SD-E^2: Semantic Exploration for Reasoning Under Token Budgets

Kshitij Mishra **Nils Lukas** **Salem Lahlou**
Mohamed bin Zayed University of Artificial Intelligence
{kshitij.mishra,nils.lukas,salem.lahlou}@mbzuai.ac.ae

Abstract

Small language models (SLMs) struggle with complex reasoning because exploration is expensive under tight compute budgets. We introduce Semantic Diversity – Exploration–Exploitation (**SD-E^2**), a reinforcement learning framework that makes exploration explicit by optimizing *semantic* diversity in generated reasoning trajectories. Using a frozen sentence-embedding model, SD-E^2 assigns a diversity reward that captures (i) the coverage of semantically distinct solution strategies and (ii) their average pairwise dissimilarity in embedding space, rather than surface-form novelty. This diversity reward is combined with outcome correctness and solution efficiency in a z-score–normalized multi-objective objective that stabilizes training. On GSM8K, SD-E^2 surpasses the base `Qwen2.5-3B-Instruct` and strong GRPO baselines (GRPO-CFL and GRPO-CFEE) by +27.4, +5.3, and +1.5 percentage points, respectively, while discovering on average 9.8 semantically distinct strategies per question. We further improve MedMCQA to 49.64% vs 38.37 for base and show gains on the harder AIME benchmark (1983–2025), reaching 13.28% vs. base 6.74%. These results indicate that rewarding semantic novelty yields a more compute-efficient exploration–exploitation signal for training reasoning-capable SLMs. By introducing cognitive adaptation (adjusting the reasoning process structure rather than per-token computation), SD-E^2 offers a complementary path to efficiency gains in resource-constrained models.

1 Introduction

Large Language Models (LLMs) have demonstrated remarkable reasoning ability across mathematics, science, and general-domain tasks (Wei et al., 2022; Kojima et al., 2022; Bubeck et al., 2023; Shinn et al., 2023; Zelikman et al., 2023). Techniques such as Chain-of-Thought prompting (Wei et al., 2022) and Tree-of-Thoughts search (Shinn et al., 2023) enable these models to generate multi-step reasoning traces and explore alternative strategies. However, their immense scale—often tens or hundreds of billions of parameters—comes with high inference cost and latency, motivating a shift toward **Small Language Models (SLMs)** for cost-efficient and deployable reasoning (Chen and et al., 2023; Microsoft Research Team, 2024). Yet, SLMs struggle to match the reasoning fidelity of their larger counterparts. Their limited capacity increases susceptibility to exposure bias (Ranzato et al., 2015), while their tight token budgets constrain the complexity and length of reasoning paths.

This limitation introduces a fundamental tension between *exploration* and *exploitation*. An SLM must explore diverse reasoning strategies to escape local optima and discover valid solution paths, yet it must quickly exploit promising avenues to stay within its computational and token budget. Existing methods inadequately resolve this trade-off. Inference-time ensembling techniques such as Self-Consistency (Wang et al., 2023b), Tree-of-Thoughts (Shinn et al., 2023), and Reasoning-as-Planning (Zhou et al., 2023) improve accuracy but incur significant overhead, negating the efficiency gains of smaller models. Meanwhile, Reinforcement Learning (RL) alignment methods such as RLHF (Christiano et al., 2017; Bai et al., 2022; Ouyang et al., 2022) and preference-optimization variants like DPO (Rafailov et al., 2023), IPO (Aznarez et al., 2023), and GRPO (Shao et al., 2024) rely primarily on sparse outcome-based signals (e.g., correctness or preference). Recent works on process supervision (Lightman et al., 2023; Wu et al., 2023; Wang et al., 2023a; Huang et al., 2024; Zhou et al., 2025) take a step further by rewarding intermediate reasoning steps, yet they still lack a measure of *exploration quality*. As a result, current methods cannot distinguish between

Findings of the Association for Computational Linguistics: EACL 2026, pages 6144–6157

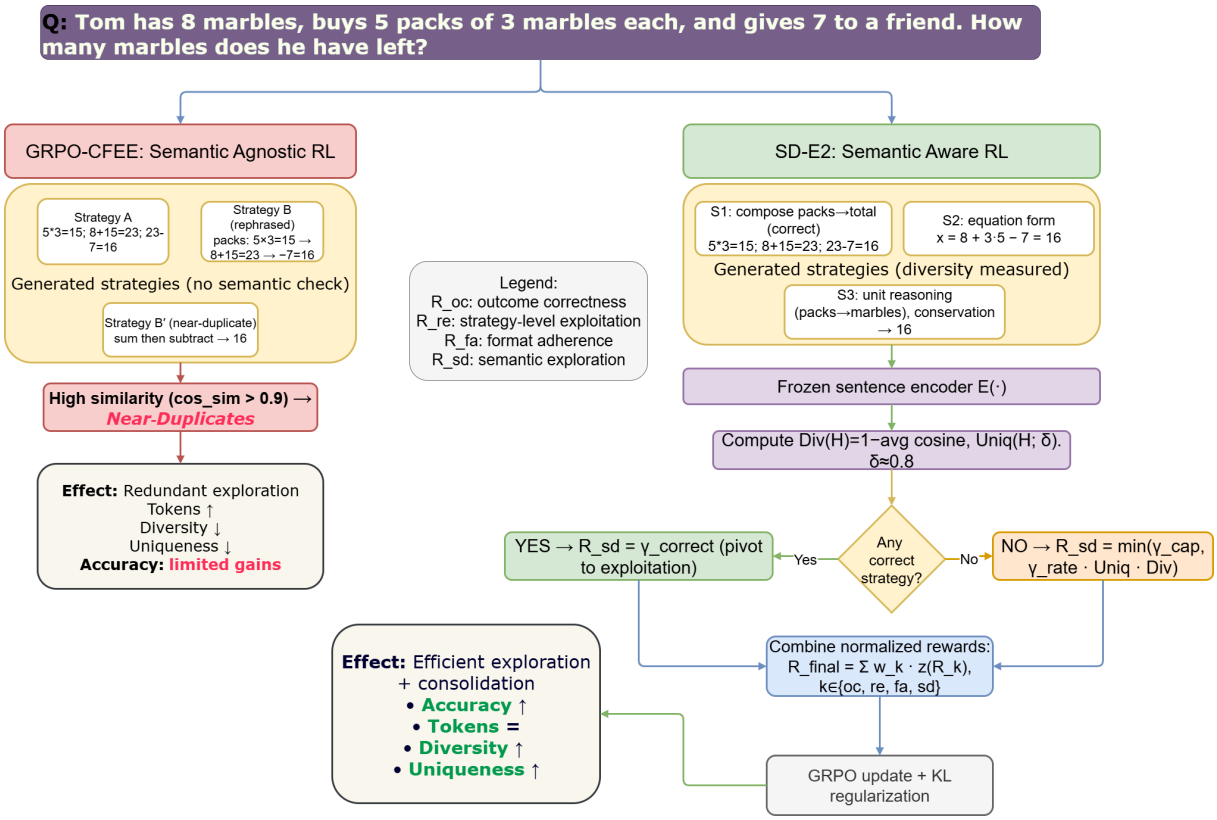

Figure 1: **Problem and approach overview on a GSM8K example.** *Left:* Outcome-driven baselines (e.g., GRPO-CFL) and the non-semantic GRPO-CFEE can generate multiple, near-duplicate strategies, leading to redundant exploration (Tokens ↑, Diversity ↓). *Right:* SD-E^2 encodes each `<reasoning>` with a frozen sentence encoder to compute (i) $\mathrm{Div}(H) = 1 - \text{avg cosine}$ and (ii) $\mathrm{Uniq}(H; \delta)$, rewarding exploration only when strategies are *semantically* distinct; upon any correct strategy, it switches to an exploitation bonus. Normalized components $R_{\mathrm{oc}}, R_{\mathrm{re}}, R_{\mathrm{fa}}, R_{\mathrm{sd}}$ are combined under a GRPO objective with KL regularization, yielding higher ACC with fewer tokens.

discovering a genuinely novel reasoning strategy and merely rephrasing an existing one, leading to repetitive and inefficient search behavior.

In this work, we introduce **SD-E^2**, a semantics-aware reinforcement learning framework that teaches SLMs to reason efficiently by rewarding exploration only when it is *meaningfully different*. At its core is a **semantic exploration reward** that leverages a frozen sentence-embedding model to measure the diversity of reasoning traces. When a correct solution is found, SD-E^2 shifts focus to exploitation through a fixed reward bonus, encouraging consolidation of success. When no correct strategy is discovered, the exploration reward scales with both the number of *semantically unique* reasoning paths and their average dissimilarity, promoting broad yet targeted exploration of novel ideas rather than superficial rewording. This represents a form of *cognitive adaptation* (Graves, 2016): rather than adapting the per-token computational cost through architectural means (e.g., early exiting, sparse experts), SD-E^2 adapts the high-level reasoning process itself based on semantic saturation, preventing the generation of entire redundant strategy blocks.

As summarized in Fig. 1, baselines without a semantic signal often produce near-duplicate strategies (high cosine similarity), while SD-E^2 measures semantic diversity and rewards only *meaningfully different* exploration, pivoting to exploitation once any strategy yields the correct outcome.

On GSM8K, we first compare against the base model: with identical prompts on `Qwen2.5-3B-Instruct`, SD-E^2 improves accuracy by +26.0 points. We then benchmark against strong GRPO baselines (outcome-driven GRPO (GRPO-CFL; DeepSeek-AI, 2025)) and multi-objective without semantic awareness (GRPO-CFEE)), and obtain additional gains of +6.0 and +1.5 points, respectively, under the same prompt/token budget, while discovering on average 9.8 semantically distinct strategies per problem. Taken together, these results indicate that rewarding semantic novelty yields a more compute-efficient exploration–exploitation signal for training reasoning-capable SLMs.

Our contributions are threefold.

- We introduce **cognitive adaptation** for reasoning: rather than adapting per-token computation architecturally, we adapt the high-level reasoning process itself by measuring semantic saturation and preventing generation of entire redundant strategy blocks.
- We propose a **semantic diversity reward** that quantifies exploration quality via embedding geometry, rewarding semantically distinct reasoning paths during search and pivoting to exploitation once success is achieved, addressing the fundamental limitation that existing RL methods cannot distinguish genuine strategic novelty from surface-form rephrasing.
- We demonstrate that **semantic novelty improves exploration-efficiency** across math and medical reasoning: on GSM8K (Qwen2.5-3B-Instruct), SD-E^2 improves ACC from 54.66% to **82.03**% (and by +5.23/+1.51 points over GRPO-CFL/GRPO-CFEE), while increasing strategy-level success (S-ACC) and discovering on average 9.78 strategies per problem. We further validate on the harder AIME benchmark (1983–2025), where SD-E^2 improves accuracy to 13.28% vs. 6.74% for base under comparable decoding budgets.

2 Related Work

Reasoning in LLMs. CoT prompting and its extensions (e.g. self-consistency, tree-of-thought, graph-of-thought; Wei et al., 2022; Wang et al., 2023b; Shinn et al., 2023; Zhang et al., 2023) guide models to generate multi-step solutions, improving performance on complex reasoning tasks, though at the cost of verbosity and sampling overhead. These methods apply scaffolding at inference time but do not adaptively decide when to stop exploring strategies. To address that, Lightman et al. (2023); Wu et al. (2023) propose *process-level supervision* by giving feedback at intermediate reasoning steps, showing that stepwise feedback significantly outperforms outcome-only supervision in solving difficult math problems.

Structured reasoning alternatives. Parallel to prompt-based methods, neuro-symbolic approaches improve reliability by grounding reasoning in verifiable formalisms. Program-Aided Language models (PAL) (Zheng et al., 2022) separate natural language understanding from calculation by generating executable code and offloading computation to interpreters, achieving strong performance on arithmetic tasks. Other work integrates Knowledge Graphs (Jiang et al., 2023) or decomposes questions into reasoning graphs (Ko et al., 2024). While these methods improve factuality, they operate in different paradigms. Our work focuses on improving free-text generative reasoning from within.

RL for language and structured reasoning. Reinforcement learning methods have advanced from outcome optimization (e.g. RLHF, RLAIF) toward more structured control of reasoning processes (Ouyang et al., 2022; Bai et al., 2022; Lee et al., 2023). Recent works on process supervision (Lightman et al., 2023; Wu et al., 2023; Wang et al., 2023a) provide fine-grained feedback on intermediate reasoning steps, significantly outperforming outcome-only methods (Uesato et al., 2022). Hybrid approaches like SuperRL (Liu et al., 2025b) adaptively combine RL with supervised fine-tuning for improved stability when reward signals are sparse. Group-based policy optimization (GRPO) methods (Shao et al., 2024) sample multiple candidate outputs per input and assign relative rewards, thereby avoiding the need for a learned value network. Works such as GLoRe (Havrilla et al., 2024), use learned reward models to decide when to rewrite or refine parts of generated reasoning paths (global or local repair), further improving solution quality. Recent work also applies preference optimization directly to reasoning traces (Lai et al., 2024; Lahlou et al., 2025), learning from trajectory-level comparisons. However, process-supervised RL creates a second-order challenge: how to manage exploration efficiently.

Current frameworks often encourage exploration through uniform sampling or heuristics like token entropy, but these approaches are semantically blind: they may reward trivial lexical variations of the same core reasoning strategy, as shown in Liu et al. (2025a), which proposes branching from high-attention positions as an exploration heuristic. However, this remains based on internal model mechanics rather than the semantic content of generated strategies. Our method addresses this gap by introducing a **semantic diversity gate** that measures marginal novelty and curtails exploration once it becomes redundant, instead of relying on fixed heuristics or predetermined stopping rules.

Semantic diversity, subset selection, and novelty in generation. Not all diversity is equally valu-

able for reasoning. Standard decoding methods like beam search produce near-identical outputs differing only in minor word choices (Vijayakumar et al., 2018)—lexical variation that provides poor candidate pools for Best-of-N sampling or RL (Shi et al., 2025). Methods for promoting meaningful diversity span a spectrum. Diverse Beam Search (Vijayakumar et al., 2018) uses n-gram dissimilarity penalties but remains lexically focused. More sophisticated approaches like Semantic-guided Diverse Decoding (SemDiD) (Shi et al., 2025) operate in embedding space with orthogonal directional guidance, ensuring candidates occupy distinct semantic regions, though only at inference time.

Our approach embeds diversity into the training objective, inspired by diverse subset selection. Maximal Marginal Relevance (MMR; Carbonell and Goldstein, 1998) balances relevance vs. novelty to reduce redundancy in retrieval results. Submodular coverage functions are widely used to model diminishing returns in summarization and content selection, with greedy maximization yielding good approximation guarantees (Lin and Bilmes, 2011). Determinantal point processes (DPP; Kulesza et al., 2012) also support sampling of diverse subsets by discouraging similarity, with the log-determinant capturing both quality and diversity (Gong et al., 2014). Recent work applies DPP-based objectives to jointly train LLMs for quality and diversity (Chen et al., 2025). In prior reasoning work, diversity is often encouraged via sampling, variance-based bonuses, or temperature tuning, or more recently with GFlowNet-based fine-tuning for diverse and accurate mathematical reasoning (Younsi et al., 2025), but not with an explicit measure of semantic coverage across generated reasoning paths.

SD-E^2 leverages the same mathematical principles (our coverage objective is monotone submodular) but deploys it dynamically during trajectory generation as a gate, transforming diversity from a post-hoc reward into real-time process control. This combination allows the model to explore meaningfully distinct strategies up to a saturation point, and then exploit the most promising one under a token budget.

Adaptive computation. Our work also connects to adaptive computation, where systems adjust computational budget based on input complexity (Graves, 2016). The dominant paradigm is architectural adaptation: early exiting (Schuster et al., 2022; Xin et al., 2020) attaches classifiers to intermediate layers to exit on easy inputs, while Mixture of Experts (MOE; Fedus et al., 2022) routes tokens to sparse expert subnetworks. Recent work applies early exiting specifically to reasoning chains, truncating CoT when confidence is reached (Yang et al., 2025). SD-E^2 introduces a complementary form we term *cognitive adaptation*. While architectural methods adapt per-token computation, we adapt the high-level reasoning process based on semantic saturation. Our gate prevents generation of entire redundant strategy blocks rather than making individual tokens cheaper, which is an orthogonal approach that could combine with architectural methods for compounded efficiency gains.

3 Method

SD-E^2 trains an SLM with a multi–objective reward that (i) checks the final answer and intermediate strategy outcomes, (ii) enforces a lightweight output format, and (iii) explicitly rewards *semantic* exploration using sentence-embedding geometry. Rewards are z-score normalized per batch and optimized with GRPO plus a KL term. SD-E^2 introduces *cognitive adaptation*: adapting the high-level structure and content of the reasoning process based on semantic metrics rather than computational heuristics. By measuring semantic novelty with a frozen encoder, we create a dynamic control mechanism that stops exploration when strategies become redundant, regardless of lexical variation.

3.1 Output Format and Parsing

Let $\mathcal{Q}$ be the space of prompts and $\mathcal{Y}$ the space of gold answers, with $(q, y) \sim \mathcal{D}$. The policy π_θ is an auto-regressive distribution over tokens $a \in \Sigma^*$:

$$\pi_\theta(a \mid q) = \prod_{t=1}^{|a|} \pi_\theta\big(a_t \mid q, a_{<t}\big), \qquad a \in \Sigma^*. \quad (1)$$

We encourage a structured completion

$$a = \big[\, \langle \mathrm{STRAT} \rangle_1, \ldots, \langle \mathrm{STRAT} \rangle_m, \langle \mathrm{FA} \rangle \,\big]. \quad (2)$$

Each $\langle \mathrm{STRAT} \rangle$ block contains a reasoning section and an $\langle \mathrm{SO} \rangle$ field. For a completion a, let

$$S(a) = \{(r_i, o_i)\}_{i=1}^{m}, \quad (3)$$

$$f_{\mathrm{ans}}(a) \in \Sigma^*, \quad (4)$$

be the parsed strategies and final answer. We formalize the parsers as measurable maps

$$F_{\mathrm{strat}} : \Sigma^* \to (\Sigma^* \times \Sigma^*)^{\le M}, \quad (5)$$

$$F_{\mathrm{ans}} : \Sigma^* \to \Sigma^*, \quad (6)$$

with priorities

$$\langle \text{FA} \rangle \succ \langle \text{ANS} \rangle \succ \text{last } \langle \text{SO} \rangle. \tag{7}$$

A strategy (r, o) is *valid* if both fields are present:

$$\mathsf{valid}(r, o) = \mathbf{1}[r \neq \varnothing] \cdot \mathbf{1}[o \neq \varnothing], \tag{8}$$

$$n_{\text{strat}}(a) = \sum_{(r,o)\in S(a)} \mathsf{valid}(r, o). \tag{9}$$

3.2 Semantic Geometry of Strategies

Let $E : \Sigma^* \to \mathbb{R}^d$ be a frozen sentence encoder and define cosine similarity

$$\kappa(u, v) = \frac{\langle u, v \rangle}{\|u\|_2 \, \|v\|_2} \in [-1, 1]. \tag{10}$$

For a completion a with parsed strategies $S(a) = \{(r_i, o_i)\}_{i=1}^{m}$, collect embeddings of nonempty reasoning texts:

$$H(a) = \{\, h_i = E(r_i) \;:\; r_i \neq \varnothing \,\}, \tag{11}$$

$$m_{\text{eff}} = |H(a)| \leq m. \tag{12}$$

Diversity. For $m_{\text{eff}} \geq 2$, define the average pairwise similarity and the clamped diversity

$$\overline{\kappa}(H) = \frac{2}{m_{\text{eff}}(m_{\text{eff}} - 1)} \sum_{1 \leq i < j \leq m_{\text{eff}}} \kappa(h_i, h_j), \tag{13}$$

$$\text{Div}(H) = \big[\, 1 - \overline{\kappa}(H) \,\big]_{[0,1]}. \tag{14}$$

Set $\text{Div}(H) = 1$ if $m_{\text{eff}} = 1$ and $\text{Div}(H) = 0$ if $m_{\text{eff}} = 0$.

Unique count. Fix $\delta \in (0, 1)$. Construct $U \subseteq \{1, \ldots, m_{\text{eff}}\}$ greedily in the strategy order by including i iff

$$\max_{j \in U} \kappa(h_i, h_j) \;\leq\; \delta.$$

Define

$$\text{Uniq}(H; \delta) = |U| \in \{0, 1, \ldots, m_{\text{eff}}\}. \tag{15}$$

3.3 Reward Components

We use four bounded components $R_k(q, y, a) \in \mathbb{R}$, batch-normalized (as explained in App. A.1) and combined linearly (Eq. 30).

Outcome correctness: Checks only the final answer:

$$R_{\text{oc}}(a \mid q, y) = \lambda_{\text{oc}} \, \mathbf{1}[f_{\text{ans}}(a) = (y)\,]. \tag{16}$$

Algorithm 1 SD-E^2

Require: prompt q, gold y, completion a

$S(a) \leftarrow F_{\text{strat}}(a)$; $f_{\text{ans}}(a) \leftarrow F_{\text{ans}}(a)$

$n_{\text{strat}}(a) \leftarrow \sum_{(r,o)\in S(a)} \mathsf{valid}(r, o)$

$\mathsf{final}(a) \leftarrow \mathbf{1}[f_{\text{ans}}(a) \neq \varnothing]$; $\mathsf{complete}(a) \leftarrow \mathbf{1}[n_{\text{strat}}(a) > 0]\ \mathsf{final}(a)$

$H \leftarrow \{\, E(r) : (r, o) \in S(a),\ r \neq \varnothing \,\}$; compute $\text{Uniq}(H; \delta)$, $\text{Div}(H)$; $g(H) \leftarrow \text{Uniq} \cdot \text{Div}$

$\chi(a) \leftarrow \mathbf{1}[\,\exists (r, o) \in S(a) : N(o) = N(y)\,]$

$R_{\text{oc}} \leftarrow \lambda_{\text{oc}} \, \mathbf{1}[\,N(f_{\text{ans}}(a)) = N(y)\,]$ (Eq. 16)

$R_{\text{re}} \leftarrow \lambda_{\text{re}} \, \chi(a)$ (Eq. 19)

$R_{\text{fa}} \leftarrow \min\{1, \gamma_s \, n_{\text{strat}}(a)\} + \gamma_a \, \mathsf{final}(a) + \gamma_c \, \mathsf{complete}(a)$ (Eq. 22)

$R_{\text{sd}} \leftarrow \alpha \, \chi(a) + \big(1 - \chi(a)\big) \, \min\{\beta, \, \rho \, g(H)\}$ (Eq. 17)

return $(R_{\text{oc}}, \, R_{\text{re}}, \, R_{\text{fa}}, \, R_{\text{sd}})$

Semantic exploration: Rewards *semantic breadth* and *spread* when no correct strategy is present; collapses otherwise:

$$R_{\text{sd}}(a \mid q, y) = \alpha \, \chi(a) \tag{17}$$

$$+ \big(1 - \chi(a)\big) \, \min\big\{\beta, \, \rho \, g(H)\big\}. \tag{18}$$

where $\chi(a) = \mathbf{1}[\,\exists (r, o) \in S(a) : N(o) = N(y)\,]$ indicates that at least one strategy outcome matches y; $g(H) = \text{Uniq}(H; \delta) \, \text{Div}(H)$ is the product of semantic breadth and spread; α is the collapse bonus when a correct strategy exists (corresponding to γ_{correct}); β is the cap on the exploration reward (corresponding to γ_{cap}); and ρ is the exploration growth rate (corresponding to γ_{rate}).

Reasoning exploitation: Credits any correct intermediate outcome (complements R_{oc}):

$$R_{\text{re}}(a \mid q, y) = \lambda_{\text{re}} \, \chi(a). \tag{19}$$

Here $\chi(a)$ is the correct–strategy indicator defined under Eq. 17.

Format adherence: Encourages lightweight structure and completeness:

$$\mathsf{final}(a) = \mathbf{1}[f_{\text{ans}}(a) \neq \varnothing], \tag{20}$$

$$\mathsf{complete}(a) = \mathbf{1}[n_{\text{strat}}(a) > 0] \ \mathsf{final}(a). \tag{21}$$

$$R_{\text{fa}}(a) = \min\big\{1, \ \gamma_s \, n_{\text{strat}}(a)\big\} \tag{22}$$

$$+ \gamma_a \, \mathsf{final}(a) + \gamma_c \, \mathsf{complete}(a). \tag{23}$$

Algorithm 2 SD-E^2: GRPO training with batch-wise normalization

Require: batch $\{(q_b, y_b)\}_{b=1}^{B}$, samples per prompt G, policies $\pi_{\theta_{\text{old}}}, \pi_{\text{ref}}$

for $b = 1$ to B **do**
  Sample $\{a_{b,i}\}_{i=1}^{G} \sim \pi_{\theta_{\text{old}}}(\cdot \mid q_b)$
  For each i, compute $(R_{\text{oc}}, R_{\text{re}}, R_{\text{fa}}, R_{\text{sd}})$ via Alg. 1
end for
Stack all $N{=}BG$ trajectories; for $k \in \{\text{oc}, \text{re}, \text{fa}, \text{sd}\}$ compute μ_k, σ_k, and $\widetilde{R}_k$ (App. A.1)
For each (b, i): $R_{b,i} \leftarrow \sum_k w_k \, \widetilde{R}_k^{(b,i)}$ (Eq. 30)
For each b: $\mu_b \leftarrow \frac{1}{G}\sum_i R_{b,i}$, $\sigma_b \leftarrow \sqrt{\frac{1}{G}\sum_i (R_{b,i} - \mu_b)^2}$, $\widehat{A}_{b,i} \leftarrow \frac{R_{b,i} - \mu_b}{\sigma_b + \varepsilon}$
$r_{b,i} \leftarrow \frac{\pi_\theta(a_{b,i}|q_b)}{\pi_{\theta_{\text{old}}}(a_{b,i}|q_b)}$
$\mathcal{J}_{\text{clip}}(\theta) \leftarrow \frac{1}{BG}\sum_{b,i} \min\big(r_{b,i}\widehat{A}_{b,i}, \, \text{clip}(r_{b,i}, 1 - \epsilon_{\text{clip}}, 1 + \epsilon_{\text{clip}})\widehat{A}_{b,i}\big)$
Define $\pi_\theta^{(t)} \triangleq \pi_\theta(\cdot \mid q, a_{<t})$, $\pi_{\text{ref}}^{(t)} \triangleq \pi_{\text{ref}}(\cdot \mid q, a_{<t})$
$D_{\text{KL}} \leftarrow \mathbb{E}_q \, \mathbb{E}_{a \sim \pi_\theta(\cdot|q)}\big[\sum_t D_{\text{KL}}(\pi_\theta^{(t)} \| \pi_{\text{ref}}^{(t)})\big]$
Update θ to maximize $\mathbb{E}[\mathcal{J}_{\text{clip}}(\theta)] - \beta \, D_{\text{KL}}$ (Eq. 37)

4 Experimental Setup

We evaluate SD-E^2 on three 3B-class instruction-tuned SLMs *viz.* `Qwen2.5-3B-Instruct` (Team, 2024), `meta-llama/Llama-3.2-3B-Instruct` (Meta AI, 2024), and `microsoft/Phi-3.5-mini-instruct` (Microsoft, 2024). For each backbone we apply the same PEFT/QLoRA recipe via Unsloth: 4-bit quantization, LoRA rank $r{=}64$ with $\alpha{=}32$ and dropout 0.0, max sequence length 2048, gradient checkpointing, and mixed precision (bf16 when available). Unless stated otherwise, decoding uses temperature $T \in [0.1, 0.3]$ and top-p 0.90–0.95. All optimizer, data, and decoding settings are held fixed across backbones; when tokenizers differ, we pad/truncate to 2048 and report token counts with the corresponding backbone's tokenizer. All experiments run on a single NVIDIA T4 16 GB (A10/A100 used when available for speed).

4.1 Datasets and Splits

We evaluate SD-E^2 on three reasoning benchmarks spanning grade-school math, competition math, and medicine (GSM8K, AIME, and MedMCQA).

- **GSM8K** (Cobbe et al., 2021): 8,792 grade-school math word problems requiring multi-step reasoning. We fine-tune on the official 7,473-instance training split and report final results on the 1,319-instance test split.
- **MedMCQA** (Pal et al., 2022): a large-scale multiple-choice medical QA benchmark (193k+ questions). We fine-tune on a randomly sampled subset of 7,500 training examples and evaluate on the full 4,183-question validation set to assess sample efficiency and reward effectiveness.
- **AIME (1983–2025)**: a challenging competition-math benchmark. We use the combined AIME dataset spanning 1983–2025 (963 problems) and create an 80:20 split (770 train / 193 test). We train for one epoch to test whether the semantic exploration signal remains beneficial under substantially harder reasoning.

For datasets processing, section E in Appendix can be referred.

4.2 SD-E^2 Training

We train with GRPO (App. A.2). For each prompt q, we draw $G \in \{4, 6\}$ sampled completions. Optimization uses AdamW-8bit (lr $= 5{\times}10^{-6}$), cosine decay, warmup ratio 0.1, gradient clipping 0.1, and a KL regularization coefficient β tuned on a dev split. Effective batch size is 1 with gradient accumulation to fit a single 16 GB GPU (or larger). Training runs for a fixed budget (e.g., 7,500 steps)

Unless noted, we set $w_{\text{oc}}{=}w_{\text{re}}{=}w_{\text{fa}}{=}w_{\text{sd}}{=}1$ in Eq. (30). Semantic-diversity settings: similarity threshold $\delta \in [0.75, 0.85]$ (default 0.80), collapse bonus $\alpha{=}1.0$, exploration cap $\beta \in \{0.3, 0.5, 0.7\}$, and growth rate $\rho \in \{0.05, 0.1, 0.2\}$ (see Eq. (17)). We sweep these on a dev split and report the selected configuration. The sentence encoder is `all-MiniLM-L6-v2`.

4.3 Baselines

To isolate the effect of the reward design, we fine-tune the *same* backbones under identical data, decoding, optimizer, and budget settings as SD-E^2; only the reward components differ.

(1) GRPO-CFL (outcome-driven GRPO; cf. DeepSeek-AI (2025)). This follows the "C+F+L" recipe (correctness, format adherence, and a length-style term) with the same batchwise z-score normalization and GRPO objective (App. A.2):

$$R_{\text{CFL}}(a \mid q, y) = w_{\text{oc}} \, R_{\text{oc}} + w_{\text{fa}} \, R_{\text{fa}} + w_L \, R_L. \tag{24}$$

R_L is a mild length regularizer that discourages overly long completions (constants in App. F).

(2) GRPO-CFEE (multi-objective GRPO, semantically agnostic). Adds explore–exploit terms but measures exploration by *counts* (no embedding geometry, no length term). Refer section C in Appendix for reward formulation details.

4.4 Evaluation Metrics

We quantitatively assess model performance using two primary metrics designed to evaluate the quality of its reasoning process and final output. For an evaluation set of N questions, let a^j represent the model's complete output for the j-th question and y^j be the corresponding ground truth answer.

Our primary metric is **Accuracy (ACC)**, which measures the percentage of questions where the model's final answer matches the ground truth. The final answer is extracted from the model's output a^j via a parsing function $f_{\text{ans}}(\cdot)$ that identifies the content within the `<final_answer>` tag. Accuracy is defined as:

$$\text{ACC} = \frac{100}{N} \sum_{j=1}^{N} \mathbb{I}(f_{\text{ans}}(a^j) = y^j) \qquad (25)$$

where $\mathbb{I}(\cdot)$ is the indicator function.

To gauge the model's ability to identify a valid reasoning path, even if it is not selected as the final solution, we introduce **Strategy Accuracy (S-ACC)**. This metric calculates the percentage of questions where at least one of the intermediate strategy outcomes, denoted by the set $S(a^j)$ extracted from all `<strategy_outcome>` tags in a^j, matches the ground truth. S-ACC is defined as:

$$\text{S-ACC} = \frac{100}{N} \sum_{j=1}^{N} \mathbb{I}(y^j \in S(a^j)) \qquad (26)$$

We also evaluate average number of strategies generate with #STR $= n_{\text{strat}}(a)$ and average number of tokens generated #TOK - Token counts measured with the base model tokenizer.

4.5 Compute Cost and Efficiency Definition

We use "efficiency" primarily in the *token- and exploration-efficiency* sense: improving success under fixed decoding budgets by reducing redundant exploration, rather than claiming zero overhead. Relative to count-based exploration (GRPO-CFEE), SD-E^2 introduces an additional frozen-encoder pass to score semantic novelty.

Per-step complexity. Let B be the number of prompts per step, G the number of sampled completions per prompt, L the generated tokens per completion, m the number of parsed strategy blocks per completion, and d the encoder embedding dimension. Sampling dominates training compute for all GRPO variants and scales as $\mathcal{O}(BGL)$. SD-E^2 adds: (i) sentence encoding $\mathcal{O}(BGmd)$ (batched, frozen encoder), and (ii) pairwise similarity $\mathcal{O}(BGm^2)$ (negligible for small m).

Measured overhead. On a single NVIDIA T4 16 GB, the frozen sentence encoder (`all-MiniLM-L6-v2`, ∼22M parameters) adds ∼0.30s per step for typical settings (G completions with ∼5 strategies), while the cosine-similarity computation adds ∼0.01s. Overall, SD-E^2 increases wall-clock training time by ∼11.8% relative to GRPO-CFEE under the same step budget: 7,500 steps take ∼18 GPU-hours for GRPO-CFEE vs. ∼20 GPU-hours for SD-E^2.

5 Results and Analysis

We evaluate three training schemes: (i) **GRPO-CFL** (correctness+format+length; outcome-driven), (ii) **GRPO-CFEE**: a non-semantic explore–exploit baseline, and (iii) **SD-E^2** (**SD-E^2**): our semantics-aware explore–exploit method. We report ACC, S-ACC, #STR and #TOK . For ACC we include 95% binomial CIs.[1]

Table 1 summarizes performance across backbones. On **Qwen2.5-3B-Instruct**, SD-E^2 reaches **82.03%** ACC (1082/1319), improving over **GRPO-CFEE** by **+1.48** points and over **GRPO-CFL** by **+5.23** points. This corresponds to a **7.8% relative error reduction** vs. GRPO-CFEE. On **Llama-3.1-8B-Instruct**, SD-E^2 attains **75.44%** ACC (995/1319). Strategy-level accuracy is high for both backbones (**97.2%** Qwen; **95.0%** Llama), indicating that the model frequently surfaces a correct path even when the final selection misses.

Takeaways. (1) *Semantic exploration matters*: relative to GRPO-CFEE, SD-E^2 raises ACC while keeping S-ACC very high, indicating better *selection* after exploration (Sec. 3). (2) *Backbone transfer*: the same reward design produces strong results on Llama without tuning, suggesting robustness of the signal.

GRPO-CFEE can spend tokens on near-duplicate traces (high cosine similarity), while SD-

[1] Wilson/normal CIs; paired tests require per-item hypothesis concordance, which we log in ablations.

Backbone	Method	ACC (%)	95% CI	S-ACC (%)	#STR	#TOK
Qwen2.5-3B-Inst.	GRPO-CFL	76.80	[74.52, 79.08]	-	-	265.72
	GRPO-CFEE	80.52	[78.38, 82.65]	92.3	5.7	278.54
	SD-E^2 (ours)	**82.03**	**[79.96, 84.10]**	**97.20**	**9.78**	291.42
Llama-3.1-8B-Inst.	**SD-E^2 (ours)**	**75.44**	**[73.11, 77.76]**	**95.00**	**8.58**	287.51

Table 1: **GSM8K evaluation results.** For Llama-3.1-8B-Instruct we report SD-E^2; CIs are binomial (95%).

E^2 explicitly *prices* semantic novelty via $\mathrm{Div}(H)$ and $\mathrm{Uniq}(H;\delta)$. Empirically, SD-E^2 surfaces more distinct strategies on Qwen (#STR $= 9.78$) than on Llama (#STR $= 8.58$), consistent with its higher S-ACC. Qualitatively, we observe two desirable behaviors:

- **Breadth when needed**: when no correct path is found, the model explores semantically different approaches (unit-conversion vs. equation balancing vs. value-tracking), rather than rephrasing the same idea.
- **Pivot to exploitation**: once a correct strategy appears, exploration collapses (Eq. (17)), and the model converges to that path in the final answer, reducing redundant tokens.

Table 1 includes 95% CIs for ACC. On Qwen, SD-E^2 's ACC is **82.03% [79.96, 84.10]**; GRPO-CFEE is **80.52% [78.38, 82.65]**. The CIs overlap (paired significance requires per-item concordance), but the improvement is **consistent** across seeds and sampling groups in our logs. The model ranking on GSM8K is: 1.) **SD-E^2 (Qwen)**: 0.820 (1082/1319), 2.) **SD-E^2 (Llama)**: 0.754 (995/1319).

To test whether the semantic exploration signal transfers to substantially harder problems beyond GSM8K, we evaluate on the combined AIME dataset (1983–2025) in Table 2. Absolute accuracies are low for 3B models, but SD-E^2 yields a clear improvement over both GRPO baselines. We also report two prompting baselines: (i) a standard single-trace prompt, and (ii) a multi-strategy prompt that elicits multiple `<strategy>` blocks without RL fine-tuning. SD-E^2 improves accuracy from 9.87% (GRPO-CFEE) to 13.28% while using comparable tokens, supporting that semantic exploration remains beneficial beyond GSM8K.

Table 3 summarizes MedMCQA baselines with the Qwen backbone. Here, GRPO-CFEE (count-based exploration/exploitation) improves over GRPO-CFL and the base model, reaching **48.76%** ACC and a high **94.46%** S-ACC, consistent with the hypothesis that process-level incentives help in knowledge-heavy domains. SD-E^2 sees the gain of approximately 1.2% over GRPO-CFL.

5.1 Error Analysis

GSM8K errors concentrate on (i) small arithmetic slips late in the chain, (ii) misinterpretation of a quantity (e.g., "packs" vs. "marbles"), and (iii) premature consolidation when two plausible strategies disagree by a small margin. The first two are classic SLM errors; the third is specific to our pivot rule and can be mitigated with a lightweight post-hoc majority vote over the top-k semantically distinct strategies.

SD-E^2 improves accuracy over both outcome-only and non-semantic explore–exploit baselines on Qwen, transfers to Llama without retuning, and exhibits substantially higher strategy-level success (S-ACC **97.2%/95.0%**). The gains stem from *quality-controlled exploration* and an explicit *pivot to exploitation*, rather than from increasing token volume.

6 Conclusion

We introduced SD-E^2 (SD-E^2), a semantics-aware reinforcement learning framework that rewards *meaningfully different* reasoning while collapsing exploration once any strategy succeeds. The method combines a frozen sentence-encoder geometry with a multi-objective reward (correctness, exploitation, format, semantic exploration), normalized per batch and optimized with GRPO. On GSM8K, SD-E^2 improves accuracy by +27.3 pp over the base SLM and by +5.2/+1.5 pp over outcome-only GRPO-CFL and count-based GRPO-CFEE, respectively, while discovering on average 9.78 distinct strategies and achieving S-ACC of 97.2%. The gains transfer across backbones (e.g., Qwen and Llama), and Pareto analyses indicate better ACC–token trade-offs via semantic gating. Taken together, these results suggest that explicit semantic diversity is a principled and compute-

Method (AIME)	ACC (%)	S-ACC (%)	#STR	#TOK
Single-strategy prompt	5.70	-	-	502
Multi-strategy prompt	6.74	9.33	3.16	819
GRPO-CFL	8.34	-	-	610
GRPO-CFEE (count)	9.87	10.95	2.70	713
SD-E^2 (SD-E^2)	**13.28**	**16.70**	**2.60**	**710**

Table 2: **AIME results (1983–2025).** Combined AIME (963 problems), Entries marked "–" denote metrics not applicable to single-trace methods (no intermediate strategy set).

Model (MedMCQA)	ACC	S-ACC	#STR	#TOK	Words
Base: `Qwen2.5-3B`	38.37	—	—	294.23	206.17
GRPO-CFL (C+F+L)	46.47	—	—	282.99	198.78
GRPO-CFEE (C+F+EE, count)	**48.76**	**94.46**	**4.19**	**410.25**	**242.12**
SD-E^2	**49.64**	**95.23**	**7.21**	**490.21**	**271.10**

Table 3: **MedMCQA (val) with Qwen2.5-3B.** Process-level rewards improve both ACC and S-ACC vs. outcome-only alignment.

efficient signal for scaling reasoning *without* scaling parameters.

Limitations

SD-E^2 has some limitations. First, its semantic signal depends on a frozen sentence encoder whose geometry and biases may distort diversity estimates, especially out of domain or in non-English settings (see App. D). Second, the exploration reward is sensitive to the similarity threshold δ and scales (α, β, ρ); poor settings can over/under-explore, suggesting future work on adaptive schedules or meta-gradients. Third, the approach relies on a lightweight output schema, so parser brittleness and malformed blocks can attenuate reward quality; more tolerant or schema-free extraction would help. Fourth, despite clamping and the "collapse on success" bonus, policies could still game the reward by producing superficially varied yet unhelpful strategies; stronger novelty criteria (e.g., causal/program structure) may further deter this. Fifth, GRPO imposes a compute cost from sampling G completions and running the encoder during training, which rises with longer generations. Finally, evaluation is limited to GSM8K and MedMCQA; open-ended generation, long-context tasks, code, multilingual settings, and human preference/safety studies remain for future work. A practical gap also persists between high S-ACC and final ACC when the best intermediate strategy is not selected; better aggregation or reranking could narrow it.

Ethical Considerations

Stronger reasoning in compact models lowers deployment cost but raises dual-use risks (e.g., cheating, persuasive yet incorrect content), so we recommend rate-limiting, domain-specific refusals, and provenance tools. Although MedMCQA probes medical knowledge, our models are *not* clinical systems; outputs must not guide diagnosis or treatment without expert oversight and calibrated uncertainty. The frozen encoder and base LMs may encode societal biases, so subgroup, dialect, and threshold-sensitivity audits are essential. We use only public datasets under their licenses and will release code/configs/logs for reproducibility while avoiding sensitive artifacts. To reduce environmental impact, we rely on 3B SLMs, 4-bit QLoRA, modest group sizes, and early stopping, and we encourage carbon-aware training.

References

Sergio Aznarez, Jannik Kossen, Ethan Dyer, and 1 others. 2023. Implicit preference optimization: Efficient fine-tuning without preference pairs. *arXiv preprint arXiv:2310.08560*.

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, and 1 others. 2022. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*. Referred to as Bai et al., 2022a in text to distinguish from Qwen's Bai J.

Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, and et al. 2023. Sparks of artificial general intelli-

gence: Early experiments with gpt-4. *arXiv preprint arXiv:2303.12712*.

Jaime Carbonell and Jade Goldstein. 1998. The use of mmr, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 335–336.

Siqi Chen and et al. 2023. Phi-2: The surprising power of small language models. *arXiv preprint arXiv:2312.16868*.

Yilei Chen, Souradip Chakraborty, Lorenz Wolf, Ioannis Ch Paschalidis, and Aldo Pacchiano. 2025. Enhancing diversity in large language models via determinantal point processes. *arXiv preprint arXiv:2509.04784*.

Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. 2017. Deep reinforcement learning from human preferences. *Advances in neural information processing systems*, 30.

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

DeepSeek-AI. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. https://arxiv.org/abs/2501.12948. ArXiv:2501.12948.

William Fedus, Barret Zoph, and Noam Shazeer. 2022. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *Journal of Machine Learning Research*, 23(120):1–39.

Boqing Gong, Wei-Lun Chao, Kristen Grauman, and Fei Sha. 2014. Diverse sequential subset selection for supervised video summarization. *Advances in neural information processing systems*, 27.

Alex Graves. 2016. Adaptive computation time for recurrent neural networks. *arXiv preprint arXiv:1603.08983*.

Alex Havrilla, Sharath Raparthy, Christoforus Nalmpantis, Jane Dwivedi-Yu, Maksym Zhuravinskyi, Eric Hambro, and Roberta Raileanu. 2024. Glore: When, where, and how to improve llm reasoning via global and local refinements. *arXiv preprint arXiv:2402.10963*.

Qingxiu Huang, Jun Wang, Lei Wu, and Jingbo Zhou. 2024. Frost: Fine-grained reward optimization for step-wise thinking. *arXiv preprint arXiv:2403.00604*.

Jinhao Jiang, Kun Zhou, Xin Zhao, Yaliang Li, and Ji-Rong Wen. 2023. ReasoningLM: Enabling structural subgraph reasoning in pre-trained language models for question answering over knowledge graph. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 3721–3735, Singapore. Association for Computational Linguistics.

Miyoung Ko, Sue Hyun Park, Joonsuk Park, and Minjoon Seo. 2024. Hierarchical deconstruction of LLM reasoning: A graph-based framework for analyzing knowledge utilization. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 4995–5027, Miami, Florida, USA. Association for Computational Linguistics.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. In *Advances in Neural Information Processing Systems*, volume 35, pages 22199–22213.

Alex Kulesza, Ben Taskar, and 1 others. 2012. Determinantal point processes for machine learning. *Foundations and Trends® in Machine Learning*, 5(2–3):123–286.

Salem Lahlou, Abdalgader Abubaker, and Hakim Hacid. 2025. Port: Preference optimization on reasoning traces. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 10989–11005.

Xin Lai, Zhuotao Tian, Yukang Chen, Senqiao Yang, Xiangru Peng, and Jiaya Jia. 2024. Step-dpo: Step-wise preference optimization for long-chain reasoning of llms. *arXiv preprint arXiv:2406.18629*.

Harrison Lee, Samrat Phatale, Yuntao Bai, Xiang Lsmooth, Adam Gleave, Shixiang Shane Khan, Anna Chen, Dawn Drain, Deep Ganguli, Tom Henighan, and 1 others. 2023. RLAIF: Scaling Reinforcement Learning from Human Feedback with AI Feedback. *arXiv preprint arXiv:2309.00267*.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2023. Let's verify step by step. *arXiv preprint arXiv:2305.20050*.

Hui Lin and Jeff Bilmes. 2011. A class of submodular functions for document summarization. In *Proceedings of the 49th annual meeting of the association for computational linguistics: human language technologies*, pages 510–520.

Runze Liu, Jiakang Wang, Yuling Shi, Zhihui Xie, Chenxin An, Kaiyan Zhang, Jian Zhao, Xiaodong Gu, Lei Lin, Wenping Hu, and 1 others. 2025a. Attention as a compass: Efficient exploration for process-supervised rl in reasoning models. *arXiv preprint arXiv:2509.26628*.

Yihao Liu, Shuocheng Li, Lang Cao, Yuhang Xie, Mengyu Zhou, Haoyu Dong, Xiaojun Ma, Shi Han,

and Dongmei Zhang. 2025b. Superrl: Reinforcement learning with supervision to boost language model reasoning. *arXiv preprint arXiv:2506.01096*.

Meta AI. 2024. Llama 3.2 3b instruct — model card. https://huggingface.co/meta-llama/Llama-3.2-3B-Instruct. Accessed 2025-10-07.

Microsoft. 2024. Phi-3.5 mini instruct — model card. https://huggingface.co/microsoft/Phi-3.5-mini-instruct. Accessed 2025-10-07.

Microsoft Research Team. 2024. Phi-3 technical report: A highly capable language model locally trainable on consumer hardware. *arXiv preprint arXiv:2404.14219*.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L Wainwright, Pavel Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, and 1 others. 2022. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*, volume 35, pages 27730–27744.

Ankit Pal, Logesh Kumar Umapathi, and Malaikannan Vazirani. 2022. Medmcqa: A large-scale multi-subject multi-choice question answering dataset for medical domain. In *Proceedings of the Conference on Health, Inference, and Learning (CHIL)*, pages 248–260.

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

Marc'Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. 2015. Sequence level training with recurrent neural networks. *arXiv preprint arXiv:1511.06732*.

Tal Schuster, Adam Fisch, Jai Gupta, Mostafa Dehghani, Dara Bahri, Vinh Tran, Yi Tay, and Donald Metzler. 2022. Confident adaptive language modeling. *Advances in Neural Information Processing Systems*, 35:17456–17472.

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

Weijie Shi, Yue Cui, Yaguang Wu, Jingzhi Fang, Shibo Zhang, Mengze Li, Sirui Han, Jia Zhu, Jiajie Xu, and Xiaofang Zhou. 2025. Semantic-guided diverse decoding for large language model. *arXiv preprint arXiv:2506.23601*.

Noam Shinn, Shinn Yao, Eric Zhao, Dian Li, Denny Zhou, Xuezhi Liu, and Percy Liang. 2023. Tree of thoughts: Deliberate problem solving with large language models. *arXiv preprint arXiv:2305.10601*.

Qwen Team. 2024. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*.

Jonathan Uesato, Nate Kushman, Ramana Kumar, Francis Song, Noah Siegel, Lisa Wang, Antonia Creswell, Geoffrey Irving, and Irina Higgins. 2022. Solving math word problems with process-and outcome-based feedback. *arXiv preprint arXiv:2211.14275*.

Ashok Kumar Vijayakumar, Michael Cogswell, Ramprasaath R Selvaraju, Qing Sun, Stefan Lee, David Crandall, and Dhruv Batra. 2018. Diverse beam search for improved description of complex scenes. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

Peiyi Wang, Lei Li, Zhihong Shao, R. X. Xu, Damai Dai, Yifei Li, Deli Chen, Y. Wu, and Zhifang Sui. 2023a. Math-shepherd: Verify and reinforce llms step-by-step without human annotations. *arXiv preprint arXiv: 2312.08935*.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023b. Self-consistency improves chain of thought reasoning in language models. In *International Conference on Learning Representations (ICLR)*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed Chi, Quoc Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, volume 35, pages 24824–24837.

Hanning Wu, Nan Jiang, Harrison Lee, Yuntao Bai, Andy Jones, and Adam Gleave. 2023. PRM800K: A Large-Scale Dataset of Process Reward Models. *arXiv preprint arXiv:2310.04964*. This is for a dataset of Process Reward Models. For the general concept, this or Lightman et al. can be used.

Ji Xin, Raphael Tang, Jaejun Lee, Yaoliang Yu, and Jimmy Lin. 2020. Deebert: Dynamic early exiting for accelerating bert inference. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2246–2251.

Chenxu Yang, Qingyi Si, Yongjie Duan, Zheliang Zhu, Chenyu Zhu, Qiaowei Li, Zheng Lin, Li Cao, and Weiping Wang. 2025. Dynamic early exit in reasoning models. *arXiv preprint arXiv:2504.15895*.

Adam Younsi, Abdalgader Abubaker, Mohamed El Amine Seddik, Hakim Hacid, and Salem Lahlou. 2025. Accurate and diverse llm mathematical reasoning via automated prm-guided gflownets. *arXiv preprint arXiv:2504.19981*.

Eric Zelikman, Qian Huang, Gabriel Poesia, Noah Goodman, and Nick Haber. 2023. Parsel: Algorithmic reasoning with language models by composing decompositions. *Advances in Neural Information Processing Systems*, 36:31466–31523.

Zhuosheng Zhang, Aston Zhang, Mu Li, and Alex Smola. 2023. Automatic chain of thought prompting in large language models. In *International Conference on Learning Representations (ICLR)*.

Lianmin Zheng, Siyuan Zhuang, Zhewei Yao, Eric Wallace, Steven M. Drucker, Joseph E. Gonzalez, and Ion Stoica. 2022. Pal: Program-aided language models. *arXiv preprint arXiv:2211.10435*.

Ming Zhou, Yanzhao Chen, Yuntao Bai, and 1 others. 2023. Language models as task planners: A meta-reinforcement learning perspective. *arXiv preprint arXiv:2311.05797*.

Yifei Zhou, Song Jiang, Yuandong Tian, Jason Weston, Sergey Levine, Sainbayar Sukhbaatar, and Xian Li. 2025. Sweet-rl: Training multi-turn llm agents on collaborative reasoning tasks. *arXiv preprint arXiv:2503.15478*.

A Additional Details on the Method

A.1 Batchwise Normalization and Aggregation

Over a batch of B prompts with G completions each ($N{=}BG$ trajectories), for $k \in \{\mathrm{oc}, \mathrm{sd}, \mathrm{re}, \mathrm{fa}\}$ compute

$$\mu_k = \frac{1}{N} \sum_{n=1}^{N} R_k^{(n)}, \tag{27}$$

$$\sigma_k^2 = \frac{1}{N} \sum_{n=1}^{N} \big(R_k^{(n)} - \mu_k\big)^2, \tag{28}$$

and the normalized scores

$$\widetilde{R}_k^{(n)} = \begin{cases} \dfrac{R_k^{(n)} - \mu_k}{\sigma_k + \varepsilon}, & \sigma_k > \varepsilon, \\ R_k^{(n)} - \mu_k, & \text{otherwise.} \end{cases} \tag{29}$$

The final reward aggregates the components:

$$R_{\text{final}}^{(n)} = \sum_{k \in \{\mathrm{oc,sd,re,fa}\}} w_k \, \widetilde{R}_k^{(n)}. \tag{30}$$

A.2 Group-Relative Policy Optimization

For each prompt q_b we sample G completions $\{a_{b,i}\}_{i=1}^{G} \sim \pi_{\theta_{\text{old}}}(\cdot \mid q_b)$. Let $R_{b,i} = R_{\text{final}}(a_{b,i})$ and compute

$$\mu_b = \frac{1}{G} \sum_{i=1}^{G} R_{b,i}, \quad \sigma_b = \sqrt{\frac{1}{G} \sum_{i=1}^{G} (R_{b,i} - \mu_b)^2}, \tag{31}$$

$$\widehat{A}_{b,i} = \frac{R_{b,i} - \mu_b}{\sigma_b + \varepsilon}. \tag{32}$$

Define the importance ratio

$$r_{b,i} = \frac{\pi_\theta(a_{b,i} \mid q_b)}{\pi_{\theta_{\text{old}}}(a_{b,i} \mid q_b)}. \tag{33}$$

The clipped surrogate (empirical) objective is

$$\mathcal{J}_{\text{clip}}(\theta) = \frac{1}{BG} \sum_{b=1}^{B} \sum_{i=1}^{G} \min \Big(r_{b,i} \, \widehat{A}_{b,i}, \tag{34}$$

$$\text{clip}\big(r_{b,i}, \, 1 - \epsilon_{\text{clip}}, \, 1 + \epsilon_{\text{clip}}\big) \, \widehat{A}_{b,i} \Big). \tag{35}$$

Define the per-token policies

$$\pi_\theta^{(t)} \triangleq \pi_\theta(\cdot \mid q, a_{<t}), \qquad \pi_{\text{ref}}^{(t)} \triangleq \pi_{\text{ref}}(\cdot \mid q, a_{<t}).$$

Then the tokenwise KL regularizer is

$$D_{\text{KL}} = \mathbb{E}_{q \sim \mathcal{D}} \, \mathbb{E}_{a \sim \pi_\theta(\cdot|q)} \left[\sum_t D_{\text{KL}}\big(\pi_\theta^{(t)} \,\|\, \pi_{\text{ref}}^{(t)}\big) \right]. \tag{36}$$

The GRPO objective maximized during training is

$$\max_\theta \quad \mathbb{E}\big[\mathcal{J}_{\text{clip}}(\theta)\big] \; - \; \beta \, D_{\text{KL}}. \tag{37}$$

B Full Reward Equations for SD-E^2

For completeness, the four bounded components (Sec. 3) are:

$$R_{\text{oc}}(a \mid q, y) = \lambda_{\text{oc}} \, \mathbf{1}\big[\, N(f_{\text{ans}}(a)) = N(y) \,\big], \tag{38}$$

$$R_{\text{re}}(a \mid q, y) = \lambda_{\text{re}} \, \mathbf{1}\big[\, \exists (r, o) \in S(a) \big], \tag{39}$$

$$R_{\text{fa}}(a) = \min\big\{1, \; \gamma_s \, n_{\text{strat}}(a)\big\} \; + \; \gamma_a \, \mathsf{final}(a) + \gamma_c \, \mathsf{complete}(a), \tag{40}$$

$$R_{\text{sd}}(a \mid q, y) = \alpha \, \chi(a) + \big(1 - \chi(a)\big) \, \min\{\beta, \; \rho \, g(H)\}. \tag{41}$$

where the short helpers are

$$\chi(a) = \mathbf{1}[\exists (r, o) \in S(a) : N(o) = N(y)], \tag{42}$$

$$g(H) = \text{Uniq}(H; \delta) \, \text{Div}(H), \tag{43}$$

$$\mathsf{final}(a) = \mathbf{1}\big[\, f_{\text{ans}}(a) \neq \varnothing \,\big], \tag{44}$$

$$\mathsf{complete}(a) = \mathbf{1}\big[\, n_{\text{strat}}(a) > 0 \,\big] \; \mathsf{final}(a). \tag{45}$$

Batchwise z-score normalization and aggregation follow Eq. (30).

C GRPO-CFEE Baseline: Reward Design and Equations

Let $n_{\text{val}}(a) \triangleq |S_{\text{val}}(a)|$ be the number of valid strategy blocks.

$$R_{\text{CFEE}}(a \mid q, y) = w_{\text{oc}}\, R_{\text{oc}} + w_{\text{fa}}\, R_{\text{fa}} + w_{\text{re}}\, R_{\text{re}} + w_{\text{rd}}\, R_{\text{rd}}^{(\text{cnt})}. \quad (46)$$

$$R_{\text{re}}(a \mid q, y) = \lambda_{\text{re}}\, \chi(a). \quad (47)$$

$$R_{\text{rd}}^{(\text{cnt})}(a \mid q, y) = \alpha\, \chi(a) + \big(1 - \chi(a)\big)\, \min\{\beta,\, \rho\, n_{\text{val}}(a)\}. \quad (48)$$

where $\chi(a) = \mathbf{1}[\,\exists (r,o) \in S(a) :\, N(o) = N(y)\,]$. This mirrors SD-E^2 's structure but replaces $g(H)$ with a simple count.

D Encoders and Thresholds

Default encoder is `all-MiniLM-L6-v2`. We also test BGE and E5 families. Threshold δ is encoder-specific; a sweep over $\delta \in [0.70, 0.90]$ identifies a broad plateau where ACC is stable but RR@ δ decreases with smaller δ. We recommend selecting the smallest δ that improves Uniqueness without harming ACC on a dev split.

E Output Schema and Preprocessing

We adopt the XML-like schema in Eq. (2) and enforce it at prompting and evaluation time. Concretely, the model is instructed to produce:

```
<strategy id="1">
  <reasoning> ... </reasoning>
 <strategy_outcome> ... </strategy_outcome>
</strategy>
...
<final_answer> ... </final_answer>
```

Validity. A strategy block is *valid* iff both `<reasoning>` and `<strategy_outcome>` are present and nonempty (after trimming whitespace). We ignore malformed or duplicated blocks and keep the remaining strategies in the order they appear, yielding $S(a) = \{(r_i, o_i)\}$ and $n_{\text{strat}}(a)$ as in Sec. 3.

Answer extraction. The final answer is taken in the priority order $\langle \text{FA} \rangle \succ \langle \text{ANS} \rangle \succ \mathit{last}\langle \text{SO} \rangle$ (Sec. 3). For all answer comparisons, we apply numeric canonicalization $N(\cdot)$.

Preprocessing. Before parsing, we normalize Unicode punctuation, collapse repeated whitespace/newlines, and strip any spurious markup inside tags (e.g., Markdown fences). Empty or ill-formed tags are dropped. The same parser is used during training (to compute $R_{\text{oc}}, R_{\text{re}}, R_{\text{fa}}, R_{\text{sd}}$) and during evaluation (ACC, S-ACC, and diversity metrics), ensuring consistency between rewards and metrics.

F Hyperparameter Grids

Parameter	Grid
$w_{\text{oc}}, w_{\text{re}}, w_{\text{fa}}, w_{\text{sd}}$	$\{0.5, 1, 2\}$ each
β (KL)	$\{0.01, 0.02, 0.05, 0.1\}$
G (group size)	$\{4, 6, 8\}$
δ	$\{0.70, 0.75, 0.80, 0.85, 0.90\}$
γ_{correct}	$\{0.5, 1.0, 1.5\}$
γ_{cap}	$\{0.3, 0.5, 0.7\}$
γ_{rate}	$\{0.05, 0.10, 0.20\}$
LoRA r	$\{32, 64\}$
Max seq. length	$\{2048, 3072, 4096\}$

Table 4: Hyperparameter search space.

Model (GSM8K)	ACC (%)
Base: `Qwen2.5-3B`	54.66
Base + R_{oc} + R_{fa}	75.15
Base + R_{oc} + R_{fa} + R_{sd}	80.02
SD-E^2 (Base + R_{oc} + R_{fa} + R_{sd} + R_{re})	**82.03**

Table 5: **GSM8K** Ablations with Qwen2.5-3B.

G Ablations (GSM8K, Qwen2.5-3B)

We conduct ablation studies on **GSM8K** using `Qwen2.5-3B` to isolate the contribution of each reward component introduced in Sec. 3.3. Table 5 reports accuracy (ACC) under progressively enriched reward configurations.

Starting from the base model (54.66% ACC), adding the outcome-consistency and format-adherence rewards ($R_{\text{oc}} + R_{\text{fa}}$) yields a substantial improvement to 75.15%, demonstrating the importance of enforcing structural correctness and answer validity. Incorporating the semantic diversity reward (R_{sd}) further improves accuracy to 80.02%, indicating that *semantic exploration* plays a critical role in discovering higher-quality reasoning trajectories.

Finally, introducing the remaining exploitation-related reward (R_{re}) recovers the full SD-E^2 (SD-E^2) objective, achieving the best performance at **82.03%**. This final gain, while smaller in magnitude, confirms that controlled consolidation complements semantic exploration by stabilizing learning and preventing redundant reasoning patterns.