DISTILLING STRUCTURE, NOT SOLUTIONS: STRUCTURAL BOTTLENECKS FOR ON-POLICY SELF-DISTILLATION

Anonymous authors
Paper under double-blind review

ABSTRACT

On-policy self-distillation (OPSD) aligns supervision with student-generated trajectories but retains an information asymmetry: the teacher receives a verified solution that is unavailable to the student at inference. We investigate whether restricting this privileged context to reasoning structure improves distillation. Structural-Bottleneck OPSD (SB-OPSD) replaces full solutions with compact, result-free skeletons that preserve ordered reasoning operations while excluding intermediate results and final answers. A frozen, skeleton-conditioned teacher supplies next-token targets at student-generated prefixes, while the student uses a question-only prompt throughout training and inference. Across Qwen3-1.7B, 4B, and 8B on three competition-mathematics benchmarks, SB-OPSD achieves the highest mean Average@12 among the compared methods, improving over the respective base models by 5.70, 3.31, and 3.22 percentage points. Structural controls favor compact, ordered, question-matched skeletons. Relative to full-solution OPSD, SB-OPSD also reduces teacher-context and output lengths and exhibits a smaller gap to its conditioned teacher during training. These results support compact reasoning structure as an effective form of privileged supervision.

1 INTRODUCTION

Post-training for mathematical reasoning commonly uses worked solutions and generated rationales (Chen et al., 2025; Hsieh et al., 2023; Zelikman et al., 2022) or verifiable rewards (Shao et al., 2024). Learning from fixed solution traces introduces a trajectory mismatch: training uses reference prefixes, whereas inference requires the student to continue from its own predictions. On-policy distillation addresses this mismatch by querying the teacher at states visited by the current student (Ross et al., 2011; Agarwal et al., 2024). On-Policy Self-Distillation (OPSD) uses solution-conditioned teacher and question-only student views of the same model to match next-token distributions along student-generated trajectories (Zhao et al., 2026a).

Trajectory alignment does not remove the difference in available information. The OPSD teacher observes a verified solution, including intermediate results, branch decisions, and the final answer; the student must learn to solve from the question alone. We call this asymmetry *privilege mismatch*. Solution-specific information can make the teacher confident without making its predictions easier for the student to learn. A more informed teacher may therefore provide less transferable supervision. Figure 1 illustrates this potential failure mode.

This motivates our central question: *what information should the privileged teacher receive?* We hypothesize that ordered reasoning operations—such as introducing an invariant, separating cases, or checking a constraint—can provide useful guidance without exposing the results of those operations. The challenge is to preserve a question-specific plan while excluding explicit intermediate results and final answers.

We operationalize this idea with *Structural-Bottleneck On-Policy Self-Distillation* (SB-OPSD). An offline extractor converts each verified solution into a compact *structural skeleton*: an ordered sequence of reasoning operations with their results removed. During distillation, a frozen teacher conditions on the question, skeleton, and student-generated prefix; the student conditions only on the question and its prefix. The trained student retains the same question-only interface at inference.

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.

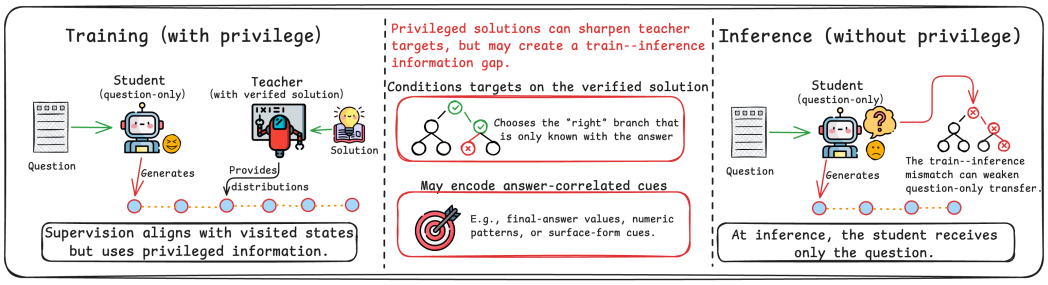

Figure 1: On-policy supervision aligns the teacher with student-visited states, but a teacher conditioned on the verified solution can exploit answer-dependent branch information or shortcuts. At inference time, the question-only student must reason without this privileged input.

We evaluate this design across three Qwen3 scales and three competition benchmarks. Comparisons from functional tags to full solutions examine how the amount of privileged information affects performance. Ordering and cross-sample controls test the contribution of coherent structure and question–skeleton alignment, while a support–query evaluation measures transfer to related problems with different answers.

Contributions. **First,** we characterize privilege mismatch as an information asymmetry that remains after on-policy trajectory alignment. **Second,** we introduce a compact structural representation for teacher-side supervision while preserving question-only student inputs. **Third,** SB-OPSD achieves the highest three-benchmark mean Average@12 among the compared methods at all three model scales. The accompanying controls favor ordered, matched skeletons, and the diagnostics show lower teacher-context and output lengths than full-solution OPSD.

2 Related Work

2.1 On-policy distillation

Knowledge distillation transfers teacher predictions (Hinton et al., 2015), while sequence-level distillation trains on teacher-generated outputs (Kim & Rush, 2016). DAgger supervises learner-visited states (Ross et al., 2011); MiniLLM and GKD extend student-generated training to language-model distillation (Gu et al., 2024; Agarwal et al., 2024). On-policy context distillation internalizes contextual knowledge (Ye et al., 2026), and OPSD obtains solution-conditioned targets at the student's own prefixes (Zhao et al., 2026a). Recent analyses examine when privileged signals help (Zhu et al., 2026). Purified OPSD constructs targets that suppress reference-induced shortcuts (Shen et al., 2026b); anti-self-distillation reverses the distillation signal under an entropy-triggered gate (Shen et al., 2026a); and AVSD combines cross-view consensus with selected view-specific signals (Nguyen et al., 2026). SB-OPSD instead intervenes on the representation supplied to the teacher, using a result-free skeleton with on-policy distribution matching.

2.2 Privileged information and reasoning supervision

Chain-of-thought prompting exposes intermediate reasoning traces (Wei et al., 2022), and rationale-supervised methods distill or bootstrap such traces (Hsieh et al., 2023; Zelikman et al., 2022; Wang et al., 2023b). Process-level and verifier-based supervision provide complementary signals for mathematical reasoning (Cobbe et al., 2021; Lightman et al., 2024). Planning-style methods explicitly separate higher-level decomposition from token-level execution (Wang et al., 2023a; Yao et al., 2023). Our setting follows learning using privileged information (Vapnik & Izmailov, 2015; Lopez-Paz et al., 2016). D-RPC organizes rationales around reusable reasoning paths (Yang et al., 2026), while PS-OPSD supplies explicit state-transition guidance to the teacher (Zhao et al., 2026b). SB-OPSD focuses on concise, ordered operations that exclude intermediate outcomes and final answers. Trace quality affects distillation (Chen et al., 2025), and a plausible chain of thought need not faith-

fully explain a model's prediction (Turpin et al., 2023; Lanham et al., 2023). Our evaluation therefore focuses on the student's answer accuracy when privileged context is absent.

3 METHOD

Figure 2 summarizes the three stages of SB-OPSD: offline skeleton construction, on-policy distillation with teacher-only structural context, and question-only inference.

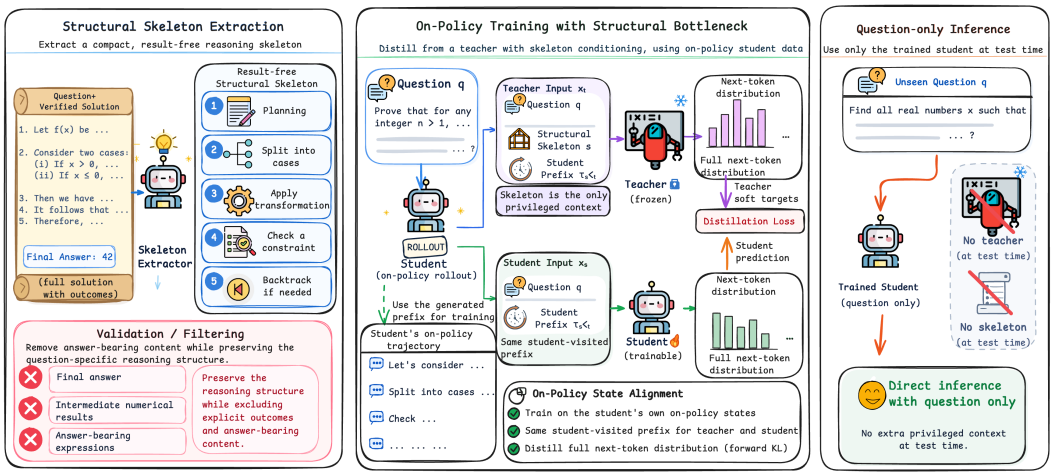

Figure 2: Overview of SB-OPSD. Left: an offline extractor converts a verified solution into a result-free structural skeleton, followed by validation and filtering. Center: the question-only student generates on-policy responses; the frozen teacher additionally receives the skeleton and supplies next-token targets at the same student-generated prefixes. Forward-KL distillation updates only the student. Right: the trained student answers new questions without the teacher or skeleton.

3.1 PROBLEM FORMULATION AND DESIGN OBJECTIVE

Let $\mathcal{D} = \{(Q_i, A_i)\}_{i=1}^{N}$ contain questions Q_i and verified solutions A_i, including final answers. The frozen teacher p_{θ_0} and trainable student p_θ start from the same checkpoint. The student's prompt contains only Q; the teacher additionally receives a structural representation $S = g(Q, A)$ during training.

On-policy distillation aligns trajectories by scoring student-generated prefixes, but leaves a *privilege mismatch*: the teacher sees solution-derived information unavailable to the student at inference. SB-OPSD restricts this information to compact, result-free operations. The goal is to retain structural guidance in the teacher's targets while training the student to solve from the question alone.

Crucially, the asymmetry is restricted to teacher conditioning rather than the student state. At each update, both distributions are evaluated after the same student-generated prefix, so their difference reflects structural conditioning rather than a change in trajectory. The bottleneck separates two roles that full-solution supervision combines: identifying a useful reasoning operation and revealing the outcome of that operation. SB-OPSD provides only the former to the teacher and requires the student to execute the operation from its question-only state.

3.2 RESULT-FREE STRUCTURAL BOTTLENECK

SB-OPSD uses a fixed structural representation $S = g(Q, A)$ as the only privileged input to the teacher. A skeleton is an ordered sequence

$$S = \big((c_1, o_1), \ldots, (c_L, o_L)\big), \tag{1}$$

where c_l is a functional tag and o_l describes one reasoning operation. The tag vocabulary covers planning, retrieval, inference, verification, summarization, branching, backtracking, and reflection. The operation text is written as an instruction, such as selecting a theorem, introducing an invariant, separating cases, propagating a constraint, or checking a candidate. It must not state the outcome of that operation; see Appendix C for a concrete example.

Construction. Skeletons are generated offline from the verified solution and remain fixed throughout student training. The extractor is asked to preserve the order and dependencies of the decisive operations while compressing away explanatory prose and routine algebra. A valid skeleton should satisfy four principles: (i) *procedural sufficiency*, retaining the operations needed to identify the solution strategy; (ii) *result exclusion*, removing final and intermediate results; (iii) *instance grounding*, referring only to objects already present in the question; and (iv) *bounded bandwidth*, using a short sequence of concise operations.

Filtering and validation. A deterministic validator first enforces the sequence schema and functional tag set. It then rejects final-answer strings, answer labels, result-bearing equations, and expressions equivalent to the verified answer. To prevent new numerical clues from being introduced, it requires $\mathrm{Nums}(S) \setminus \mathrm{Nums}(Q) = \emptyset$. It also checks that named entities and symbols are grounded in Q and limits lexical copying from A. Invalid candidates are regenerated up to a fixed budget and otherwise removed from the skeleton training set. Construction and filtering are detailed in Appendix B.3.

Asymmetric visibility. The skeleton is never appended to the student input. It is visible only to the frozen teacher while producing training targets. This separation is essential: giving the skeleton directly to the student would train a different inference problem and would not test whether structural guidance can be internalized.

3.3 Structural-Bottleneck On-Policy Self-Distillation

For a sampled training question, the current student first generates an on-policy completion without privileged information,

$$\widetilde{Y} = (\widetilde{y}_1, \ldots, \widetilde{y}_T) \sim p_\theta(\cdot \mid Q). \tag{2}$$

The sampled completion may be correct, partially correct, or incorrect. We then construct two views of the same prefix. The student view contains $(Q, \widetilde{y}_{<t})$, whereas the teacher view contains $(Q, S, \widetilde{y}_{<t})$. At response position t, the corresponding predictive distributions are

$$p^S_{T,t}(v) = p_{\theta_0}(v \mid Q, S, \widetilde{y}_{<t}), \tag{3}$$

$$p_{S,t}(v) = p_\theta(v \mid Q, \widetilde{y}_{<t}). \tag{4}$$

Both models are evaluated at the same student-generated prefixes under teacher forcing; their input contexts differ only in the structural skeleton. The teacher remains frozen at θ_0, and only the student parameters are optimized. Unlike rationale distillation, the teacher does not generate a replacement solution trace: supervision is provided by its full next-token distribution along the student's own trajectory.

Let $\mathcal{M}$ denote the valid completion-token positions, excluding the prompt and padding. We minimize the forward KL divergence from the privileged teacher to the question-only student,

$$\mathcal{L}_{\mathrm{SB}} = \frac{1}{|\mathcal{M}|} \sum_{t \in \mathcal{M}} D_{\mathrm{KL}}\big(\mathrm{sg}[p^S_{T,t}] \,\big\|\, p_{S,t}\big), \tag{5}$$

where sg stops gradients through the teacher. This objective transfers the teacher's privileged next-token distribution on states visited by the current student rather than on a fixed offline trajectory. Equivalently, each token loss is

$$\mathcal{L}_t = \sum_v p^S_{T,t}(v) \log \frac{p^S_{T,t}(v)}{p_{S,t}(v)}, \tag{6}$$

so probability mass over the full vocabulary is transferred, including alternative continuations that do not appear in the sampled response.

Training procedure. Each optimization step consists of four operations: (1) sample questions and generate responses from the current question-only student; (2) pair each response with its fixed skeleton only on the teacher side; (3) score the same response positions with the frozen teacher and trainable student; and (4) update the student using Equation 5. The structural contexts are fixed across training, whereas the responses are repeatedly resampled as the student policy changes. In our matched OPSD and SB-OPSD implementations, the teacher is frozen and the training procedure is shared; the teacher-side context changes from the full solution to the structural skeleton.

3.4 Question-only inference

After training, both the frozen teacher and the skeleton extractor are discarded. The resulting model generates from

$$Y \sim p_\theta(\cdot \mid Q), \tag{7}$$

using the same question-only interface as the base model. SB-OPSD therefore adds no privileged tokens, retrieval stage, or teacher-model computation at inference time. Any improvement must be represented in the updated student parameters rather than supplied by an external skeleton.

4 Experiments

4.1 Experimental setup

Models and datasets. We study instruction-tuned Qwen3-1.7B, 4B, and 8B (Yang et al., 2025), using OpenThoughts mathematics for training (Guha et al., 2025) and AIME 2024/2025 (Mathematical Association of America, n.d.) and HMMT February 2025 (Harvard–MIT Mathematics Tournament, 2025) for evaluation. Our primary metric, Average@12, is the percentage of correct responses across 12 samples per problem. Avg is the unweighted mean over the three benchmarks; gains are expressed in percentage points.

Baselines and variants. We compare the base model (B_0), solution SFT, GRPO (Shao et al., 2024), full-solution OPSD (B_4) (Zhao et al., 2026a), and AVSD (Nguyen et al., 2026). The privileged-information hierarchy comprises functional tags (B_1), short skeletons or SB-OPSD (B_2), detailed skeletons (B_3), and full-solution OPSD (B_4). Structural controls include summary-only context, step-order shuffling, reversed order, and cross-sample skeletons.

Implementation details. In our OPSD implementations, teacher and student start from the same checkpoint; the teacher remains frozen while only the student's LoRA parameters (Hu et al., 2022) are updated. Reported means and standard deviations summarize three independent training seeds. Training and decoding settings appear in Appendix A.1, and the algorithm is given in Appendix A.2. Condition definitions and data construction are detailed in Appendix B.

4.2 Main results

Table 1 compares SB-OPSD with the principal baselines at three model scales. SB-OPSD achieves the highest three-benchmark mean Average@12 in each group: 42.36%, 65.07%, and 65.53% at 1.7B, 4B, and 8B, respectively. The corresponding gains over Base are 5.70, 3.31, and 3.22 points, computed from the benchmark-wise differences before rounding. Relative to full-solution OPSD, the mean advantages are 1.28, 3.37, and 1.95 points. Thus, the aggregate benefit of compact structural guidance holds across all three evaluated model sizes.

Table 1: Average@12 (%). Trained-method entries report the mean ± standard deviation across three independent training seeds; Base rows have no reported dispersion. Boldface marks the best mean within each model scale and benchmark column; Avg is recomputed from the displayed benchmark means.

Method	AIME24	AIME25	HMMT25	Avg
Qwen3-1.7B				
Base	48.89	38.33	22.78	36.67
SFT	51.11±0.28	38.70±0.70	26.76±0.70	38.86±0.39
GRPO	49.44±0.56	40.19±0.32	23.43±0.98	37.69±0.09
OPSD	53.80±0.89	41.67±0.83	**27.78**±0.56	41.08±0.28
AVSD	51.85±0.16	38.61±0.73	26.57±1.53	39.01±0.57
SB-OPSD	**55.83**±0.84	**43.61**±0.92	27.65±0.88	**42.36**±0.55
Qwen3-4B				
Base	75.28	66.39	43.61	61.76
SFT	74.35±0.42	66.30±1.53	44.17±1.00	61.61±0.79
GRPO	75.74±0.85	66.57±1.40	44.81±0.70	62.37±0.84
OPSD	73.06±0.73	68.06±1.00	43.98±0.70	61.70±0.79
AVSD	75.46±0.70	66.67±0.28	46.20±0.42	62.78±0.19
SB-OPSD	**77.20**±0.62	**71.20**±0.68	**46.80**±0.75	**65.07**±0.42
Qwen3-8B				
Base	75.00	66.94	45.00	62.31
SFT	76.02±0.58	67.78±1.69	43.80±1.63	62.53±1.02
GRPO	75.83±0.48	69.35±0.58	44.72±1.00	63.30±0.46
OPSD	75.74±0.98	70.09±0.89	44.91±0.98	63.58±0.62
AVSD	77.78±0.73	66.67±1.00	**46.48**±1.25	63.64±0.96
SB-OPSD	**78.30**±0.55	**72.00**±0.61	46.30±0.66	**65.53**±0.36

SB-OPSD also leads both AIME benchmarks at every scale and all three benchmarks at 4B. On HMMT 2025, OPSD is 0.13 points higher at 1.7B and AVSD is 0.18 points higher at 8B. The principal finding is therefore a consistent advantage in the three-benchmark average, with variation across individual benchmarks.

5 Analysis

5.1 Privileged-information bandwidth across model scales

Figure 3 compares five conditions: the base model (B_0), functional-tag OPSD (B_1), SB-OPSD with short skeletons (B_2), detailed-skeleton OPSD (B_3), and standard full-solution OPSD (B_4). We evaluate all conditions question-only across three Qwen3 scales and three benchmarks; detailed definitions appear in Appendix B.1.

Within this hierarchy, SB-OPSD (B_2) has the highest three-benchmark average at every scale and leads both AIME benchmarks. It also leads HMMT 2025 at 4B and 8B among B_0–B_4. The comparison favors short skeletons over both coarse functional tags and more detailed privileged contexts. Increasing the amount of teacher-side information therefore does not yield a corresponding increase in student accuracy in these experiments.

5.2 Query-transfer evaluation

We measure how quickly training on each privileged-information condition transfers to related queries once privilege is removed. Starting from the same Qwen3-1.7B initialization, we evaluate B_1–B_4 question-only at $k \in \{0, 1, 5, 20\}$ update steps. The support–query split, answer separation, and evaluation accounting are detailed in Appendix B.2.

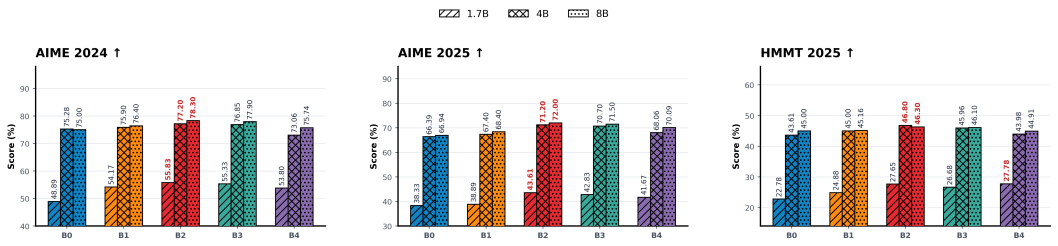

Figure 3: Effect of privileged-information bandwidth across model scales. Average@12 results on AIME 2024, AIME 2025, and HMMT 2025 for Qwen3-1.7B, Qwen3-4B, and Qwen3-8B. B_0 is the base model, B_1 is functional-tag OPSD, B_2 is SB-OPSD, B_3 is detailed-skeleton OPSD, and B_4 is standard full-solution OPSD.

The fixed Base reference answers 1,122 of 1,275 queries correctly (88.00%). We report accuracy and its percentage-point change from this reference.

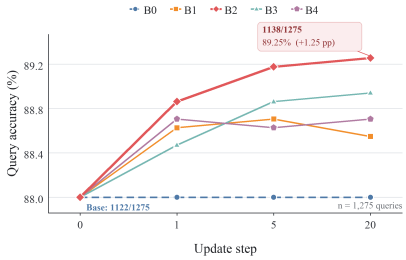

Figure 4: Query accuracy across update steps.

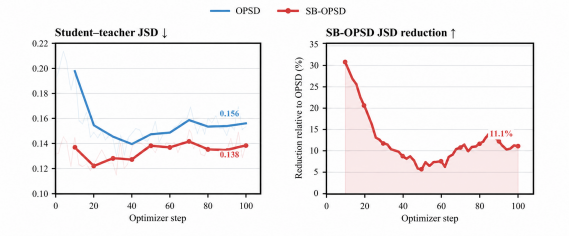

Figure 5: Teacher–student gap during training. Left: logged JSD. Right: SB-OPSD's relative JSD reduction over OPSD.

In Figure 4, SB-OPSD (B_2) leads at every post-update checkpoint. Its gains increase from $+0.86$ points at step 1 to $+1.18$ at step 5 and $+1.25$ at step 20, reaching 1,138/1,275 correct (89.25%). At step 20, the gains for B_1, B_3, and B_4 are $+0.55$, $+0.94$, and $+0.71$ points. Compact skeletons thus provide the largest observed transfer gain throughout the measured update range.

5.3 Optimization Quality and Evaluation Efficiency

Figure 6 shows that SB-OPSD produces shorter responses on all three benchmarks: 18.3k versus 18.4k tokens on AIME 2024, 18.9k versus 19.9k on AIME 2025, and 19.7k versus 20.3k on HMMT 2025. Together with Figure 3, these measurements show that the accuracy gains are accompanied by shorter average outputs.

The training diagnostics also favor SB-OPSD: the ratio of logged clipped-token fractions (OPSD/SB-OPSD) stays above 1 and ends at $1.10\times$, while the final repetition diagnostic is lower (7.8% versus 9.2%). The teacher-to-student prompt-length ratio decreases from $7.9\times$ to $2.2\times$. These measurements indicate lower privileged context overhead and improved logged generation behavior.

5.4 Structural Content and Ordering

Table 2 tests skeleton content, operation order, and question–skeleton alignment on Qwen3-1.7B.

SB-OPSD leads in average accuracy (42.36%) and on both AIME benchmarks, beating all result-free controls on every benchmark. Full-solution OPSD slightly leads on HMMT (27.78% versus 27.65%) but has a lower average.

Order shuffling and reversal fall below Base by 0.50 and 2.66 points on average. Within these controls, excluding results alone is not sufficient; the ordering of the retained operations also matters.

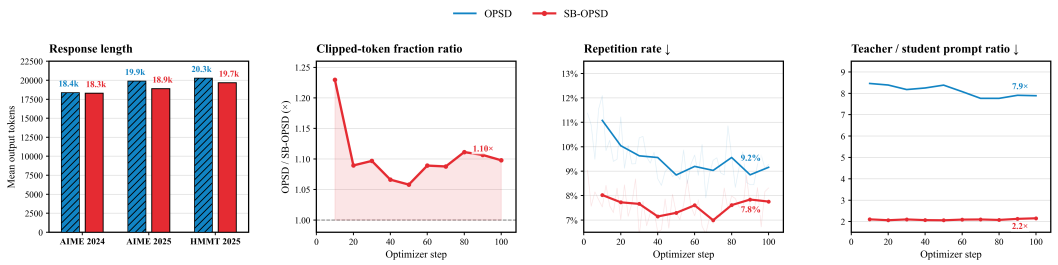

Figure 6: Efficiency diagnostics for full-solution OPSD and SB-OPSD. Left to right: mean output length across 12 samples per problem, the logged clipped-token fraction ratio (OPSD/SB-OPSD), the recorded repetition diagnostic, and the teacher-to-student prompt-length ratio. Endpoints are final-window values; downward arrows indicate lower is better.

Table 2: Structural-control analysis for Qwen3-1.7B on AIME 2024, AIME 2025, and HMMT 2025. Results are average@12 accuracies (%); Avg denotes the arithmetic mean across the three benchmarks.

Condition	AIME24	AIME25	HMMT25	Avg
Base model (B_0)	48.89	38.33	22.78	36.67
SB-OPSD (B_2)	**55.83**	**43.61**	27.65	**42.36**
Detailed skeleton (B_3)	55.33	42.83	26.68	41.61
Summary-only	52.83	40.56	26.97	40.12
Step-order shuffle	47.78	38.61	22.11	36.17
Reversed order	45.50	36.06	20.46	34.01
Cross-sample shuffle	47.82	40.34	24.87	37.68
Full-solution OPSD (B_4)	53.80	41.67	**27.78**	41.08

Cross-sample shuffling retains a 1.01-point average gain with mixed benchmark outcomes. Overall, the strongest and most consistent gains arise when the skeleton preserves both the order and question-specific alignment of the teacher's reasoning structure.

5.5 Majority-vote accuracy

Table 3 reports majority-vote@12 accuracy for Qwen3-1.7B (Wang et al., 2023c): the most frequent answer among 12 responses determines correctness. For trained methods, each benchmark's 30 problems are evaluated with three independently trained checkpoints, yielding 90 problem–seed instances. Parentheses count correct majority-vote outcomes; they do not count individual responses.

Table 3: Majority-vote@12 accuracy on AIME 2024 and AIME 2025. Each benchmark contains 30 unique problems; trained methods use three independent seed checkpoints. Parentheses report correct majority-vote outcomes out of 90 evaluations.

Test set	Base model	Solution SFT	OPSD	SB-OPSD
AIME 2024	72.22% (65/90)	52.22% (47/90)	61.11% (55/90)	**75.56% (68/90)**
AIME 2025	47.78% (43/90)	44.44% (40/90)	53.33% (48/90)	**56.67% (51/90)**

SB-OPSD leads on both benchmarks, exceeding Base by 3.33 and 8.89 percentage points on AIME 2024 and 2025, and OPSD by 14.44 and 3.33 points. Solution SFT falls below Base on both. Majority voting thus provides a complementary problem-level view of SB-OPSD's advantage.

5.6 Problem-level robustness and hard-case retention

Table 4 evaluates Qwen3-4B on 90 unique problems across the three benchmarks. ΔBase is the change in the three-benchmark mean Average@12 from Table 1. For each problem, Bet-

ter/Worse/Tied compare the method's 12-sample accuracy with Base. The hard subset contains the 23 problems for which Base has no correct sample: Hard Avg is Average@12 on that subset, and Hard Pass is the fraction with at least one correct sample. Ret. is the fraction of the other 67 Base-solved problems that remain solved, and Net Exp. is newly solved hard problems minus lost Base-solved problems.

Table 4: Robustness and hard-subset retention on 90 problems across AIME 2024, AIME 2025, and HMMT 2025. ΔBase is the base-relative accuracy change in this evaluation, in percentage points. Better, Worse, and Tied are problem counts.

	Pairwise comparison with Base				Hard-subset analysis			
Method	**ΔBase**	**Better**	**Worse**	**Tied**	**Hard Avg**	**Hard Pass**	**Ret.**	**Net Exp.**
Qwen3-4B	+0.00	0	0	90	0.00	0.00	**100.00**	0
+ SFT	−0.15	13	15	62	0.85	4.35	97.01	−1
+ GRPO	+0.61	17	13	60	**6.04**	13.04	98.51	+2
+ OPSD	−0.06	13	11	**66**	1.69	8.70	97.01	0
+ SB-OPSD	**+3.31**	**17**	**11**	**62**	4.06	**17.39**	98.51	**+3**

SB-OPSD gains 3.31 points, with 17 improved, 11 degraded, and 62 unchanged problems: two fewer regressions than GRPO at the same improvement count. Compared with OPSD, hard-case pass rate rises from 8.70% to 17.39%, retention from 97.01% to 98.51%, and net expansion from 0 to +3. SB-OPSD has the highest hard-case pass rate and net expansion, matching GRPO's retention; GRPO has higher hard-case average accuracy (6.04% versus 4.06%).

5.7 Teacher–student gap dynamics

Figure 5 compares the logged mean Jensen–Shannon divergence (JSD) between teacher and student next-token distributions over valid completion positions. Lower JSD indicates closer distributional agreement. At step k, the relative reduction is $100(J_k^{\mathrm{OPSD}} - J_k^{\mathrm{SB}})/J_k^{\mathrm{OPSD}}$; a positive value favors SB-OPSD. Its logged JSD remains lower throughout training and ends at 0.138, compared with 0.156 for OPSD. This is consistent with the student more closely matching its skeleton-conditioned teacher.

The relative reduction is largest early in training, at about 31%, decreases to roughly 6% midway through optimization, and recovers to approximately 11% at the endpoint. The gap advantage therefore persists across training stages, with a varying magnitude.

6 Conclusion

We introduced SB-OPSD, which uses compact, result-free reasoning skeletons as teacher-side context for on-policy self-distillation. The student learns from token-level targets at its own generated prefixes and retains question-only inference. Across three Qwen3 scales and three competition-mathematics benchmarks, SB-OPSD achieves the highest mean Average@12 among the evaluated methods. The gains over Base are 5.70, 3.31, and 3.22 percentage points at 1.7B, 4B, and 8B, respectively. Structural controls favor coherent operation order and question–skeleton alignment, while the support–query evaluation shows sustained gains over the measured update range.

The benefits extend to several complementary measurements. SB-OPSD uses shorter teacher context and produces shorter evaluation outputs than full-solution OPSD. On Qwen3-4B, it attains a 17.39% hard-case pass rate and a net gain of three solved problems. Separate training diagnostics show closer agreement with the skeleton-conditioned teacher. Together, these findings support compact structural guidance as an effective form of privileged supervision in the evaluated setting. They motivate further study of which reasoning operations transfer across tasks and how the appropriate amount of structural detail varies with model capacity.

AI USE STATEMENT

Generative AI tools were used solely for English-language editing and stylistic polishing. All scientific ideas, methods, experiments, data, analyses, code, and conclusions were produced and verified by the authors.

REPRODUCIBILITY STATEMENT

The anonymized implementation is available at `https://anonymous.4open.science/r/sb-opsd-6D27/`. It provides the training and evaluation pipeline, skeleton construction and validation, baseline implementations, and launch configurations. Training settings and the SB-OPSD algorithm are documented in Appendices A.1 and A.2; Appendix B describes the data and evaluation protocols. External datasets, model weights, and generated artifacts are prepared separately using the repository's documented entry points.

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://arxiv.org/abs/2306.13649.

Xinghao Chen, Zhijing Sun, Wenjin Guo, Miaoran Zhang, Yanjun Chen, Yirong Sun, Hui Su, Yijie Pan, Dietrich Klakow, Wenjie Li, and Xiaoyu Shen. Unveiling the key factors for distilling chain-of-thought reasoning. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 15094–15119. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.findings-acl.782. URL https://aclanthology.org/2025.findings-acl.782/.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021. URL https://arxiv.org/abs/2110.14168.

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. MiniLLM: Knowledge distillation of large language models. In *International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2306.08543.

Etash Guha, Ryan Marten, Sedrick Keh, Negin Raoof, Georgios Smyrnis, Hritik Bansal, et al. OpenThoughts: Data recipes for reasoning models. *arXiv preprint arXiv:2506.04178*, 2025. URL https://arxiv.org/abs/2506.04178.

Harvard–MIT Mathematics Tournament. Archive of february 2025, 2025. URL https://www.hmmt.org/www/archive/282. Official competition problems and solutions. Accessed September 7, 2026.

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.

Cheng-Yu Hsieh, Chun-Liang Li, Chih-kuan Yeh, Hootan Nakhost, Yasuhisa Fujii, Alex Ratner, Ranjay Krishna, Chen-Yu Lee, and Tomas Pfister. Distilling step-by-step! outperforming larger language models with less training data and smaller model sizes. In *Findings of the Association for Computational Linguistics: ACL 2023*, pp. 8003–8017. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.findings-acl.507. URL https://aclanthology.org/2023.findings-acl.507/.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. URL https://arxiv.org/abs/2106.09685.

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. Association for Computational Linguistics, 2016. doi: 10.18653/v1/D16-1139. URL https://aclanthology.org/D16-1139/.

Tamera Lanham, Anna Chen, Ansh Radhakrishnan, Benoit Steiner, Carson Denison, Danny Hernandez, et al. Measuring faithfulness in chain-of-thought reasoning. *arXiv preprint arXiv:2307.13702*, 2023. URL https://arxiv.org/abs/2307.13702.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2305.20050.

David Lopez-Paz, Léon Bottou, Bernhard Schölkopf, and Vladimir Vapnik. Unifying distillation and privileged information. In *International Conference on Learning Representations*, 2016. URL https://arxiv.org/abs/1511.03643.

Mathematical Association of America. MAA invitational competitions, n.d. URL https://maa.org/maa-invitational-competitions/. Official competition resource. Accessed September 7, 2026.

Duy Nguyen, Hanqi Xiao, Archiki Prasad, Zaid Khan, Anirban Das, Austin Zhang, Sambit Sahu, Hyunji Lee, Elias Stengel-Eskin, and Mohit Bansal. AVSD: Adaptive-view self-distillation by balancing consensus and teacher-specific privileged signals. *arXiv preprint arXiv:2605.20643*, 2026. URL https://arxiv.org/abs/2605.20643.

Stephane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, volume 15, pp. 627–635. PMLR, 2011. URL https://proceedings.mlr.press/v15/ross11a.html.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024. URL https://arxiv.org/abs/2402.03300.

Guobin Shen, Xiang Cheng, Chenxiao Zhao, Lei Huang, Jindong Li, Dongcheng Zhao, and Xing Yu. Anti-self-distillation for reasoning RL via pointwise mutual information. *arXiv preprint arXiv:2605.11609*, 2026a. URL https://arxiv.org/abs/2605.11609.

Zhanming Shen, Jintao Tong, Shaotian Yan, Chen Shen, Hao Chen, Wentao Ye, Xiaomeng Hu, Rui Miao, Haobo Wang, Junbo Zhao, Gang Chen, and Jieping Ye. Purified OPSD: On-policy self-distillation without losing how to think. *arXiv preprint arXiv:2607.02234*, 2026b. URL https://arxiv.org/abs/2607.02234.

Miles Turpin, Julian Michael, Ethan Perez, and Samuel R. Bowman. Language models don't always say what they think: Unfaithful explanations in chain-of-thought prompting. In *Advances in Neural Information Processing Systems*, 2023. URL https://arxiv.org/abs/2305.04388.

Vladimir Vapnik and Rauf Izmailov. Learning using privileged information: Similarity control and knowledge transfer. *Journal of Machine Learning Research*, 16(61):2023–2049, 2015. URL https://jmlr.org/papers/v16/vapnik15b.html.

Lei Wang, Wanyu Xu, Yihuai Lan, Zhiqiang Hu, Yunshi Lan, Roy Ka-Wei Lee, and Ee-Peng Lim. Plan-and-solve prompting: Improving zero-shot chain-of-thought reasoning by large language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 2609–2634. Association for Computational Linguistics, 2023a. doi: 10.18653/v1/2023.acl-long.147. URL https://aclanthology.org/2023.acl-long.147/.

Peifeng Wang, Zhengyang Wang, Zheng Li, Yifan Gao, Bing Yin, and Xiang Ren. SCOTT: Self-consistent chain-of-thought distillation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 5546–5558. Association for Computational Linguistics, 2023b. URL https://aclanthology.org/2023.acl-long.304/.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. In *International Conference on Learning Representations*, 2023c. URL https://arxiv.org/abs/2203.11171.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, volume 35, 2022. URL https://proceedings.neurips.cc/paper/2022/hash/9d5609613524ecf4f15af0f7b31abca4-Abstract-Conference.html.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025. URL https://arxiv.org/abs/2505.09388.

Jialin Yang, Jiankun Wang, Jiajun Wu, Henry Leung, Jiayu Zhou, and Steve Drew. Structural rationale distillation via reasoning space compression. *arXiv preprint arXiv:2605.07139*, 2026. URL https://arxiv.org/abs/2605.07139.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. In *Advances in Neural Information Processing Systems*, volume 36, 2023. URL https://arxiv.org/abs/2305.10601.

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.

Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah D. Goodman. STaR: Bootstrapping reasoning with reasoning. In *Advances in Neural Information Processing Systems*, volume 35, 2022. URL https://arxiv.org/abs/2203.14465.

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*, 2026a. URL https://arxiv.org/abs/2601.18734.

Xuyang Zhao, Liting Zhang, Zichen Xu, Zhihu Wang, Caiyue Xu, Shiwan Zhao, and Qicheng Li. Is more privileged information better? from solution traces to problem-solving structure in self-distilled reasoning. *arXiv preprint arXiv:2608.01589*, 2026b. URL https://arxiv.org/abs/2608.01589.

Siqi Zhu, Xuyan Ye, Hongyu Lu, Weiye Shi, and Ge Liu. The many faces of on-policy distillation: Pitfalls, mechanisms, and fixes. *arXiv preprint arXiv:2605.11182*, 2026. URL https://arxiv.org/abs/2605.11182.

APPENDIX ROADMAP

Appendix A Implementation and Reproducibility Details: training configuration, the SB-OPSD training procedure, and measured efficiency.

Appendix B Extended Experimental Protocol: condition definitions, skeleton construction, filtering, the support–query split, and metrics.

Appendix C Worked Example of the Privileged-Information Hierarchy: a concrete comparison of the teacher-side information available under conditions B_0–B_4.

A IMPLEMENTATION AND REPRODUCIBILITY DETAILS

This section records the training and evaluation configuration, the SB-OPSD training procedure, and efficiency measurements.

A.1 TRAINING AND EVALUATION CONFIGURATION

Table 5 records the initial configuration used for the OPSD reproduction and SB-OPSD experiments. The full-training horizon is 100 steps. The frozen teacher and trainable student start from the same checkpoint; only the student uses trainable LoRA adapters (Hu et al., 2022).

Table 5: Initial training and evaluation configuration.

Parameter	**Value**
Trainable parameters	LoRA only
LoRA rank / alpha	64 / 128
LoRA target modules	q, k, v, o, gate, up, down projections
Optimizer / precision	AdamW / bfloat16
Learning rate	5×10^{-6}
Effective batch size	32
Training steps	100 (full-training horizon)
Rollouts per prompt	1
Training temperature	1.1
Maximum training completion	1,024 tokens
Primary divergence	forward KL, full vocabulary
Completed training seeds	3 independent seeds
Evaluation samples	12 per problem
Evaluation temperature / top-p	1.0 / 0.95
Maximum evaluation completion	38,912 tokens

A.2 SB-OPSD TRAINING ALGORITHM

Algorithm 1 gives the SB-OPSD training procedure.

A.3 TRAINING TIME PER STEP

Table 6 compares the training time per step for OPSD and SB-OPSD. Relative time is normalized to OPSD and computed before rounding: $39.7/41.5 \approx 0.96$ for SB-OPSD.

A.4 OUTPUT-TOKEN EFFICIENCY

Table 7 reports the median and 90th percentile (P90) of output length, together with tokens per correct response. Compared with OPSD, SB-OPSD reduces the median by 1,056 tokens and P90 by 1,022 tokens. Tokens per correct response decrease from 47,384 to 43,958, a relative reduction of $100 \times (47{,}384 - 43{,}958)/47{,}384 \approx 7.23\%$.

Algorithm 1 Structural-Bottleneck On-Policy Self-Distillation

Require: Dataset $\mathcal{D} = \{(Q_i, A_i)\}$, extractor g, validator V, frozen teacher p_{θ_0}, student p_θ, training steps K
Ensure: Trained question-only student p_θ

Stage 1: Offline skeleton construction
$\mathcal{D}_S \leftarrow \emptyset$
for all $(Q_i, A_i) \in \mathcal{D}$ **do**
  Extract $S_i \leftarrow g(Q_i, A_i)$, regenerating within the fixed budget
  if $V(Q_i, A_i, S_i)$ accepts **then**
    $\mathcal{D}_S \leftarrow \mathcal{D}_S \cup \{(Q_i, S_i)\}$
  end if
end for

Stage 2: On-policy self-distillation
for $k = 1, \ldots, K$ **do**
  Sample a minibatch $\mathcal{B} \subset \mathcal{D}_S$
  $\mathcal{L} \leftarrow 0$ and $N \leftarrow 0$
  for all $(Q, S) \in \mathcal{B}$ **do**
    Sample an on-policy response $\widetilde{Y} \sim p_\theta(\cdot \mid Q)$
    Let $\mathcal{M}$ be the non-prompt, non-padding token positions
    for all $t \in \mathcal{M}$ **do**
      $p_t^T \leftarrow p_{\theta_0}(\cdot \mid Q, S, \widetilde{y}_{<t})$ ▷ frozen teacher
      $p_t^S \leftarrow p_\theta(\cdot \mid Q, \widetilde{y}_{<t})$
      $\mathcal{L} \leftarrow \mathcal{L} + KL(\mathrm{sg}[p_t^T] \| p_t^S)$
      $N \leftarrow N + 1$
    end for
  end for
  Update only θ using $\mathcal{L}/N$
end for

Stage 3: Question-only deployment
Discard the teacher, extractor, and skeletons
return p_θ

Table 6: Training time per step. Relative time is normalized to OPSD ($1.00\times$); lower values indicate less time per step.

Method	Time / step (s)	Relative time
OPSD	41.5	$1.00\times$
SB-OPSD	39.7	$0.96\times$

B Extended Experimental Protocol

B.1 Privileged-information conditions

The bandwidth comparison varies what the frozen teacher sees, while the student remains question-only during training and evaluation. Table 8 specifies the five conditions. B_0 is the unmodified base model.

The default skeleton specification in the experimental configuration uses two to four operations for B_2 and four to eight for B_3. Summary-only retains a high-level description rather than an operation sequence; order shuffling and reversal alter the sequence; cross-sample assignment breaks the match between the question and the skeleton.

B.2 Support–query dataset and evaluation protocol

This evaluation uses 425 support problems and 1,275 query instances. The query set is drawn from the same problem families as the support set, but each query has a different target answer from its corresponding support context. This separation is intended to measure transfer of the update signal

Table 7: Output-token efficiency of SB-OPSD and OPSD. The last column gives SB-OPSD minus OPSD in tokens for the first two rows and the relative change with respect to OPSD for the last row.

Metric	**SB-OPSD (B_2)**	**OPSD**	**Difference**
Median output tokens	19,705	20,761	$-1{,}056$
P90 output tokens	29,724	30,746	$-1{,}022$
Tokens per correct response	43,958	47,384	-7.23%

Table 8: Privileged-information hierarchy used in the analysis. Only B_2 is the proposed SB-OPSD method; the other levels are comparison conditions.

Level	**Teacher-side context**	**Information retained**	**Analytical role**
B_0	Base model	Question-only initial policy; no post-training teacher	Provides the unadapted reference in Table 1
B_1	Functional tags	Coarse operation types without an instance-level plan	Tests whether semantic categories alone are sufficient
B_2	Short result-free skeleton	Ordered operations, branches, and checks without intermediate or final results	Proposed SB-OPSD operating point
B_3	Detailed result-free skeleton	More operations and dependencies, still excluding explicit results	Tests whether additional structural detail improves transfer
B_4	Standard OPSD	Complete verified solution with full structure, intermediate values, conclusions, and final answer	Measures unrestricted solution-conditioned OPSD

rather than reproduction of a privileged answer. All B_1–B_4 conditions start from the same Qwen3-1.7B initialization and are updated only on the support problems. At checkpoints $k \in \{0, 1, 5, 20\}$, each condition is evaluated with one question-only sample per query; no skeleton, solution, or other privileged context is supplied. The four updated conditions and four checkpoints therefore produce $4 \times 4 \times 1{,}275 = 20{,}400$ single-sample evaluations. B_0 remains fixed and is reused as the reference rather than counted as another updated condition.

Correctness is determined by the query verifier. At step 0, every condition matches B_0 at 1,122 correct queries out of 1,275 (88.00%). All gains reported for this evaluation are percentage-point differences from this fixed reference. Table 9 gives the correct-query counts underlying Figure 4.

Table 9: Correct queries out of 1,275 in the support–query evaluation. Each checkpoint uses one question-only response per query.

Condition	Step 0	Step 1	Step 5	Step 20
Base (B_0)	1,122	1,122	1,122	1,122
Functional tags (B_1)	1,122	1,130	1,131	1,129
SB-OPSD (B_2)	1,122	1,133	1,137	1,138
Detailed skeleton (B_3)	1,122	1,128	1,133	1,134
Full-solution OPSD (B_4)	1,122	1,131	1,130	1,131

B.3 SKELETON CONSTRUCTION AND FILTERING

The mathematical-reasoning subset of OpenThoughts supplies verified problem–solution pairs (Guha et al., 2025). A fixed offline extractor maps (Q, A) to an ordered sequence of functional tags and operation instructions. It preserves decisive dependencies while omitting routine algebra, intermediate numerical results, and final conclusions. The student never receives either the solution or the extracted skeleton as input.

Validation checks the sequence schema, permitted tags, length bounds, and grounding of symbols and named entities in the question. It rejects explicit answer strings, result-bearing equations, and answer-equivalent expressions, and requires

$$\mathrm{Nums}(S) \setminus \mathrm{Nums}(Q) = \emptyset.$$

Lexical copying from the solution is also restricted. Extraction permits at most three attempts per example; examples without an accepted skeleton are removed from the skeleton training set.

B.4 METRICS

Average@12 is the fraction of correct generations across 12 samples per problem. Pass@12 counts a problem as solved if any of its samples is correct; majority-vote@12 evaluates its most frequent extracted answer.

Reported means and standard deviations summarize variation across three independent training seeds. The support–query evaluation in Appendix B.2 uses its separately specified single-sample protocol.

C WORKED EXAMPLE OF THE PRIVILEGED-INFORMATION HIERARCHY

The following illustrative example compares the information permitted at B_0–B_4. All levels use the same question. For the trained conditions B_1–B_4, only the teacher-side context changes, while the student receives the question alone. B_0 instead denotes the unmodified question-only base model, with no post-training teacher.

Shared question.

> Consider the quadratic equation in x,
>
> $$x^2 - 2(a+1)x + a^2 + 1 = 0.$$
>
> Find all real values of a for which the equation has two distinct positive real roots.

The blocks below specify the additional teacher context, appended to this shared question. Functional labels use the vocabulary described in Section 3.2; mathematical tools such as the discriminant and Vieta's formulas appear only in the operation descriptions at B_2 and above.

B_0: question-only base model. The base model receives only the shared question. No solution, functional tags, or skeleton is supplied, and no distillation update is performed.

B_1: functional tags only.

> Planning; retrieval; inference; verification; summarization.

These labels indicate broad reasoning functions, not an instance-specific sequence of operations. They provide neither mathematical expressions nor intermediate results.

B_2: short result-free skeleton.

1. **Inference:** Use the discriminant to require distinct real roots.
2. **Retrieval:** Express the root sum and product using Vieta's formulas.

3. **Inference:** Require positive root sum and product; intersect the parameter conditions.
4. **Verification:** Check boundary cases against the distinctness and positivity requirements.

This four-operation skeleton retains the order of the main checks without providing the simplified discriminant, evaluated root sum or product, solved parameter inequalities, or final range.

B_3: detailed result-free skeleton.

1. **Planning:** Separate real-root existence, distinctness, and positivity requirements.
2. **Inference:** Compute the discriminant; require strict positivity for distinct real roots.
3. **Retrieval:** Use Vieta's formulas to express the root sum and product.
4. **Inference:** Given real roots, require positive product to ensure matching signs.
5. **Inference:** Require positive root sum to exclude two negative roots.
6. **Inference:** Solve each parameter condition and intersect the resulting sets.
7. **Verification:** Exclude discriminant-zero boundaries because the roots must remain distinct.
8. **Verification:** Substitute the candidate parameter range into every required condition.

Relative to B_2, this eight-operation skeleton makes the logical dependencies and sign checks more explicit. It still withholds all instance-specific computed results and the final answer.

B_4: complete worked solution. Let the roots be x_1 and x_2. Two distinct real roots require

$$\begin{aligned}\Delta &= [-2(a+1)]^2 - 4(a^2+1) \\ &= 4(a+1)^2 - 4(a^2+1) = 8a > 0,\end{aligned}$$

which is equivalent to $a > 0$. By Vieta's formulas,

$$x_1 + x_2 = 2(a+1), \qquad x_1 x_2 = a^2 + 1.$$

Once the roots are real, both are positive if and only if their sum and product are positive. Here,

$$x_1 + x_2 > 0 \iff a > -1, \qquad x_1 x_2 = a^2 + 1 > 0 \quad \text{for every } a \in \mathbb{R}.$$

Intersecting these conditions gives $a > 0$. Conversely, every $a > 0$ makes the discriminant, root sum, and root product positive, so it indeed yields two distinct positive real roots. At $a = 0$, the equation becomes $(x-1)^2 = 0$ and the roots coincide, so the boundary is excluded. Therefore,

$$\boxed{a \in (0, +\infty)}.$$

Information boundary. Generic instructions such as requiring a positive discriminant or checking root signs are permitted in B_2 and B_3. In contrast, the evaluated expressions $\Delta = 8a$ and $x_1 + x_2 = 2(a+1)$, the intermediate condition $a > -1$, and the final conclusion $a > 0$ are reserved for B_4. Thus, the short and detailed skeletons describe what to do without revealing the instance-specific results of those operations.