higher Psychological Processes*; Harvard University Press: Cambridge, MA, USA, 1978.
2. Bruner, J. *The Process of Education: Revised Edition*; Harvard University Press: Cambridge, MA, USA, 2009.
3. Wood, D.; Bruner, J.S.; Ross, G. The role of tutoring in problem solving. *J. Child Psychol. Psychiatry* **1976**, *17*, 89–100. [[CrossRef](#)] [[PubMed](#)]

4. Koedinger, K.R.; Aleven, V. Exploring the assistance dilemma in experiments with cognitive tutors. *Educ. Psychol. Rev.* **2007**, *19*, 239–264. [\[CrossRef\]](#)
5. VanLehn, K. The relative effectiveness of human tutoring, intelligent tutoring systems, and other tutoring systems. *Educ. Psychol.* **2011**, *46*, 197–221. [\[CrossRef\]](#)
6. Kulik, J.A.; Fletcher, J.D. Effectiveness of intelligent tutoring systems: A meta-analytic review. *Rev. Educ. Res.* **2016**, *86*, 42–78.
7. Macina, J.; Daheim, N.; Chowdhury, S.; Sinha, T.; Kapur, M.; Gurevych, I.; Sachan, M. MathDial: A dialogue tutoring dataset with rich pedagogical properties grounded in math reasoning problems. In *Findings of the Association for Computational Linguistics: EMNLP 2023*; Association for Computational Linguistics: Singapore, 2023; pp. 5602–5621.
8. Sonkar, S.; Ni, K.; Chaudhary, S.; Baraniuk, R. Pedagogical alignment of large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2024*; Association for Computational Linguistics: Miami, FL, USA, 2024; pp. 13641–13650.
9. Puech, R.; Macina, J.; Chatain, J.; Sachan, M.; Kapur, M. Towards the pedagogical steering of large language models for tutoring: A case study with modeling productive failure. In *Findings of the Association for Computational Linguistics: ACL 2025*; Association for Computational Linguistics: Vienna, Austria, 2025; pp. 26291–26311.
10. Dinucu-Jianu, D.; Macina, J.; Daheim, N.; Hakimi, I.; Gurevych, I.; Sachan, M. From problem-solving to teaching problem-solving: Aligning LLMs with pedagogy using reinforcement learning. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, Suzhou, China, 4–9 November 2025; pp. 272–292.
11. Zhao, J.; Knežević, M.; Käser, T. Evaluating Answer Leakage Robustness of LLM Tutors against Adversarial Student Attacks. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, San Diego, CA, USA, 2–7 July 2026; pp. 30588–30617.
12. Gangrade, A.; Kag, A.; Saligrama, V. Selective classification via one-sided prediction. In *Proceedings of the International Conference on Artificial Intelligence and Statistics*, PMLR, Virtual, 13–15 April 2021; pp. 2179–2187.
13. Tack, A.; Piech, C. The AI teacher test: Measuring the pedagogical ability of blender and GPT-3 in educational dialogues. *arXiv* **2022**, arXiv:2205.07540.
14. Tack, A.; Kochmar, E.; Yuan, Z.; Bibauw, S.; Piech, C. The BEA 2023 shared task on generating AI teacher responses in educational dialogues. In *Proceedings of the 18th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2023)*, Toronto, ON, Canada, 13 July 2023; pp. 785–795.
15. Scarlatos, A.; Liu, N.; Lee, J.; Baraniuk, R.; Lan, A. Training LLM-based tutors to improve student learning outcomes in dialogues. In *Proceedings of the 26th International Conference, AIED 2025, Palermo, Italy, 22–26 July 2025*; Springer: Cham, Switzerland, 2025; pp. 251–266.
16. Rafailov, R.; Sharma, A.; Mitchell, E.; Manning, C.D.; Ermon, S.; Finn, C. Direct preference optimization: Your language model is secretly a reward model. *Adv. Neural Inf. Process. Syst.* **2023**, *36*, 53728–53741. [\[CrossRef\]](#)
17. Maurya, K.K.; Srivatsa, K.A.; Petukhova, K.; Kochmar, E. Unifying AI tutor evaluation: An evaluation taxonomy for pedagogical ability assessment of LLM-powered AI tutors. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, Albuquerque, NM, USA, 29 April–4 May 2025; pp. 1234–1251.
18. Kochmar, E.; Maurya, K.K.; Petukhova, K.; Srivatsa, K.A.; Tack, A.; Vasselli, J. Findings of the BEA 2025 shared task on pedagogical ability assessment of AI-powered tutors. In *Proceedings of the 20th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2025)*, Vienna, Austria, 31 July–1 August 2025; pp. 1011–1033.
19. Macina, J.; Daheim, N.; Hakimi, I.; Kapur, M.; Gurevych, I.; Sachan, M. MathTutorBench: A benchmark for measuring open-ended pedagogical capabilities of LLM tutors. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, Suzhou, China, 4–9 November 2025; pp. 204–221.
20. Shi, Y.; Liang, R.; Xu, Y. EducationQ: Evaluating LLMs’ teaching capabilities through multi-agent dialogue framework. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Vienna, Austria, 27 July–1 August 2025; pp. 32799–32828.
21. van de Pol, J.; Volman, M.; Beishuizen, J. Scaffolding in Teacher–Student Interaction: A Decade of Research. *Educ. Psychol. Rev.* **2010**, *22*, 271–296. [\[CrossRef\]](#)
22. Ryan, R.M.; Deci, E.L. Self-Determination Theory and the Facilitation of Intrinsic Motivation, Social Development, and Well-Being. *Am. Psychol.* **2000**, *55*, 68–78. [\[CrossRef\]](#) [\[PubMed\]](#)
23. Corbett, A.T.; Anderson, J.R. Knowledge tracing: Modeling the acquisition of procedural knowledge. *User Model. User Adapt. Interact.* **1994**, *4*, 253–278. [\[CrossRef\]](#)
24. Scarlatos, A.; Baker, R.S.; Lan, A. Exploring knowledge tracing in tutor-student dialogues using LLMs. In *Proceedings of the 15th International Learning Analytics and Knowledge Conference*, Dublin, Ireland, 3–7 March 2025; pp. 249–259.
25. Markel, J.M.; Opferman, S.G.; Landay, J.A.; Piech, C. GPTeach: Interactive TA training with GPT-based students. In *Proceedings of the Tenth ACM Conference on Learning @ Scale*, Copenhagen, Denmark, 20–22 July 2023; pp. 226–236.

26. Liu, Z.; Yin, S.X.; Lin, G.; Chen, N. Personality-aware student simulation for conversational intelligent tutoring systems. In Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, Miami, FL, USA, 12–16 November 2024; pp. 626–642.
27. Scarlatos, A.; Lee, J.; Woodhead, S.; Lan, A. Simulated Students in Tutoring Dialogues: Substance or Illusion? In Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), San Diego, CA, USA, 2–7 July 2026; pp. 42349–42385.
28. Zent, M.; Smith, D.; Woodhead, S. Piivot: A lightweight NLP anonymization framework for question-anchored tutoring dialogues. In Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, Suzhou, China, 4–9 November 2025; pp. 27467–27476.
29. Miller, P.; DiCerbo, K. LLM based math tutoring: Challenges and dataset. *OSF Preprint* **2024**. [[CrossRef](#)]
30. Lu, X.; West, P.; Zellers, R.; Le Bras, R.; Bhagavatula, C.; Choi, Y. NeuroLogic decoding: (Un)supervised neural text generation with predicate logic constraints. In Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Online, 6–11 June 2021; pp. 4288–4299.
31. Yang, K.; Klein, D. FUDGE: Controlled text generation with future discriminators. In Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Online, 6–11 June 2021; pp. 3511–3535.
32. Liu, A.; Sap, M.; Lu, X.; Swayamdipta, S.; Bhagavatula, C.; Smith, N.A.; Choi, Y. DExperts: Decoding-time controlled text generation with experts and anti-experts. In Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), Online, 1–6 August 2021; pp. 6691–6706. [[CrossRef](#)]
33. Krause, B.; Gotmare, A.D.; McCann, B.; Keskar, N.S.; Joty, S.; Socher, R.; Rajani, N.F. GeDi: Generative discriminator guided sequence generation. In Proceedings of the Findings of the Association for Computational Linguistics: EMNLP 2021, Punta Cana, Dominican Republic, 7–11 November 2021; pp. 4929–4952.
34. Bai, Y.; Kadavath, S.; Kundu, S.; Askell, A.; Kernion, J.; Jones, A.; Chen, A.; Goldie, A.; Mirhoseini, A.; McKinnon, C.; et al. Constitutional AI: Harmlessness from AI feedback. *arXiv* **2022**, arXiv:2212.08073.
35. Glaese, A.; McAleese, N.; Trębacz, M.; Aslanides, J.; Firoiu, V.; Ewalds, T.; Rauh, M.; Weidinger, L.; Chadwick, M.; Thacker, P.; et al. Improving alignment of dialogue agents via targeted human judgements. *arXiv* **2022**, arXiv:2209.14375.
36. Madaan, A.; Tandon, N.; Gupta, P.; Hallinan, S.; Gao, L.; Wiegrefe, S.; Alon, U.; Dziri, N.; Prabhunoye, S.; Yang, Y.; et al. Self-Refine: Iterative refinement with self-feedback. *Adv. Neural Inf. Process. Syst.* **2023**, *36*, 46534–46594. [[CrossRef](#)]
37. Achiam, J.; Held, D.; Tamar, A.; Abbeel, P. Constrained policy optimization. In Proceedings of the International Conference on Machine Learning, PMLR, Sydney, Australia, 6–11 August 2017; pp. 22–31.
38. Geifman, Y.; El-Yaniv, R. Selective classification for deep neural networks. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 4878–4887.
39. Geifman, Y.; El-Yaniv, R. SelectiveNet: A deep neural network with an integrated reject option. In Proceedings of the International Conference on Machine Learning, PMLR, Long Beach, CA, USA, 9–15 June 2019; pp. 2151–2159.
40. Chen, J.; Yoon, J.; Ebrahimi, S.; Arik, S.O.; Pfister, T.; Jha, S. Adaptation with self-evaluation to improve selective prediction in LLMs. In Proceedings of the Findings of the Association for Computational Linguistics: EMNLP 2023, Singapore, 6–10 December 2023; pp. 5190–5213.
41. Angelopoulos, A.; Bates, S.; Fisch, A.; Lei, L.; Schuster, T. Conformal risk control. In Proceedings of the International Conference on Learning Representations, Vienna, Austria, 7–11 May 2024; Volume 2024, pp. 55198–55218.
42. Pal Chowdhury, S.; Zouhar, V.; Sachan, M. AutoTutor meets large language models: A language model tutor with rich pedagogy and guardrails. In Proceedings of the Eleventh ACM Conference on Learning @ Scale, Atlanta, GA, USA, 18–20 July 2024; pp. 5–15.
43. Kadir, N. Auditable Release Control for Pedagogical Leakage in LLM Tutors. *arXiv* **2026**, arXiv:2608.00515.
44. Pisan, Y. Teaching a Large Language Model Tutor to Withhold the Answer: A Supervisor Architecture and an Evidence-Driven Method for Tuning Socratic Behavior. *arXiv* **2026**, arXiv:2608.12292.
45. Wang, J.; Dai, Y.; Zhang, Y.; Ma, Z.; Li, W.; Chai, J. Training Turn-by-Turn Verifiers for Dialogue Tutoring Agents: The Curious Case of LLMs as Your Coding Tutors. In Proceedings of the Findings of the Association for Computational Linguistics: ACL 2025, Vienna, Austria, 27 July–1 August 2025; pp. 12416–12436. [[CrossRef](#)]
46. NLP-UnMask Contributors. UnMask: Socratic AI Tutor with Progressive Context Revelation. Software Repository. Available online: <https://github.com/Sanika0212/NLP-UnMask> (accessed on 12 October 2026).
47. Wang, X.; Song, S.; Liu, W.; Zou, X. Beyond Direct Answering: Aligning Educational LLMs as Socratic Guides via Heuristic Reinforcement Learning. *arXiv* **2026**, arXiv:2607.22996. [[CrossRef](#)]
48. Shao, Z.; Wang, P.; Zhu, Q.; Xu, R.; Song, J.; Bi, X.; Zhang, H.; Zhang, M.; Li, Y.; Wu, Y.; et al. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models. *arXiv* **2024**, arXiv:2402.03300.

49. Qwen Team. Qwen2.5 Technical Report. *arXiv* **2025**, arXiv:2412.15115. [[CrossRef](#)]
50. Zhang, T.; Kishore, V.; Wu, F.; Weinberger, K.Q.; Artzi, Y. BERTScore: Evaluating text generation with BERT. *arXiv* **2019**, arXiv:1904.09675.

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.