SE-OPSD: SELF-ELICITED ON-POLICY SELF-DISTILLATION FOR OPEN-ENDED QA

Anonymous authors
Paper under double-blind review

ABSTRACT

On-policy distillation provides dense token-level supervision on model-generated trajectories. Recent on-policy self-distillation (OPSD) removes the need for external teachers by supervising a model with a privileged copy of itself. In verifiable domains such as mathematics and code, this privilege can be supplied by reference answers or verified reasoning traces. In open-ended QA, however, response quality depends not only on factual content but also on recognizing missing context, calibrating uncertainty, respecting task-specific boundaries, and communicating appropriately. A reference answer may leave these response-framing requirements implicit. A model may identify them when explicitly prompted to analyze a query, yet fail to apply them consistently when answering directly. These elicited judgments constitute natural privileged information: a teacher that has already formed them can supervise a student that has not. We propose SE-OPSD, a reference-free self-distillation method that uses query-specific response context as training-time privileged information. For each query, a frozen teacher copy first elicits a compact, structured checklist of response requirements through complementary views, forming Self-Elicited Privileged Context (SEPC). The student observes only the original query, while the teacher conditions on both the query and SEPC to provide token-level supervision on the student's on-policy trajectory. SE-OPSD thus uses additional inference computation to construct its own supervision and distills these response-framing judgments into single-call answer generation. On HealthBench Consensus, SE-OPSD improves Qwen3-1.7B from 67.86 to 72.88 and Qwen3-4B from 82.15 to 86.58, outperforming reference-answer SFT and reference-conditioned OPSD baselines. Gains in context awareness, communication quality, and completeness, together with improvements on general-domain WildBench, support self-elicited response context as an effective source of supervision for open-ended QA. Our code is available at https://anonymous.4open.science/r/SE-OPSD.

1 INTRODUCTION

Recent advances in language-model post-training have been driven largely by reinforcement learning from outcome-level signals, including verifiable rewards (RLVR) and external judges (Shao et al., 2024; DeepSeek-AI et al., 2025; Yu et al., 2025). These methods are effective when correctness can be reliably evaluated after generation, but these outcome-level signals assign each rollout a scalar reward, with little direct indication of which intermediate tokens or decisions should be revised.

On-policy distillation (OPD) (Lu & Thinking Machines Lab, 2025) addresses this limitation by training a student on its own rollouts while a teacher, conditioned on the same query, provides dense token-level supervision over the student's on-policy prefixes (Agarwal et al., 2024; Gu et al., 2023; LLM-Core Xiaomi, 2026; DeepSeek-AI, 2026). On-policy self-distillation (OPSD) further simplifies this paradigm by removing the need for an external teacher: the same model acts both as the student and as a privileged teacher, with the teacher conditioned on additional information unavailable at inference time (Zhao et al., 2026; Shenfeld et al., 2026). In this way, OPSD offers a route to self-improvement using the model's own conditional predictions. Its effectiveness depends on what privileged information is given to the teacher and how that information is obtained.

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

Reference-conditioned OPSD provides the teacher with an answer or reasoning trace that constrains the target behavior (Zhao et al., 2026); related approaches use richer context or feedback to shape teacher supervision (Ye et al., 2026; Hübotter et al., 2026). In open-ended QA, response quality also depends on decisions that a reference answer may leave implicit: what context is missing, which uncertainties should remain explicit, and how the answer should be tailored to the user. Medical QA is a representative setting in which these decisions are especially consequential (Arora et al., 2025; Lai et al., 2025). A reference answer captures one possible realization of them, while an on-policy teacher must guide continuations from the student's own prefixes. This motivates a central question: *Can a model construct useful teacher-side response context from the query alone, and learn to apply it when answering directly?*

We study *response context* as self-elicited privileged information. Our starting point is a *judgment–generation gap*: a model can articulate useful response requirements when prompted to analyze a query, yet may not consistently apply them when answering directly. Eliciting these requirements makes the model's response-framing knowledge available as an explicit training signal. Direct answer generation bundles these decisions together. Decomposing them into several explicit judgments lets the same model spend additional inference computation on preparing the response context. This context can then guide the teacher during distillation, transferring the benefit of explicit preparation into ordinary direct-answer generation.

We propose **SE-OPSD**, a **reference-free Self-Elicited** OPSD method for open-ended QA. For each training query, a frozen copy of the base model generates SEPC, a structured description of missing information, uncertainty handling, risk boundaries, audience requirements, and coverage requirements. The student observes only the original query and directly generates the final answer. The teacher conditions on both the query and SEPC to provide dense token-level supervision on the student's on-policy trajectory. The view-specific prompts organize elicitation, while the model generates the query-specific content. This creates a teacher–student asymmetry through additional computation on the query itself. Training requires only queries, and the resulting model answers in a single call. We evaluate this approach through detailed medical QA experiments and a general-domain instantiation. Our contributions are as follows:

- We introduce SEPC, a compact multi-view representation of query-specific response guidance elicited from the base model itself. SEPC supplies privileged teacher context for open-ended self-distillation without reference answers or per-query rubric annotations.
- We develop SE-OPSD to transfer this guidance into a direct-answer policy through on-policy token-level supervision. In a full-set Qwen3-4B comparison, SE-OPSD outperforms SFT on SEPC-conditioned responses by **2.08** points and retains **86.5%** of the test-time SEPC gain while reducing inference from six model calls to one.
- We demonstrate improvements across Qwen and Llama models, medical benchmarks, and general-domain WildBench. On HealthBench Consensus, Qwen3-4B improves from **82.15** to **86.58**, exceeding both reference-based training baselines. Context-source and view ablations connect these gains to the design of query-specific response guidance; additional grader and training-seed evaluations establish their consistency in the tested settings.

2 Related Work

From autoregressive distillation to on-policy distillation. Knowledge distillation (KD) transfers predictive behavior from a teacher to a student (Hinton et al., 2015), with sequence-level extensions training students on teacher-generated outputs (Kim & Rush, 2016). Such fixed-output supervision can differ from the contexts the student encounters during generation. MiniLLM and Generalized Knowledge Distillation (GKD) incorporate student-generated trajectories into distillation (Gu et al., 2023; Agarwal et al., 2024). This formulation is now often viewed as a representative instance of on-policy distillation (OPD) (Lu & Thinking Machines Lab, 2025). Recent work further extends OPD to black-box teacher access and large-scale post-training pipelines, suggesting that on-policy teacher supervision has become a practical alternative to purely off-policy distillation (Ye et al., 2025; LLM-Core Xiaomi, 2026; DeepSeek-AI, 2026). Our work builds on this on-policy view.

On-policy self-distillation with privileged context. A recent line of work uses the same model as teacher and student under different conditioning contexts, applying teacher supervision on the

Table 1: Comparison with representative post-training paradigms.

Method	Sample	Efficiency	Signal	Teacher	Ref. Answer
SFT	✗ Off-policy	✓ High	✓ Dense	– None	✗ Required
RLVR / GRPO	✓ On-policy	✓ High	✗ Sparse	– None	✗ Required
External OPD	✓ On-policy	✗ Low	✓ Dense	✗ External	✓ None
Reference-based OPSD	✓ On-policy	✓ High	✓ Dense	✓ Self	✗ Required
Ours	**✓ On-policy**	**✓ High**	**✓ Dense**	**✓ Self**	**✓ None**

student's own rollout (Zhao et al., 2026; Shenfeld et al., 2026). Related formulations study agentic privileged information, environment feedback, context-conditioned behavior, and consensus gating (Penaloza et al., 2026; Hübotter et al., 2026; Ye et al., 2026; Stein et al., 2026). Closely related recent work uses response criteria as supervision: RGSD and RuPI condition the self-teacher on supplied rubrics (Rezaei et al., 2026; Bablani et al., 2026), while RCSD learns a rubric generator through reference-conditioned supervision and also studies directly prompted rubrics (Gu et al., 2026). Compute as Teacher converts aggregated answers and generated rubrics into judge-scored RL rewards (Jayalath et al., 2025). These works share the goal of turning richer context or additional inference computation into training signals. Recent analysis also finds that the value of privileged references depends on the student's rollout mode and training dynamics (Zhang et al., 2026b). SE-OPSD focuses on eliciting query-specific response context with the frozen base model and distilling its guidance along student trajectories. Table 1 summarizes the broader post-training paradigms.

3 Method

3.1 Preliminary and Motivation

RL-style post-training. Let $x \sim \mathcal{D}$ be a training query and $y^{(i)}$ a sampled response of length T_i. Common RLVR methods, such as GRPO, sample a group of responses $\{y^{(i)}\}_{i=1}^{G} \sim \pi_{\theta_{\text{old}}}(\cdot \mid x)$, assign each response a scalar reward R_i, and optimize a clipped policy-gradient objective with group-normalized advantages:

$$\hat{A}_i = \frac{R_i - \text{mean}_j(R_j)}{\text{std}_j(R_j) + \delta}, \quad \rho_{i,t} = \frac{\pi_\theta(y_t^{(i)} \mid x, y_{<t}^{(i)})}{\pi_{\theta_{\text{old}}}(y_t^{(i)} \mid x, y_{<t}^{(i)})},$$

$$\mathcal{J}_{\text{GRPO}}(\theta) = \mathbb{E}\left[\frac{1}{G}\sum_{i=1}^{G}\frac{1}{T_i}\sum_{t=1}^{T_i}\min\Big(\rho_{i,t}\hat{A}_i, \text{clip}(\rho_{i,t}, 1-\epsilon, 1+\epsilon)\hat{A}_i\Big) - \beta D_{\text{KL}}(\pi_\theta \| \pi_{\text{ref}})\right].$$

Although this objective is on-policy, its supervision is sparse: the reward R_i is assigned at the response level and then broadcast to token decisions through $\hat{A}_i$. This is effective when reliable outcome rewards are available, but it gives only indirect token-level credit assignment.

On-policy distillation. On-policy distillation (OPD) replaces scalar rewards with dense teacher distributions on the student's own trajectory. The student first samples $y \sim \pi_\theta(\cdot \mid x)$, and at each assistant token position t defines $p_t^\theta(\cdot) = \pi_\theta(\cdot \mid x, y_{<t})$. A teacher policy π_ϕ is evaluated on the same prefix, $q_t^\phi(\cdot) = \pi_\phi(\cdot \mid x, y_{<t})$, and the student minimizes token-level forward KL:

$$\mathcal{L}_{\text{OPD}}(\theta) = \mathbb{E}_{x\sim\mathcal{D},\, y\sim\pi_\theta(\cdot|x)}\left[\frac{1}{\sum_{t=1}^{T} m_t}\sum_{t=1}^{T} m_t\, D_{\text{KL}}\Big(q_t^\phi \,\|\, p_t^\theta\Big)\right], \tag{1}$$

where $m_t \in \{0, 1\}$ masks supervised assistant-answer tokens. Compared with RL-style post-training, OPD provides dense token-level supervision, but it typically relies on an external teacher.

On-policy self-distillation with privileged information. OPSD removes the external teacher by using a privileged copy of the same model. The student still predicts from the deployed input x, while the teacher receives additional training-time privileged information z:

$$p_t^\theta(\cdot) = \pi_\theta(\cdot \mid x, y_{<t}), \qquad q_t^{\bar{\theta},z}(\cdot) = \pi_{\bar{\theta}}(\cdot \mid x, z, y_{<t}).$$

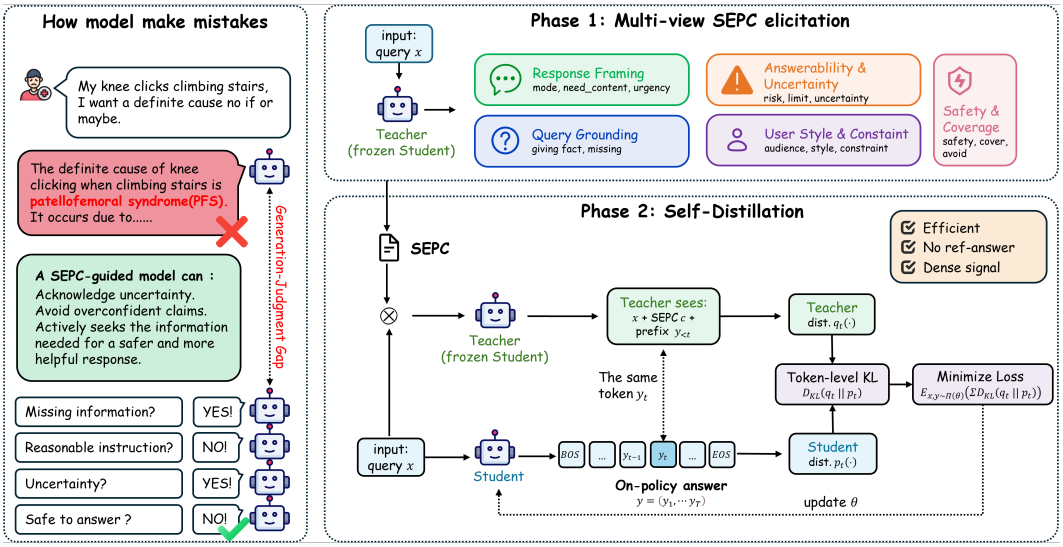


Figure 1: Overview of SE-OPSD. Phase 1 elicits multi-view SEPC signals from a frozen teacher. Phase 2 distills these signals: the student answers from the query alone, while the teacher conditions on the query, SEPC, and student prefix to provide token-level KL supervision.

The OPSD loss is therefore

$$\mathcal{L}_{\text{OPSD}}(\theta; z) = \mathbb{E}_{x\sim\mathcal{D},\, y\sim\pi_\theta(\cdot|x)} \left[\frac{1}{\sum_{t=1}^{T} m_t} \sum_{t=1}^{T} m_t D_{\text{KL}} \left(q_t^{\bar{\theta},z} \,\|\, p_t^{\theta} \right) \right], \tag{2}$$

with stop-gradient applied to the teacher distribution. SE-OPSD constructs the privileged information z through query-specific self-elicitation and uses this standard objective to transfer its guidance.

Choosing the privileged context. In open-ended QA, a reference answer supplies one response but may leave its underlying framing decisions implicit. These decisions matter when the teacher guides continuations from student-sampled prefixes. Our Qwen3-1.7B reference-answer OPSD baseline exhibits excessive structure, repetition, and failure-to-stop behavior (Appendix E), motivating a closer study of teacher-side context.

SE-OPSD therefore uses structured response context as z. Concretely, SEPC is a compact multi-view checklist that specifies how the answer should be framed, grounded, bounded, and covered, without providing the final answer itself.

3.2 Self-Elicited OPSD

Problem setup and notation. Let $\mathcal{D}$ denote the training query dataset and let $x \sim \mathcal{D}$ be a query or conversation. The deployed model is required to generate a direct answer $y = (y_1, \ldots, y_T)$ from x alone. We denote the student policy by π_θ and a frozen teacher copy by $\pi_{\bar{\theta}}$. SE-OPSD constructs training-time privileged information from the query itself. Formally, the privileged context is a structured object c elicited from x: $c = E_{\bar{\theta}}(x)$, where $E_{\bar{\theta}}$ is an elicitation procedure implemented by the teacher copy. The context c is visible only to the teacher during distillation and is never provided to the student at inference time.

Self-elicited privileged context. Rather than eliciting c with a single unconstrained prompt, we decompose response-context construction into K local views. Let $\mathcal{V} = \{v_1, \ldots, v_K\}$ denote the set of view-specific elicitation tasks. For each view v_k, the teacher produces a short structured output

$$c^{(k)} = E_{\bar{\theta}}^{(k)}(x) = \text{Decode}(\pi_{\bar{\theta}}(\cdot \mid x, v_k)), \qquad k = 1, \ldots, K.$$

The final Self-Elicited Privileged Context (SEPC) is obtained by serializing the view outputs:

$$c = \text{Serialize}\left(c^{(1)}, \ldots, c^{(K)}\right).$$

We use $K = 5$ local views: Task Contract; Grounding and State; Correctness and Verification; Audience and Presentation; and Coverage, Quality, and Safety. This multi-view construction treats response framing as a decomposition problem: each view asks the model to make one localized judgment under a constrained schema, rather than requiring a single generation to jointly decide answer mode, missing context, uncertainty, safety boundaries, audience, and coverage. The view-specific prompts define the schema; the model fills its fields from the query to describe how the final answer should be framed. Compact domain-neutral prompt templates are provided in Appendix B.4. The medical and general-domain instantiations share this five-view construction and the same elicitation and distillation procedure.

On-policy self-distillation with SEPC. After SEPC elicitation, SE-OPSD instantiates the privileged information in Eq. 2 as:

$$z = c = E_{\bar{\theta}}(x).$$

The student samples an on-policy answer $y \sim \pi_\theta(\cdot \mid x)$ and predicts from the deployed input only:

$$p_t^\theta(\cdot) = \pi_\theta(\cdot \mid x, y_{<t}).$$

The frozen teacher is evaluated on the same student prefix but additionally conditions on SEPC:

$$q_t^{\bar{\theta},c}(\cdot) = \pi_{\bar{\theta}}(\cdot \mid x, c, y_{<t}).$$

We then optimize Eq. 2 with this teacher distribution. Gradients are applied only to the student parameters, and SEPC is used only during training. At inference time, the model answers directly from x without eliciting or conditioning on c. Figure 1 illustrates the training-time asymmetry between the direct-answer student and the SEPC-conditioned privileged teacher.

4 Experiments

We evaluate how self-elicited privileged context improves open-ended QA and how effectively its guidance transfers into a direct-answer model. Our experiments examine performance against reference-based training baselines, transfer from SEPC-conditioned generation, and applicability across models and domains. HealthBench provides a detailed analysis of context-sensitive response behavior, complemented by general-domain WildBench, knowledge-retention tests, and controlled context ablations.

4.1 Experimental Setup

Training data. We train on a mixture of two open-ended medical QA corpora: RaR-Medicine and K-QA(Gunjal et al., 2025; Manes et al., 2024). For the general-domain experiment, we separately train Qwen3-4B on queries from WildChat-IF with domain-neutral SEPC prompts. More details are provided in Appendix B. For SE-OPSD, only the query is used: reference answers, rubrics, and ground-truth statements are not used to elicit SEPC and are not visible to the student. They are used only for Ref-answer SFT and Ref-answer OPSD baselines.

Evaluation benchmarks. Our main evaluation is HealthBench Consensus, an open-ended rubric-based benchmark in which model responses are scored against physician-written criteria using the official GPT-4.1 grader. We use HealthBench only for evaluation: no HealthBench conversations, rubrics, physician-written responses, or grader outputs are used for training, SEPC elicitation, or model selection. We extend evaluation to HealthBench Hard and HealthBench Professional (Arora et al., 2025; Hicks et al., 2026), and to general-domain WildBench (Lin et al., 2024). Additional HealthBench evaluations use DeepSeek-V4-Flash; we also rescore the same cached Qwen3-4B Consensus responses with this grader to assess grader sensitivity. MedQA and MMLU measure knowledge retention using 5-shot likelihood ranking in the EleutherAI lm-evaluation-harness (Hendrycks et al., 2020; Jin et al., 2021; Gao et al., 2023).

Models and baselines. We evaluate Qwen3-1.7B and Qwen3-4B. Baselines are the base model, Ref-answer SFT on available reference answers, and Ref-answer OPSD, where the student answers from x while the teacher conditions on x and the reference answer. All OPD-style methods use one student rollout per query and the same KL objective; they differ only in teacher-side privileged

Table 2: Main results on HealthBench Consensus and knowledge-retention benchmarks. HealthBench reports overall and axis-level scores with GPT-4.1; MedQA and MMLU report 5-shot likelihood-ranking accuracy. All values are percentages; Δ denotes SE-OPSD minus the corresponding Base model. SFT and OPSD use reference answers. Bold values mark the best method within each model size and metric.

Model	Method	HealthBench Consensus						Knowledge retention	
		Overall	Axes					MedQA	MMLU
			Acc.	Comm.	Instr.	Aware	Complete		
Qwen3-1.7B	Base	67.86	68.85	68.01	79.75	61.07	71.47	42.49	**60.24**
	SFT	69.26	68.56	73.53	84.05	60.72	68.65	**44.54**	60.09
	OPSD	69.34	**71.99**	67.48	79.24	62.18	67.39	42.34	60.19
	SE-OPSD	**72.88**	71.21	**76.81**	**86.07**	**65.20**	**77.42**	43.18	60.13
	Δ	**+5.02**	**+2.36**	**+8.80**	**+6.32**	**+4.13**	**+5.95**	**+0.69**	**-0.11**
Qwen3-4B	Base	82.15	83.08	87.67	89.62	71.30	82.75	55.54	70.07
	SFT	82.64	**84.37**	87.91	90.12	70.29	81.50	**56.48**	**70.82**
	OPSD	82.95	84.24	89.26	87.34	70.79	85.89	56.17	70.10
	SE-OPSD	**86.58**	83.63	**90.12**	**93.46**	**75.22**	**87.47**	55.62	70.25
	Δ	**+4.43**	**+0.55**	**+2.45**	**+3.84**	**+3.92**	**+4.72**	**+0.08**	**+0.18**

information. Additional evaluations use Llama-3.1-8B-Instruct and Qwen3.5-35B-A3B, each eliciting its own SEPC. Section 4.3 compares SE-OPSD with SFT on fixed SEPC-conditioned responses and with the full test-time SEPC pipeline. Main hyperparameters and prompts are in Appendix B.

4.2 Main Results

Table 2 reports HealthBench Consensus performance alongside MedQA and MMLU knowledge-retention results; the HealthBench theme-level breakdown is provided in Appendix B.5. On HealthBench Consensus, SE-OPSD improves over the base model and the reference-based baselines across both model sizes. SE-OPSD uses only query-derived SEPC during training, while Ref-answer SFT and Ref-answer OPSD use the available reference answers. For Qwen3-1.7B, SE-OPSD improves the average score from 67.86 to 72.88, outperforming Ref-answer SFT by **+3.62** and Ref-answer OPSD by **+3.54**. For Qwen3-4B, SE-OPSD improves from 82.15 to 86.58, with a gain of **+3.94** over Ref-answer SFT and **+3.63** over Ref-answer OPSD.

To quantify evaluation uncertainty, we report bootstrap confidence intervals for the HealthBench Consensus overall scores. Figure 2 shows the corresponding error bars, and the full numerical bootstrap results are provided in Appendix B.6. These intervals reflect uncertainty from finite benchmark sampling rather than variance over independent training runs. Paired comparisons on Qwen3-1.7B confirm positive gains over both reference-based baselines, with little change in length relative to SFT and substantially shorter responses than OPSD (Appendix B.6.1).

Response-context behaviors. The HealthBench axis-level results in Table 2 show gains in response-context behaviors. Relative to Base, SE-OPSD improves context awareness by **+4.13/+3.92**, communication quality by **+8.80/+2.45**, and completeness by **+5.95/+4.72** for Qwen3-1.7B/Qwen3-4B. These improvements support the value of explicit guidance about missing context, user needs, and response coverage. The theme-level results in Appendix B.5 further show gains in hedging and emergency referral. By contrast, Ref-answer OPSD provides dense supervision from answer text, but does not necessarily expose the latent response-context decisions behind the answer.

Knowledge retention. We assess whether improvements in open-ended responses preserve general and medical knowledge using MMLU and MedQA, respectively. Table 2 reports accuracy under 5-shot likelihood ranking: the evaluation harness selects the option label with the highest conditional likelihood, without generating a rationale. SE-OPSD remains close to the base model on both benchmarks. MMLU accuracy changes by **-0.11/+0.18** points for Qwen3-1.7B/Qwen3-4B, while MedQA accuracy improves by **+0.69/+0.08** points. These results support knowledge retention

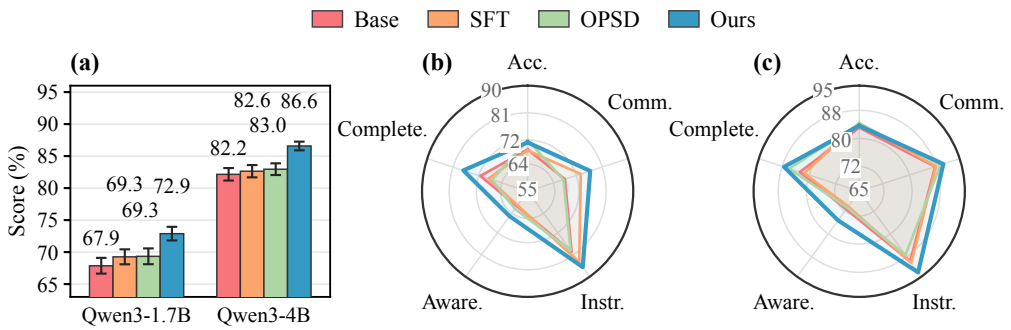


Figure 2: HealthBench Consensus results. (a) Overall scores by model size, with 95% bootstrap confidence intervals over evaluation examples. (b) Qwen3-1.7B and (c) Qwen3-4B axis-level breakdowns.

alongside the improvements in open-ended response quality. Option selection and rubric-based answer evaluation measure complementary aspects of model behavior.

Additional model families and evaluation domains. Table 3 extends the evaluation beyond the two primary Qwen3 models. On HealthBench Consensus, SE-OPSD improves Llama-3.1-8B-Instruct by **5.27** points and Qwen3.5-35B-A3B by **1.44** points. The latter is a mixture-of-experts model with 35B total and 3B active parameters. Each model constructs its own privileged context, demonstrating that the self-elicitation procedure applies across these different backbones.

On HealthBench Professional, the improvements are **2.67** and **3.45** points for Qwen3-4B and Qwen3.5-35B-A3B, respectively. Gains on HealthBench Hard are smaller, at 0.23 and 0.87 points. For general-domain QA, we train a separate Qwen3-4B instance on WildChat-IF queries with domain-neutral five-view prompts. Its WildBench score increases from **20.47** to **23.62**. This extends SE-OPSD's application from medical response guidance to open-ended requests from real users.

Table 3: Additional model and benchmark evaluations. All HealthBench rows use DeepSeek-V4-Flash. WildBench uses a separate Qwen3-4B model trained on WildChat-IF queries. Scores use each benchmark's own scale; Δ is the within-row gain over Base.

Benchmark	Model	Base	SE-OPSD	Δ
HealthBench Consensus	Qwen3-4B	80.11	83.84	+3.73
	Llama-3.1-8B-Instruct	71.17	76.44	+5.27
	Qwen3.5-35B-A3B	85.28	86.72	+1.44
HealthBench Hard	Qwen3-4B	2.31	2.54	+0.23
	Qwen3.5-35B-A3B	20.97	21.84	+0.87
HealthBench Professional	Qwen3-4B	16.60	19.27	+2.67
	Qwen3.5-35B-A3B	36.70	40.15	+3.45
WildBench	Qwen3-4B	20.47	23.62	+3.15

Grader and training-seed consistency. Rescoring the same Qwen3-4B responses with DeepSeek-V4-Flash gives a **3.73**-point gain over Base, consistent with the **4.43**-point gain under GPT-4.1. Three SE-OPSD training seeds evaluated with DeepSeek-V4-Flash yield **83.56** $\pm$ **0.56** (mean $\pm$ sample standard deviation), with all three scores above the Base score of 80.11. Appendix C.4 reports the grader comparison and individual seed results.

4.3 Analysis and Ablations

We first examine how SEPC-guided generation transfers through training, then analyze the source and structure of the privileged context.

Distilling SEPC-guided generation. We compare two ways to train from the same self-elicited context. **SEPC-SFT** uses fixed answers generated by the frozen base model conditioned on SEPC as supervised targets. **SE-OPSD** instead supplies the SEPC-conditioned teacher's token distributions on student-generated prefixes. We match the training queries, SEPC, optimization budget, and decoding configuration. We also evaluate **test-time SEPC**: the frozen base model elicits all five views for each evaluation query and generates an answer conditioned on their concatenated context. Table 4 reports the comparison on the full HealthBench Consensus set with DeepSeek-V4-Flash.

Table 4: Transferring SEPC-guided generation into a direct-answer model. Qwen3-4B is evaluated on all 3,671 HealthBench Consensus examples with DeepSeek-V4-Flash; the SE-OPSD result is seed 42. Δ is the score gain over Base. Calls count model invocations per evaluation query.

Method	Overall	Δ	Calls/query
Base	80.11	–	1
SEPC-SFT	81.76	+1.65	1
SE-OPSD	**83.84**	**+3.73**	**1**
Base + test-time SEPC	84.42	+4.31	6

SEPC-SFT improves over Base, showing that SEPC-conditioned answers supply useful training targets. SE-OPSD achieves a further **2.08**-point improvement under the matched setting. Dense teacher supervision on the student's own trajectories therefore transfers SEPC-guided behavior more effectively than learning from fixed generated answers in this comparison.

From additional inference computation to single-call answers. Test-time SEPC improves the Base score by 4.31 points, while SE-OPSD retains 3.73 points of this gain, or **86.5%**, with a single model call. The score gap to the full test-time pipeline is 0.58 points. Across the evaluation set, SE-OPSD uses **3.106M** total inference tokens, compared with **25.811M** for five-view SEPC elicitation plus answer generation, an **8.31**$\times$ token ratio. These results show that the guidance constructed through additional inference computation can be incorporated into model parameters and reused in direct-answer generation. Appendix C.2 gives the axis-level scores and token accounting.

The remaining context and view ablations use Qwen3-4B on a **fixed 500-example subset** of HealthBench Consensus with GPT-4.1. Training queries, decoding settings, and the distillation objective are held fixed across context variants. Tables 5 and 6 use the same subset throughout.

Does the source of privileged information matter? We test whether query-specific response context provides useful teacher guidance relative to answer text or generic medical instructions. We therefore compare different teacher-side contexts while keeping the OPD objective fixed. **Reference answer** gives the teacher the reference answer. **Generic medical context** uses the same safety and communication instruction for every query, testing shared contextual guidance in the spirit of context distillation(Ye et al., 2026). **Shuffled SEPC** gives each query an SEPC sampled from another training example, controlling for the distribution and format of SEPC while removing query alignment. **Full SEPC** is our main method.

Table 5: Ablation of teacher-side privileged information on a fixed 500-example HealthBench Consensus subset. Base(No PI) is the no-training anchor.

Teacher context	Query-specific	Overall	Acc.	Aware.	Emerg.	Seek.
Base(No PI)	-	80.67	82.64	70.80	88.46	66.12
Reference answer	✓	82.34	**83.53**	71.08	89.45	66.76
Generic medical context	✗	81.57	81.91	69.54	88.31	68.44
Shuffled SEPC	✗	78.44	77.78	74.21	85.07	69.28
Full SEPC	✓	**84.78**	82.55	**74.64**	**91.78**	**70.07**

Table 5 shows that Full SEPC outperforms all alternative teacher contexts. All non-base rows use the same OPD objective and differ only in the teacher-side context. Generic medical context controls for

Table 6: SEPC guidance-component ablations on the same 500-example subset.

SEPC variant	Overall	Acc.	Comm.	Complete	Aware.	Ins.
Full SEPC	**84.78**	82.55	**89.26**	87.30	**74.64**	92.11
w/o Task Contract	83.48	81.97	87.38	85.41	73.58	91.44
w/o Grounding and State	83.12	82.32	87.44	85.95	71.77	92.48
w/o Correctness and Verification	81.10	79.24	86.94	87.55	74.22	**93.80**
w/o Audience and Presentation	83.77	**82.81**	86.72	**88.24**	73.96	89.89
w/o Coverage, Quality, and Safety	81.60	80.08	86.31	86.62	72.04	91.26

longer safety-oriented prompting, while shuffled SEPC controls for the format and distribution of SEPC without preserving query alignment. These controls are designed to separate three possible explanations. Generic medical context tests whether the improvement comes merely from giving the teacher a longer safety-oriented medical instruction. Shuffled SEPC tests whether the improvement comes from the surface format or distribution of checklist text, while removing alignment between the query and the elicited judgments. Full SEPC's advantage over the generic and shuffled controls supports the importance of aligning the elicited judgments with the query.

The results support structured, query-specific response context as a useful source of teacher-side information.

How sensitive is SE-OPSD to the view decomposition? We use leave-one-out analysis to study the contribution of localized guidance components, organized by the five-view schema in Appendix B.4. Table 6 shows that each removal reduces the overall score. The largest drop, 3.68 points, occurs when answerability and uncertainty guidance in Correctness and Verification is removed. Removing missing-information guidance in Grounding and State especially affects context awareness, while removing safety guidance in Coverage, Quality, and Safety lowers overall score and communication. These results highlight the value of explicit response boundaries and query-specific requirements within the multi-view context.

5 Conclusion and Limitations

We introduced SE-OPSD, a reference-free method that constructs privileged teacher context through the base model's own query analysis and distills its guidance into direct-answer generation. The method turns explicit judgments about response requirements into dense supervision on student trajectories. SE-OPSD improves HealthBench Consensus over reference-answer SFT and reference-conditioned OPSD, with further gains across model families, additional medical benchmarks, and general-domain WildBench. A matched comparison with SEPC-SFT demonstrates the effectiveness of the distillation recipe, while the full-set test-time comparison shows that a single-call model retains 86.5% of the gain from explicit five-view elicitation. Together, these results establish self-elicited response guidance as a practical source of supervision for open-ended QA.

The quality of self-elicited guidance depends on the base model, so incomplete or incorrect judgments can propagate through training. Our evaluation is most extensive in medical QA and includes one separately trained general-domain setting. Three-seed results establish consistency for Qwen3-4B under the additional grader; the other training comparisons use single runs, and bootstrap intervals quantify evaluation-sample uncertainty. Further work can broaden domain coverage, study adaptive elicitation budgets, and assess response quality with human experts. Clinical applications would additionally require validation of usefulness and safety in their intended setting.

Ethics Statement

This work studies language-model post-training using publicly available datasets, benchmarks, and pretrained models. It involves no human-subject experiments or collection of personal data. Improved response quality may support education and scientific work, but language models can also produce misleading or incorrect content. Downstream applications should use verification and safeguards appropriate to their intended use.

Reproducibility Statement

Section 3 describes SEPC construction and the on-policy distillation objective. Section 4.1 and Appendix B document the datasets, benchmarks, and evaluation protocols. Table 7 summarizes the main training and baseline configurations. Appendix B.4 provides the shared elicitation instruction and five-view prompt templates. Section 4.3 and Appendix C describe the ablation protocols and additional evaluations. Our code is available at https://anonymous.4open.science/r/SE-OPSD.

AI Use Statement

We used large language models to polish the manuscript and improve the quality of the writing. The trajectories used by SE-OPSD are generated by LLM models as part of the training procedure described in Section 3. The authors reviewed and revised the AI-assisted materials and take responsibility for the final content of this work.

References

Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos, Matthieu Geist, and Olivier Bachem. On-Policy Distillation of Language Models: Learning from Self-Generated Mistakes. In *International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=3zKtaqxLhW.

Rahul K Arora, Jason Wei, Rebecca Soskin Hicks, Preston Bowman, Joaquin Quiñonero-Candela, Foivos Tsimpourlas, Michael Sharman, Meghan Shah, Andrea Vallone, Alex Beutel, et al. Healthbench: Evaluating large language models towards improved human health. *arXiv preprint arXiv:2505.08775*, 2025. URL https://arxiv.org/abs/2505.08775.

Deepika Bablani, Ajay Gupta, and Wanming Chen. Rubrics as Privileged Information for Open-Ended Generation. *arXiv preprint arXiv:2608.02948*, 2026. URL https://arxiv.org/abs/2608.02948.

DeepSeek-AI. Deepseek-v4: Towards highly efficient million-token context intelligence, 2026. URL https://arxiv.org/abs/2606.19348.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025. URL https://arxiv.org/abs/2501.12948.

Yuqian Fu, Haohuan Huang, Kaiwen Jiang, Jiacai Liu, Zhuo Jiang, Yuanheng Zhu, and Dongbin Zhao. Revisiting on-policy distillation: Empirical failure modes and simple fixes. *arXiv preprint arXiv:2603.25562*, 2026. URL https://arxiv.org/abs/2603.25562.

Kanishk Gandhi, Ayush Chakravarthy, Anikait Singh, Nathan Lile, and Noah D Goodman. Cognitive behaviors that enable self-improving reasoners, or, four habits of highly effective stars. *arXiv preprint arXiv:2503.01307*, 2025. URL https://arxiv.org/abs/2503.01307.

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika,

Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. A framework for few-shot language model evaluation, 12 2023. URL https://zenodo.org/records/10256836.

Siyi Gu, Jialin Chen, Sophia Zhou, Arman Cohan, and Rex Ying. Rethinking Reward Supervision: Rubric-Conditioned Self-Distillation. *arXiv preprint arXiv:2606.19327*, 2026. URL https://arxiv.org/abs/2606.19327.

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. MiniLLM: On-Policy Distillation of Large Language Models. *arXiv preprint arXiv:2306.08543*, 2023. URL https://arxiv.org/abs/2306.08543.

Anisha Gunjal, Anthony Wang, Elaine Lau, Vaskar Nath, Yunzhong He, Bing Liu, and Sean Hendryx. Rubrics as rewards: Reinforcement learning beyond verifiable domains. *arXiv preprint arXiv:2507.17746*, 2025. URL https://arxiv.org/abs/2507.17746.

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

Yinghui He, Simran Kaur, Adithya Bhaskar, Yongjin Yang, Jiarui Liu, Narutatsu Ri, Liam Fowl, Abhishek Panigrahi, Danqi Chen, and Sanjeev Arora. Self-distillation zero: Self-revision turns binary rewards into dense supervision. *arXiv preprint arXiv:2604.12002*, 2026. URL https://arxiv.org/abs/2604.12002.

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

Rebecca Soskin Hicks, Mikhail Trofimov, Dominick Lim, Rahul K Arora, Foivos Tsimpourlas, Preston Bowman, Michael Sharman, Chi Tong, Kavin Karthik, Arnav Dugar, et al. Healthbench professional: Evaluating large language models on real clinician chats. *arXiv preprint arXiv:2604.27470*, 2026. URL https://arxiv.org/abs/2604.27470.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015. URL https://arxiv.org/abs/1503.02531.

Jiaxin Huang, Shixiang Gu, Le Hou, Yuexin Wu, Xuezhi Wang, Hongkun Yu, and Jiawei Han. Large language models can self-improve. In *Proceedings of the 2023 conference on empirical methods in natural language processing*, pp. 1051–1068, 2023a. URL https://aclanthology.org/2023.emnlp-main.67/.

Jie Huang, Xinyun Chen, Swaroop Mishra, Huaixiu Steven Zheng, Adams Wei Yu, Xinying Song, and Denny Zhou. Large language models cannot self-correct reasoning yet. *arXiv preprint arXiv:2310.01798*, 2023b. URL https://arxiv.org/abs/2310.01798.

Jonas Hübotter, Frederike Lübeck, Lejs Behric, Anton Baumann, Marco Bagatella, Daniel Marta, Ido Hakimi, Idan Shenfeld, Thomas Kleine Buening, Carlos Guestrin, et al. Reinforcement learning via self-distillation. *arXiv preprint arXiv:2601.20802*, 2026. URL https://arxiv.org/abs/2601.20802.

Ijun Jang, Jewon Yeom, Juan Yeo, Hyunggyu Lim, and Taesup Kim. Stable on-policy distillation through adaptive target reformulation. *arXiv preprint arXiv:2601.07155*, 2026. URL https://arxiv.org/abs/2601.07155.

Dulhan Jayalath, Shashwat Goel, Thomas Foster, Parag Jain, Suchin Gururangan, Cheng Zhang, Anirudh Goyal, and Alan Schelten. Compute as Teacher: Turning Inference Compute Into Reference-Free Supervision. *arXiv preprint arXiv:2509.14234*, 2025. URL https://arxiv.org/abs/2509.14234.

Di Jin, Eileen Pan, Nassim Oufattole, Wei-Hung Weng, Hanyi Fang, and Peter Szolovits. What disease does this patient have? a large-scale open domain question answering dataset from medical exams. *Applied Sciences*, 11(14):6421, 2021. URL https://doi.org/10.3390/app11146421.

Woogyeol Jin, Taywon Min, Yongjin Yang, Dennis Wei, Yi Zhou, Swanand Ravindra Kadhe, Nathalie Baracaldo, and Kimin Lee. Entropy-aware on-policy distillation of language models. *arXiv preprint arXiv:2603.07079*, 2026. URL https://arxiv.org/abs/2603.07079.

Ryo Kamoi, Yusen Zhang, Nan Zhang, Jiawei Han, and Rui Zhang. When can llms actually correct their own mistakes? a critical survey of self-correction of llms. *Transactions of the Association for Computational Linguistics*, 12:1417–1440, 2024. URL https://aclanthology.org/2024.tacl-1.78/.

Yoon Kim and Alexander M Rush. Sequence-level knowledge distillation. In *Proceedings of the 2016 conference on empirical methods in natural language processing*, pp. 1317–1327, 2016. URL https://aclanthology.org/D16-1139/.

Jongwoo Ko, Sara Abdali, Young Jin Kim, Tianyi Chen, and Pashmina Cameron. Scaling reasoning efficiently via relaxed on-policy distillation. *arXiv preprint arXiv:2603.11137*, 2026. URL https://arxiv.org/abs/2603.11137.

Aviral Kumar, Vincent Zhuang, Rishabh Agarwal, Yi Su, John D Co-Reyes, Avi Singh, Kate Baumli, Shariq Iqbal, Colton Bishop, Rebecca Roelofs, et al. Training language models to self-correct via reinforcement learning. *arXiv preprint arXiv:2409.12917*, 2024. URL https://arxiv.org/abs/2409.12917.

Yunghwei Lai, Kaiming Liu, Ziyue Wang, Weizhi Ma, and Yang Liu. Doctor-r1: Mastering clinical inquiry with experiential agentic reinforcement learning. *arXiv preprint arXiv:2510.04284*, 2025. URL https://arxiv.org/abs/2510.04284.

Yaxuan Li, Yuxin Zuo, Bingxiang He, Jinqian Zhang, Chaojun Xiao, Cheng Qian, Tianyu Yu, Huanang Gao, Wenkai Yang, Zhiyuan Liu, et al. Rethinking on-policy distillation of large language models: Phenomenology, mechanism, and recipe. *arXiv preprint arXiv:2604.13016*, 2026. URL https://arxiv.org/abs/2604.13016.

Bill Yuchen Lin, Yuntian Deng, Khyathi Chandu, Faeze Brahman, Abhilasha Ravichander, Valentina Pyatkin, Nouha Dziri, Ronan Le Bras, and Yejin Choi. WildBench: Benchmarking LLMs with challenging tasks from real users in the wild. *arXiv preprint arXiv:2406.04770*, 2024. URL https://arxiv.org/abs/2406.04770.

LLM-Core Xiaomi. MiMo-V2-Flash Technical Report. *arXiv preprint arXiv:2601.02780*, 2026. URL https://arxiv.org/abs/2601.02780.

Kevin Lu and Thinking Machines Lab. On-policy distillation. *Thinking Machines Lab: Connectionism*, 2025. doi: 10.64434/tml.20251026. https://thinkingmachines.ai/blog/on-policy-distillation.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. Self-refine: Iterative refinement with self-feedback. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 46534–46594. Curran Associates, Inc., 2023. doi: 10.52202/075280-2019. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/91edff07232fb1b55a505a9e9f6c0ff3-Paper-Conference.pdf.

Itay Manes, Naama Ronn, David Cohen, Ran Ilan Ber, Zehavi Horowitz-Kugler, and Gabriel Stanovsky. K-QA: A real-world medical Q&A benchmark. In Dina Demner-Fushman, Sophia Ananiadou, Makoto Miwa, Kirk Roberts, and Junichi Tsujii (eds.), *Proceedings of the 23rd Workshop on Biomedical Natural Language Processing*, pp. 277–294, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.bionlp-1.22. URL https://aclanthology.org/2024.bionlp-1.22/.

Emiliano Penaloza, Dheeraj Vattikonda, Nicolas Gontier, Alexandre Lacoste, Laurent Charlin, and Massimo Caccia. Privileged information distillation for language models. *arXiv preprint arXiv:2602.04942*, 2026. URL https://arxiv.org/abs/2602.04942.

MohammadHossein Rezaei, Anas Mahmoud, Zihao Wang, Utkarsh Tyagi, Advait Gosai, Razvan-Gabriel Dumitru, Aakash Sabharwal, Bing Liu, and Yunzhong He. Rubric-Guided Self-Distillation: Post-Training Without Rubric Verifiers. *arXiv preprint arXiv:2606.12507*, 2026. URL https://arxiv.org/abs/2606.12507.

Hejian Sang, Yuanda Xu, Zhengze Zhou, Ran He, Zhipeng Wang, and Jiachen Sun. Crisp: Compressed reasoning via iterative self-policy distillation. *arXiv preprint arXiv:2603.05433*, 2026. URL https://arxiv.org/abs/2603.05433.

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

Idan Shenfeld, Mehul Damani, Jonas Hübotter, and Pulkit Agrawal. Self-distillation enables continual learning. *arXiv preprint arXiv:2601.19897*, 2026. URL https://arxiv.org/abs/2601.19897.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. *Advances in neural information processing systems*, 36:8634–8652, 2023. URL https://papers.nips.cc/paper_files/paper/2023/hash/1b44b878bb782e6954cd888628510e90-Abstract-Conference.html.

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

Mingyang Song and Mao Zheng. A survey of on-policy distillation for large language models. *arXiv preprint arXiv:2604.00626*, 2026. URL https://arxiv.org/abs/2604.00626.

Alex Stein, Furong Huang, and Tom Goldstein. Gates: Self-distillation under privileged context with consensus gating. *arXiv preprint arXiv:2602.20574*, 2026. URL https://arxiv.org/abs/2602.20574.

Yuanda Xu, Hejian Sang, Zhengze Zhou, Ran He, and Zhipeng Wang. Paced: Distillation and on-policy self-distillation at the frontier of student competence. *arXiv preprint arXiv:2603.11178*, 2026. URL https://arxiv.org/abs/2603.11178.

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

Chenxu Yang, Chuanyu Qin, Qingyi Si, Minghui Chen, Naibin Gu, Dingyu Yao, Zheng Lin, Weiping Wang, Jiaqi Wang, and Nan Duan. Self-distilled rlvr. *arXiv preprint arXiv:2604.03128*, 2026. URL https://arxiv.org/abs/2604.03128.

Tianzhu Ye, Li Dong, Zewen Chi, Xun Wu, Shaohan Huang, and Furu Wei. Black-box on-policy distillation of large language models. *arXiv preprint arXiv:2511.10643*, 2025. URL https://arxiv.org/abs/2511.10643.

Tianzhu Ye, Li Dong, Xun Wu, Shaohan Huang, and Furu Wei. On-policy context distillation for language models. *arXiv preprint arXiv:2602.12275*, 2026. URL https://arxiv.org/abs/2602.12275.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, et al. Dapo: An open-source llm reinforcement learning system at scale. *arXiv preprint arXiv:2503.14476*, 2025. URL https://arxiv.org/abs/2503.14476.

Weizhe Yuan, Richard Yuanzhe Pang, Kyunghyun Cho, Xian Li, Sainbayar Sukhbaatar, Jing Xu, and Jason Weston. Self-rewarding language models. *arXiv preprint arXiv:2401.10020*, 2024. URL https://arxiv.org/abs/2401.10020.

Zheng Yuan, Hongyi Yuan, Chengpeng Li, Guanting Dong, Keming Lu, Chuanqi Tan, Chang Zhou, and Jingren Zhou. Scaling relationship on learning mathematical reasoning with large language models. *arXiv preprint arXiv:2308.01825*, 2023. URL https://arxiv.org/abs/2308.01825.

Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah Goodman. Star: Bootstrapping reasoning with reasoning. *Advances in Neural Information Processing Systems*, 35:15476–15488, 2022. URL https://papers.nips.cc/paper/2022/hash/639a9a172c044fbb64175b5fad42e9a5-Abstract-Conference.html.

Dongxu Zhang, Zhichao Yang, Sepehr Janghorbani, Jun Han, Andrew Ressler II, Qian Qian, Gregory D Lyng, Sanjit Singh Batra, and Robert E Tillman. Fast and effective on-policy distillation from reasoning prefixes. *arXiv preprint arXiv:2602.15260*, 2026a. URL https://arxiv.org/abs/2602.15260.

XiuYu Zhang, Wei Chow, Junfeng Fang, Xingyu Zhu, Zhenkai Liang, and Tat-Seng Chua. What does privileged information add to on-policy self-distillation? *arXiv preprint arXiv:2609.20612*, 2026b. URL https://arxiv.org/abs/2609.20612.

Siyan Zhao, Zhihui Xie, Mengchen Liu, Jing Huang, Guan Pang, Feiyu Chen, and Aditya Grover. Self-distilled reasoner: On-policy self-distillation for large language models. *arXiv preprint arXiv:2601.18734*, 2026. URL https://arxiv.org/abs/2601.18734.

A Additional Related Work

Recent developments in on-policy distillation. Recent and concurrent work has broadened the design space of on-policy distillation (OPD) for large language models. Several works survey or analyze the emerging OPD literature, highlighting factors such as teacher–student compatibility, token-level signal quality, sampled-prefix reliability, and implementation details that affect whether OPD succeeds (Song & Zheng, 2026; Li et al., 2026; Fu et al., 2026). Other methods modify the OPD objective, supervision granularity, or training schedule to improve stability and efficiency, including relaxed likelihood-ratio objectives, intermediate or entropy-aware target distributions, prefix-level supervision, difficulty-paced training, and reasoning-compression distillation (Ko et al., 2026; Jang et al., 2026; Jin et al., 2026; Zhang et al., 2026a; Xu et al., 2026; Sang et al., 2026). A parallel line combines OPD-style token supervision with RLVR, revision-based self-training, or large-scale industrial post-training pipelines (Yang et al., 2026; He et al., 2026; LLM-Core Xiaomi, 2026; Yang et al., 2025; DeepSeek-AI, 2026). These works are complementary to ours: they primarily study how to make on-policy distillation more stable, efficient, or scalable, often in verifiable reasoning or strong-to-weak distillation settings. SE-OPSD instead keeps the standard token-level OPD objective fixed and studies a different design axis: how to construct query-specific teacher guidance from the base model itself for open-ended QA when reference answers are unavailable.

Self-critique, self-refinement, and self-generated feedback. A broad line of work shows that LLMs can improve their outputs by generating feedback, critiques, reflections, or revisions(Singh et al., 2023; Yuan et al., 2023; 2024; Huang et al., 2023a). STaR improves reasoning through iterative generation and fine-tuning (Zelikman et al., 2022), while Self-Refine uses repeated feedback and revision at inference time (Madaan et al., 2023). Agentic reflection methods similarly convert task feedback into verbal reflections that are stored or reused as additional context in later attempts(Shinn et al., 2023). These methods support the premise that models may possess useful judgment abilities that are not fully expressed in their first response. Prompting-based self-refinement methods elicit such behavior through critiques, reflections, or revisions at inference time (Madaan et al., 2023), but often require additional model calls and provide limited gains without external feedback or weight updates (Huang et al., 2023b; Kamoi et al., 2024). Training-based variants use supervised learning or reinforcement learning to internalize correction behavior (Kumar et al., 2024; Havrilla et al., 2024; Gandhi et al., 2025), but typically depend on revision traces, feedback signals, or multi-turn inference. In contrast, SE-OPSD elicits response-framing context only during training and distills it into a single-pass direct-answer model.

B Experimental Details

B.1 Training data

RaR-Medicine is the medical-domain dataset released with *Rubrics as Rewards*; each example contains a medical question, a reference answer, the source of the question, and checklist-style rubric annotations generated from the reference answer. The released dataset contains 17,926 training, 2,240 validation, and 2,242 test examples. We use only the question field for SE-OPSD, while reference answers are reserved for reference-answer baselines.

K-QA is a real-world medical QA benchmark collected from patient inquiries on the K Health platform. It contains authentic patient-written medical questions and an expert-annotated answer subset with fine-grained ground-truth statements. Compared with RaR-Medicine, K-QA is more patient-facing and reflects natural consumer health questions rather than exam-style or synthetic medical reasoning prompts. This mixture gives us both broad medical reasoning coverage and realistic patient-facing language.

The two datasets play complementary roles. RaR-Medicine provides broad coverage of medical reasoning questions with reference answers and structured rubrics, which is useful for constructing reference-answer baselines. K-QA provides authentic patient-facing questions and expert-grounded answer statements, making it closer to the open-ended consumer-health setting targeted by Health-Bench.

All datasets and model checkpoints used in this work are publicly released research assets, and we use them in accordance with their respective licenses, terms of use, and citation requirements. In particular, Qwen3 model checkpoints are released under Apache-2.0, HealthBench and K-QA are released under MIT licenses, and RaR-Medicine is a publicly released research dataset associated with Rubrics as Rewards; we use only the question field from RaR-Medicine for SE-OPSD and reserve its reference answers and rubrics only for constructing the reference-answer baselines.

General-domain training data. For WildBench, we separately train Qwen3-4B on the query portion of WildChat-IF, using the released `allenai/tulu-3-wildchat-if-on-policy-8b` dataset.[1] The released asset contains user prompts with instruction constraints and paired model completions; SE-OPSD uses the queries to elicit its own teacher context through the domain-neutral five-view construction illustrated in Appendix B.4. Exact and semantic overlaps with the WildBench evaluation queries are removed from the training data.

B.2 Evaluation Details

HealthBench. We evaluate HealthBench using the official rubric-based evaluation pipeline. Each example consists of a health-related conversation, and the evaluated model produces an answer to the final user message. The answer is then scored against conversation-specific physician-written rubric items using the recommended GPT-4.1 model-based grader for the primary Qwen3 comparisons. Additional model, benchmark, SEPC-SFT, and full-set test-time comparisons use DeepSeek-V4-Flash, as specified in the corresponding tables. For each rubric item, the grader predicts whether the model response satisfies the criterion. The example score is computed from the weighted fraction of satisfied criteria, and aggregate scores are reported over examples. We report the overall HealthBench Consensus score and axis-level breakdown in Table 2, with theme-level results in Appendix B.5.

Within each HealthBench comparison, methods use the same split, answer-generation settings, grader, and rubric set; the test-time SEPC variant additionally receives its self-elicited context. For grader sensitivity, we keep the cached Qwen3-4B answers fixed and change only the grading model. The rubrics and grader outputs are used only for evaluation and are never used for training, SEPC elicitation, model selection, or inference-time prompting.

Evaluation protocol. The trained models use direct-answer inference from the query alone. The test-time SEPC comparator makes five elicitation calls and one answer-generation call. HealthBench rubrics and grader outputs are confined to evaluation. The medical checkpoints are also evaluated on HealthBench Hard and HealthBench Professional with DeepSeek-V4-Flash. Cross-model comparisons match the training queries, token and optimization budgets, and evaluation protocol, with each model generating its own SEPC.

For MedQA and MMLU, we use the standard 5-shot likelihood-ranking protocol in the EleutherAI lm-evaluation-harness (Gao et al., 2023). The model scores the candidate option labels conditioned on the few-shot prompt, and the highest-likelihood option is selected. This protocol computes multiple-choice accuracy without generating free-form answers or chain-of-thought rationales. We apply the same protocol to the base and trained checkpoints in the knowledge-retention columns of Table 2.

WildBench. WildBench comprises 1,024 challenging open-ended requests drawn from real user conversations (Lin et al., 2024). We evaluate the base Qwen3-4B model and the separate SE-OPSD checkpoint trained on WildChat-IF queries described above. The trained model answers each evaluation query directly, without SEPC elicitation at evaluation time. Table 3 reports scores of 20.47 for Base and 23.62 for SE-OPSD, a gain of 3.15 points on WildBench's own scale. This comparison evaluates the same elicitation and distillation procedure with domain-neutral response guidance on general-domain requests.

B.3 Training configuration.

We provide the training configurations for Ref-answer SFT, Ref-OPSD, and SE-OPSD in Table 7. All methods are trained on the same set of medical queries. Ref-answer SFT uses reference answers

[1] WildChat-IF dataset release on Hugging Face.

Table 7: Training configuration for Ref-answer SFT, Ref-OPSD, and SE-OPSD. Ref-answer SFT performs completion-only supervised fine-tuning on reference answers. Ref-OPSD and SE-OPSD use the same on-policy distillation objective and rollout budget; the only difference is the teacher-side privileged information.

Parameter	**Ref-answer SFT**	**Ref-answer OPSD**	**SE-OPSD**
Learning Rate	5×10^{-6}	5×10^{-6}	5×10^{-6}
Effective Batch Size	32	32	32
Per Device Train Batch Size	4	4	4
Number of GPUs	8	8	8
LoRA Rank (r)	64	64	64
LoRA Alpha (α)	128	128	128
LoRA Target Modules	`q_proj`, `k_proj`, `v_proj`, `o_proj`, `gate_proj`, `up_proj`, `down_proj`		
Max Sequence Length	20000	–	–
Max Completion Length	–	1024	1024
Number of Generations per Prompt	–	1	1
Sampling Temperature	–	1.0	1.0
Teacher Context	–	Reference answer	SEPC
Teacher Update	–	fixed	fixed
Training Steps	500	100	100

as supervised targets. Ref-OPSD and SE-OPSD both use one on-policy student rollout per query and optimize the same token-level distillation objective; they differ only in the privileged information given to the teacher. Ref-OPSD conditions the teacher on the reference answer, while SE-OPSD conditions the teacher on self-elicited privileged context generated from the query alone.

The primary Qwen3 training runs use 8 A800 GPUs with gradient checkpointing and Flash Attention 2 for memory efficiency. These runs use the AdamW optimizer and bfloat16 precision. For Ref-OPSD and SE-OPSD, unless otherwise stated, we use full-vocabulary logit distillation with forward KL, and gradients are applied only to the student logits. For SE-OPSD, SEPC is generated from the query alone. The teacher is frozen throughout these experiments, allowing SEPC to be cached before training. The student answers from the query alone at inference.

SEPC-conditioned SFT control. The additional SEPC-SFT control uses fixed responses generated by the frozen base model conditioned on the same SEPC as SE-OPSD. It matches the training queries, SEPC, optimization budget, and decoding configuration of the paired SE-OPSD run. The comparison changes the supervision recipe from fixed generated-answer tokens to teacher token distributions evaluated on student-sampled prefixes. This control is separate from the reference-answer SFT baseline in Table 7.

B.4 SEPC Elicitation Prompt

We present compact, domain-neutral templates for the shared elicitation instruction and five views, with field descriptions abbreviated for readability. The views are elicited in five independent calls, each combining the same shared instruction with a view-specific task and the user's query or conversation.

SEPC Elicitation: Shared Instruction

Read the user's query or conversation and identify the response requirements for the latest requested task. Fill in the JSON fields for the requested view using the provided inputs and relevant conversation state. Return JSON only.

Input supplied to each call:

```
{query}
```

General rules:

- Use only the user's query or conversation.
- Describe the requested operation and its requirements; do not carry out the task or include a candidate answer.
- State answer-independent checks without performing them or importing external facts.
- Do not copy field descriptions or allowed-value instructions into JSON values.
- Preserve explicit inputs, prior-turn corrections, and task constraints.
- Include clarification needs, assumptions, and safety boundaries only when justified by the query.
- Keep list items concise. Use `[]` for inapplicable list fields and `unspecified` for unstated audience or presentation attributes.
- Use strings for single-value fields and arrays of strings for list fields.
- Return exactly one valid JSON object with the requested `view` identifier and its fields.

SEPC View 1: Task Contract

Task focus: Identify the latest requested task, deliverable, necessary clarification, success conditions, and any conflicting instructions.

Fields to fill:

- `task_type`, `deliverable`, `response_action`: identify the task category, requested output type, and operation, such as explaining, transforming, planning, or debugging.
- `need_context`: one of `no`, `partial`, or `yes`; `clarification_questions`: up to two blocking or materially useful questions, normally `[]`.
- `success_conditions`: up to four explicit or logically necessary completion conditions.
- `constraint_conflicts`: up to two incompatible requirement pairs; `resolution_strategy`: `not_applicable` when none exists, otherwise a brief strategy for resolving the conflict.

View identifier: `task_contract`.

SEPC View 2: Grounding and State

Task focus: Capture salient explicit inputs and prior-turn state needed for the latest task, plus indispensable assumptions and response-changing omissions.

Fields to fill:

- `explicit_inputs`: up to four salient provided artifacts, values, facts, or instructions.
- `conversation_state`: up to three relevant prior-turn commitments, corrections, failures, or preferences; `[]` for a single-turn query.
- `source_fidelity`: `not_applicable`, `preserve_exactly`, `preserve_meaning`, `extend_consistently`, or `use_as_evidence`.
- `assumptions`, `missing_information`: up to two indispensable assumptions and two omissions that materially affect the response; normally empty.

View identifier: `grounding_and_state`.

SEPC View 3: Correctness and Verification

Task focus: State compact, answer-independent checks, evidence needs, and claim boundaries. Do not perform the checks or include candidate answers.

Fields to fill:

- `correctness_modes`: relevant kinds of correctness, such as factual, mathematical, executable, source-faithful, or constraint satisfaction.
- `verification_targets`: up to four answer-independent checks that the final response should pass.
- `evidence_needs`, `edge_cases`: up to two necessary source, recency, test, or calculation needs and two query-specific boundary conditions.
- `uncertainty`: `none`, `reducible`, `irreducible`, or `mixed`; `claim_limits`: up to two necessary boundaries on claims, otherwise `[]`.

View identifier: `correctness_and_verification`.

SEPC View 4: Audience and Presentation

Task focus: Capture explicit audience and presentation constraints without inventing a persona, expertise level, tone, or structure.

Fields to fill:

- `audience`: the explicit intended audience or `unspecified`; `expertise`: `novice`, `mixed`, `expert`, or `unspecified`.
- `language`, `format`, `length`: the requested language, output format, and length constraint; default to `match_user`, `free_form`, and `unspecified`.
- `tone`: up to two requested or necessary attributes; `structure`: up to four required sections or ordering constraints.
- `instruction_constraints`: up to four remaining explicit presentation constraints.

View identifier: `audience_and_presentation`.

SEPC View 5: Coverage, Quality, and Safety

Task focus: Identify required coverage, task-specific quality, actionability, visible safety boundaries, and important avoidances without duplicating other views.

Fields to fill:

- `must_cover`: up to five required subtasks or content areas; `quality_requirements`: up to three query-specific qualities.
- `actionability`: `none`, `optional`, or `required`; use `required` for requested advice, plans, steps, recommendations, code, debugging, or implementable procedures.
- `safety_or_ethics`: up to three concrete boundaries justified by a visible risk in the query; normally `[]`.
- `must_avoid`: up to three task-specific mistakes or explicit prohibitions, normally zero to two.
- `priority_order`: up to three priorities when requirements genuinely compete; otherwise `[]`.

View identifier: `coverage_quality_safety`.

B.5 HealthBench Theme-Level Results

Table 8 complements the HealthBench overall and axis-level results in Table 2 with the theme-level breakdown on the same HealthBench Consensus evaluation set and GPT-4.1 grader. The methods and model checkpoints are unchanged; the Overall column repeats the aggregate score for ease of comparison.

Table 8: Theme-level HealthBench Consensus results with GPT-4.1. Overall repeats the aggregate score from Table 2. Scores are percentages; Δ denotes SE-OPSD minus the corresponding Base model. Bold values mark the best method within each model size and metric.

Model	Method	Overall	Themes				
			Emerg.	Hedging	Seek.	Complex	HealthData
Qwen3-1.7B	Base	67.86	81.01	74.12	62.62	49.53	75.95
	SFT	69.26	78.25	74.36	67.03	**51.70**	79.36
	OPSD	69.34	80.13	75.05	63.72	43.03	77.34
	SE-OPSD	**72.88**	**84.87**	**75.66**	**71.20**	50.77	**80.50**
	Δ Improve	**+5.02**	**+3.86**	**+1.54**	**+8.58**	**+1.24**	**+4.55**
Qwen3-4B	Base	82.15	89.95	81.90	70.95	64.50	87.84
	SFT	82.64	90.06	81.90	70.22	66.82	88.73
	OPSD	82.95	91.72	81.34	70.22	66.20	85.56
	SE-OPSD	**86.58**	**93.56**	**83.14**	**73.85**	**70.57**	**91.83**
	Δ Improve	**+4.43**	**+3.61**	**+1.24**	**+2.90**	**+6.07**	**+3.99**

SE-OPSD improves over Base across all five themes for both model sizes. For Qwen3-1.7B/Qwen3-4B, the improvements include **+3.86/+3.61** in emergency referral and **+1.54/+1.24** in hedging. Together with the context-awareness and completeness gains in the main text, these results show that self-elicited guidance improves how models handle uncertainty and safety-related response requirements.

B.6 Bootstrap Confidence Intervals for Main HealthBench Results

HealthBench reports aggregate scores together with bootstrap uncertainty estimates over evaluation examples. We use this evaluation bootstrap to quantify uncertainty in the overall HealthBench Consensus score. For each model and method, let s_i denote the normalized score of example i and let N be the number of evaluation examples. The reported point estimate is

$$\hat{\mu} = 100 \cdot \frac{1}{N} \sum_{i=1}^{N} s_i.$$

For bootstrap resample b, we sample N examples with replacement from the evaluation set and compute

$$\hat{\mu}^{(b)} = 100 \cdot \frac{1}{N} \sum_{i \in \mathcal{B}_b} s_i.$$

The bootstrap standard deviation is the empirical standard deviation of $\{\hat{\mu}^{(b)}\}_{b=1}^{B}$. We report 95% confidence intervals using the 2.5th and 97.5th percentiles of the bootstrap distribution. These confidence intervals measure evaluation-set uncertainty and do not capture training-seed variance, since each method is trained once in our main experiments.

B.6.1 Paired Improvements and Response-Length Diagnostics

Because all methods are evaluated on the same HealthBench examples, we compute paired per-example differences between SE-OPSD and the reference-based baselines. We bootstrap examples to obtain 95% confidence intervals for the mean paired gain. We also analyze response length to test whether the gains are explained by verbosity.

Table 9: Bootstrap confidence intervals for HealthBench Consensus overall scores. Scores are reported on the same percentage scale as Table 2. Intervals are 95% bootstrap confidence intervals over evaluation examples.

Model	Method	Overall score with 95% bootstrap CI
Qwen3-1.7B	Base	67.86 [66.64, 69.10]
	Ref-answer SFT	69.26 [68.09, 70.43]
	Ref-answer OPSD	69.34 [68.11, 70.58]
	SE-OPSD	**72.88 [71.82, 73.95]**
Qwen3-4B	Base	82.15 [81.18, 83.13]
	Ref-answer SFT	82.64 [81.68, 83.60]
	Ref-answer OPSD	82.95 [82.04, 83.87]
	SE-OPSD	**86.58 [85.90, 87.26]**

Table 10: Paired and length diagnostics on Qwen3-1.7B HealthBench Consensus. Paired intervals are 95% bootstrap CIs over examples. Win/tie/loss uses a 1.0 point tie threshold.

Comparison	Δ score	95% CI	Win/Tie/Loss	Δ tokens	$\rho(\Delta\ell, \Delta s)$
SE-OPSD vs Ref-answer SFT	3.62	[2.57, 4.67]	698/2491/482	+22.3	-0.085
SE-OPSD vs Ref-answer OPSD	3.54	[2.47, 4.59]	717/2462/492	-417.5	-0.058

Table 10 shows positive SE-OPSD gains under paired resampling with only a small length increase over SFT and a substantial reduction from OPSD. Relative to Ref-answer SFT, SE-OPSD gains 3.62 points while adding only 22 tokens on average. Relative to Ref-answer OPSD, SE-OPSD gains 3.54 points while being much shorter. The correlation between length increase and score improvement is weakly negative in both comparisons, so the examples where SE-OPSD writes more are not the examples where it gains more.

For the paired length-gain diagnostic, we define $\Delta\ell_i = \ell_i^{\text{SE-OPSD}} - \ell_i^{\text{baseline}}$ and $\Delta s_i = s_i^{\text{SE-OPSD}} - s_i^{\text{baseline}}$. The last column of Table 10 reports their Spearman correlation.

Table 11 further reports the distribution of response lengths and the Spearman correlation between length and per-example HealthBench score within each method. These within-method correlations are also weakly negative.

Table 11: Response-length diagnostics on Qwen3-1.7B HealthBench Consensus. Token lengths are computed consistently across methods. Spearman ρ measures the correlation between per-example output length and HealthBench score within each method.

Method	Score	Mean tokens	Median	P95	Spearman ρ
Base	67.86	656.6	590	1303.5	-0.229
Ref-answer SFT	69.26	393.5	337	795.0	-0.225
Ref-answer OPSD	69.34	833.3	718	1812.5	-0.236
SE-OPSD	72.88	415.8	321	949.5	-0.259

This diagnostic also reveals a failure mode of reference-conditioned OPSD: on Qwen3-1.7B, Ref-answer OPSD produces substantially longer outputs than all other methods, often with excessive structure, repeated phrases, or failure to stop. Appendix E provides over-generation statistics and examples.

C Additional SEPC Ablations

C.1 Why decompose SEPC into multiple local judgments?

SEPC is elicited through several short views rather than a single long prompt because we treat response-context construction as a decomposition problem. A single prompt asks the model to jointly

decide response mode, missing context, uncertainty, safety, audience, and coverage, which can be brittle for small models. Multi-view elicitation turns this into several smaller judgments, each with a narrow function and a constrained output format.

To quantify this effect, we compare single-view and multi-view SEPC elicitation on a fixed set of 500 medical queries. The single-view setting uses one prompt that asks the model to produce all SEPC fields in a single JSON object. The multi-view setting uses five independent view-specific calls, and the resulting JSON objects are concatenated to form the final SEPC. Appendix B.4 illustrates this construction with compact domain-neutral templates. We evaluate elicitation quality for both Qwen3-1.7B and Qwen3-4B.

We report two automatic metrics. JSON valid measures whether the raw output can be parsed as valid JSON. Schema valid additionally requires the output to contain the required keys, use the correct field types, and respect the allowed enum values. For multi-view elicitation, an example is counted as valid only if all five view outputs pass the corresponding check.

Table 12: Format stability of single-view and multi-view SEPC elicitation on 500 medical queries.

Model	Elicitation	JSON valid (%)	Schema valid (%)
Qwen3-1.7B	Single-view	61.4	6.8
Qwen3-1.7B	Multi-view	94.8	90.0
Qwen3-4B	Single-view	96.8	83.0
Qwen3-4B	Multi-view	99.6	98.0

Table 12 shows that multi-view elicitation improves both JSON validity and schema adherence for SEPC generation. The effect is especially important for the smaller Qwen3-1.7B model, where the single-view prompt more frequently produces invalid JSON or violates the required schema. Multi-view elicitation reduces the burden on each generation by isolating one response-context judgment at a time. This supports our use of multi-view SEPC as a reliability-oriented design choice rather than only as a downstream performance optimization.

The view decomposition also makes SEPC easier to audit: each view has a narrow intended function, so failures can be localized to a specific aspect such as task identification, source fidelity, or audience requirements. This separation helps preserve the requested operation even when the source material evokes a different response pattern.

Case study: task misclassification under single-view elicitation. Table 12 measures aggregate stability, but the failure mode is also visible qualitatively. In the example below, the user asks the model to revise a clinician-facing sentence about suspected opioid overdose. The requested operation is text transformation: the response should improve the wording while preserving the supplied meaning. The failure arises when the model lets the subject matter of the source text override the user's instruction to revise it. Separating the task contract from grounding, verification, presentation, and coverage makes the relevant requirements explicit: sufficient input for an edit, source fidelity, and no additional advice.

Case Study: Confusing Source Content with the Requested Task

User query.

> *revise Dr. Anderson, I agree with you, and you have touched on all the critical steps for suspected opioid overdose, including administering naloxone IV, repeating it if there is no improvement, ensuring airway patency, and monitoring for sedation reemergence. The salient feature here is the potential for recurrent respiratory depression.*

Intended task. The query is a **transformation** request. The model should revise the provided text while preserving its meaning and intended recipient.

Single-view failure pattern

A single long elicitation output misclassified the request as a direct medical answer and introduced an unnecessary context requirement:

- It labeled the response mode as `direct_medical_answer`.
- It set `need_context: yes`.
- It mixed response-context planning with clinical advice.

Illustrative Task Analysis under the Five-view Schema

The following illustrative excerpts show the relevant fields for this revision request under the domain-neutral templates in Appendix B.4.

View 1: task_contract

```
{
  "view": "task_contract",
  "task_type": "transformation",
  "deliverable": "A revised version of the supplied text.",
  "response_action": "transform",
  "need_context": "no"
}
```

View 2: grounding_and_state

```
{
  "view": "grounding_and_state",
  "explicit_inputs": [
    "The source text is addressed to Dr. Anderson.",
    "Topic: overdose care and recurrent respiratory depression."
  ],
  "source_fidelity": "preserve_meaning",
  "missing_information": []
}
```

View 3: correctness_and_verification

```
{
  "view": "correctness_and_verification",
  "correctness_modes": ["source_faithful"],
  "verification_targets": [
    "Preserve the source meaning and degree of certainty.",
    "Retain the named interventions and monitoring details."
  ],
  "uncertainty": "none",
  "claim_limits": ["Do not add recommendations beyond the source."]
}
```

View 4: audience_and_presentation

```
{
  "view": "audience_and_presentation",
  "audience": "Dr. Anderson",
  "language": "match_user",
  "format": "free_form"
}
```

View 5: coverage_quality_safety

```
{
  "view": "coverage_quality_safety",
  "must_cover": [
    "Revise the source, retaining all substantive points."
  ],
```

```
  "quality_requirements": [
    "Improve sentence flow while retaining the original emphasis."
  ],
  "actionability": "none",
  "safety_or_ethics": [],
  "must_avoid": [
    "Do not replace the edit with advice about the source topic.",
    "Do not request context that is unnecessary for the revision."
  ]
}
```

C.2 Does Test-Time SEPC Prompting Explain the Gains?

SEPC can guide answer generation either explicitly at inference or through the trained student's parameters. The test-time comparator uses the frozen base model to elicit five views from the evaluation query, then conditions its final answer on their concatenation. Table 13 gives the original 500-example subset comparison with GPT-4.1. Table 4 in the main text reports the full-set comparison with DeepSeek-V4-Flash.

Table 13: Diagnostic comparison with test-time SEPC prompting on Qwen3-4B. Base and SE-OPSD use single-pass direct-answer inference, while Base + SEPC-at-inference first elicits SEPC from the query and then conditions on both the query and SEPC to generate the final answer. Scores use the same fixed 500-example HealthBench Consensus subset as the context ablations, with GPT-4.1 grading.

Method	Overall	Acc.	Comm.	Instr.	Aware.	Complete
Base	80.67	82.64	86.77	89.94	70.80	81.60
SE-OPSD	84.78	82.55	89.26	92.11	74.64	87.30
Base + SEPC-at-inference	85.50	86.98	90.52	88.78	73.33	91.18

Both comparisons show that explicit SEPC improves the base model and that distillation transfers much of this benefit to direct-answer generation. On the 500-example subset, SE-OPSD scores 84.78 and test-time SEPC scores 85.50, compared with 80.67 for Base.

Full-set axis-level comparison. Table 14 reports the five axes for the full 3,671-example comparison. Relative to Base, SE-OPSD improves context awareness by 5.69 points and completeness by 7.14 points. Relative to test-time SEPC, it scores higher in communication and completeness, and lower in accuracy, instruction following, and awareness. The aggregate gap is 0.58 points.

Table 14: Full-set Qwen3-4B HealthBench Consensus axis scores with DeepSeek-V4-Flash. SE-OPSD uses seed 42, as in Table 4.

Method	Overall	Acc.	Comm.	Instr.	Aware.	Complete
Base	80.11	80.78	91.13	82.02	68.93	74.92
Base + test-time SEPC	84.42	81.78	89.11	82.61	77.81	80.90
SE-OPSD	83.84	80.92	91.84	78.48	74.62	82.06

Inference token accounting. Table 15 counts generation calls and input/output tokens over the full evaluation set. The test-time pipeline comprises five SEPC elicitation calls and one final-answer call per query; the trained SE-OPSD model uses one answer call. The retained gain is

$$\frac{S_{\text{SE-OPSD}} - S_{\text{Base}}}{S_{\text{test-time SEPC}} - S_{\text{Base}}} \times 100\% = \frac{83.84 - 80.11}{84.42 - 80.11} \times 100\% \approx 86.5\%.$$

The total-token ratio is $25.811/3.106 \approx 8.31$. These totals cover evaluation-time model generation, including elicitation for the test-time pipeline; training and grading compute are outside this accounting.

Table 15: Inference token accounting on 3,671 HealthBench Consensus queries. M denotes million tokens. The ratio uses SE-OPSD as the denominator.

Method	Calls	Input	Output	Total	Ratio
SE-OPSD	3,671	0.638M	2.468M	3.106M	1.00×
Test-time SEPC + answer	22,026	21.994M	3.817M	25.811M	8.31×

C.3 Effect of Teacher Thinking Mode

Prior work on on-policy self-distillation for mathematical reasoning reports that an asymmetric thinking configuration can be beneficial: the privileged teacher is allowed to use thinking mode, while the student is trained to answer without thinking at inference time (Zhao et al., 2026). This design is natural for verifiable reasoning tasks, where the teacher's explicit reasoning trace can provide a more informative training-time context while the student still learns a compact direct-answer policy.

We found that this configuration is less stable in open-ended medical QA. In a pilot ablation, we enabled thinking mode for the teacher while keeping the student in non-thinking direct-answer mode. During training, both the teacher and student token distributions became increasingly high-entropy, and the student's generated answers rapidly increased in length. In the middle and later stages of training, the student often learned to repeat generic safety and disclaimer templates until reaching the maximum generation length. This behavior was not observed to the same extent when both teacher and student used the same non-thinking direct-answer format.

We hypothesize that the instability comes from a mismatch between the thinking teacher and the direct-answer student in a high-entropy, open-ended setting. In mathematics or code, the privileged reasoning context is usually anchored by a well-defined correctness criterion, such as a final answer, verified trace, or executable outcome. In open-ended medical QA, however, there is often no unique gold answer or perfect reasoning trajectory. The diagnostic in Appendix D shows higher token-level entropy in our medical setup than in the compared mathematical-reasoning setup. Introducing a thinking-mode asymmetry can further enlarge the KL mismatch between the teacher and student distributions: the teacher assigns probability mass to continuation patterns induced by explicit deliberation, while the student is trained to produce a direct answer. Reduced probability of terminating with an EOS token is one possible explanation for the progressively longer answers observed in this pilot.

Figure 3 shows the corresponding training dynamics: teacher thinking mode increases both entropy and student output length, whereas the non-thinking configuration remains more stable. This pilot supports the compact response-context representation and matched non-thinking configuration used in our experiments.

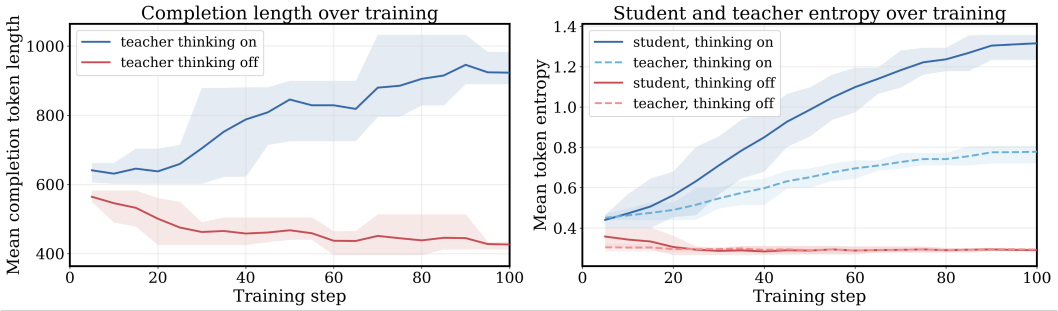


Figure 3: Training dynamics under teacher thinking mode. Left: student output length. Right: student and teacher token entropy.

Therefore, in our main experiments, we disable thinking mode for both teacher and student. The teacher receives SEPC as structured privileged context, while the student is trained to produce a direct answer from the original query. This keeps the teacher-student interface closer to the deployed inference format and avoids amplifying length and entropy through a thinking-mode mismatch.

C.4 Additional Grader and Training-Seed Results

Grader sensitivity. We rescore the same cached Qwen3-4B Base and SE-OPSD responses with DeepSeek-V4-Flash, holding response generation fixed. Table 16 reports a positive SE-OPSD gain with both graders. The difference between their absolute scores reflects the grading protocol; each gain is computed within a grader.

Table 16: Grader sensitivity on the same Qwen3-4B HealthBench Consensus responses.

Grader	Base	SE-OPSD	Δ
GPT-4.1	82.15	86.58	+4.43
DeepSeek-V4-Flash	80.11	83.84	+3.73

Training-seed variation. Table 17 reports three Qwen3-4B SE-OPSD training runs evaluated under the same DeepSeek-V4-Flash protocol. The mean is 83.56 and the sample standard deviation is 0.56. The Base score under this grader is 80.11, so each run improves over Base. These results measure variation across training runs; the bootstrap intervals in Appendix B.6 measure variation from resampling evaluation examples. Full-set SEPC transfer and token comparisons use the seed-42 checkpoint.

Table 17: Qwen3-4B SE-OPSD training-seed results on HealthBench Consensus with DeepSeek-V4-Flash. The summary uses the sample standard deviation over three runs.

Training seed	Overall
42	83.84
43	83.92
44	82.91
Mean ± sample SD	83.56 ± 0.56

D Difference between math and medical openQA

D.1 Diagnostic Entropy Analysis

Figure 4 provides a diagnostic comparison between open-ended medical QA and verifiable mathematical reasoning. The goal of this analysis is not to evaluate downstream performance, but to illustrate the different token-level uncertainty structure faced by on-policy distillation in the two settings.

We use the same Qwen3-4B instruction model for all conditions. For the open-ended setting, we use the medical QA training queries used in our experiments. For the mathematical reasoning setting, we use the OpenThought dataset following the OPSD setting. For each query x, we first sample one student trajectory

$$y = (y_1, \ldots, y_T) \sim \pi_\theta(\cdot \mid x).$$

We then evaluate both the student and the privileged teacher distributions on the same student prefixes. Specifically, for each assistant-answer token position t, we compute

$$p_t(\cdot) = \pi_\theta(\cdot \mid x, y_{<t})$$

for the student and

$$q_t(\cdot) = \pi_{\bar{\theta}}(\cdot \mid x, c, y_{<t})$$

for the privileged teacher, where c denotes the training-time privileged context. In the open-ended medical setting, c is the self-elicited privileged context (SEPC). In the mathematical setting, c is the privileged information used by the OPSD-style math setup. We compute token entropy as

$$H_t(p) = -\sum_v p_t(v) \log p_t(v),$$

and analogously for q_t. Entropies are computed only on supervised assistant-answer tokens; prompt tokens, padding tokens, and special tokens are excluded.

Table 18: Over-generation diagnostics for Ref-answer OPSD on Qwen3-1.7B HealthBench Consensus.

Diagnostic	Count	Fraction
Outputs longer than 1024 tokens	895 / 3671	24.4%
Outputs longer than 1536 tokens	296 / 3671	8.1%
Outputs reaching at least 4096 tokens	74 / 3671	2.0%
Repetitive outputs	260 / 3671	7.1%

The left panel of Figure 4 shows the empirical distribution of token entropies along student trajectories. The right panel plots the high-entropy token fraction

$$F(\tau) = \frac{\sum_{i,t} \mathbf{1}[H_{i,t} > \tau]}{\sum_i T_i},$$

as a function of the entropy threshold τ.

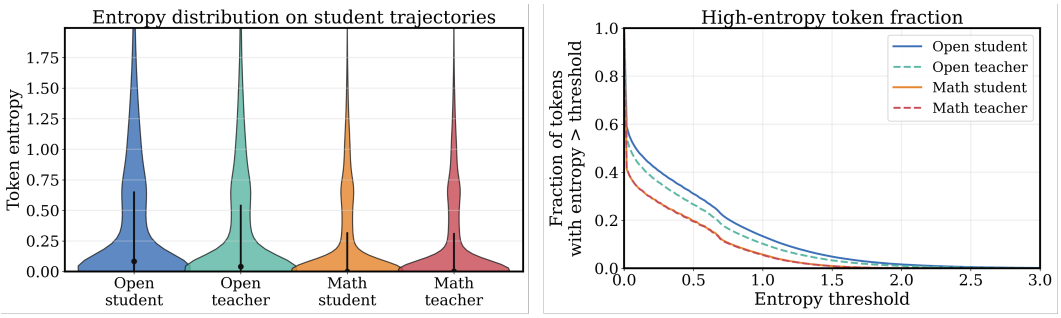


Figure 4: Diagnostic entropy comparison between open-ended medical QA and mathematical reasoning using Qwen3-4B. Left: token-level entropy distributions. Right: fraction of assistant-answer tokens above each entropy threshold.

In this comparison, the SEPC-conditioned medical teacher has a more diffuse next-token distribution than the teacher in the mathematical-reasoning setup. This pattern is consistent with the range of acceptable phrasings, ordering, levels of detail, and response modes in open-ended QA. It motivates expressing teacher guidance through response requirements such as missing context, uncertainty, safety boundaries, and audience needs. The diagnostic describes the two evaluated setups; the context ablations in Section 4.3 evaluate the corresponding design choices through downstream response quality.

E Reference-answer OPSD Over-generation and Failure Modes

This appendix provides additional diagnostics for the reference-answer OPSD baseline. The goal is not to argue that reference answers are unhelpful in general: Ref-answer SFT remains a stable and useful baseline. Rather, we analyze why using reference answers as teacher-side privileged information in on-policy token-level distillation can be fragile in open-ended medical QA.

E.1 Over-generation statistics

Ref-answer OPSD produces substantially longer outputs than the other methods on Qwen3-1.7B HealthBench Consensus. Many of these long outputs are not more complete answers, but instead contain excessive Markdown-style structure, generic safety boilerplate, repeated phrases, or failures to emit an end-of-sequence token. Table 18 summarizes these diagnostics.

E.2 Maximum-length outputs are low-scoring failures

To test whether the longest Ref-answer OPSD outputs are beneficially comprehensive or pathological, we examine the subset of examples for which Ref-answer OPSD reaches at least 4096 tokens. On

Table 19: Scores on examples where Ref-answer OPSD reaches at least 4096 output tokens.

Method evaluated on the same subset	Mean HealthBench score
Ref-answer OPSD	11.94
SE-OPSD	32.88

these examples, Ref-answer OPSD obtains an average score of 11.94, while SE-OPSD obtains 32.88 on the same examples. This indicates that the extreme length is usually not useful completeness, but a failure mode involving repetition, hallucinated structure, or inability to stop.

E.3 Qualitative failure patterns

Manual inspection of high-length and low-scoring Ref-answer OPSD outputs reveals several recurring failure patterns. Table 20 summarizes representative categories. These failures are especially common in multilingual or code-switched queries, transformation or rephrasing tasks, and long clinician-facing inputs, where the model must distinguish the requested operation from the subject matter of the source text.

Table 20: Representative failure patterns of Ref-answer OPSD in long or repetitive outputs.

Query type	Ref-answer OPSD failure	SE-OPSD behavior
Code-switched injury query	Repeats an irrelevant phrase until the maximum generation length, receiving near-zero score.	Provides concise first-aid guidance with appropriate escalation boundaries.
Diet or lifestyle query	Copies or paraphrases the user question repeatedly, without converging to a useful answer.	Gives bounded dietary guidance without excessive repetition.
Minor injury or blister query	Starts with relevant advice but then repeats generic reassurance or safety text.	Gives concise care advice and red-flag guidance.
Transformation or rephrasing request	Misclassifies the task as a new medical QA prompt, expands into a long clinical note, and may invent placeholders.	Preserves the source meaning and performs the requested transformation.
Prevention or vaccine-related query	Produces long structured lists with hallucinated or unreliable details.	Provides bounded prevention guidance and avoids unsupported specificity.