ABSORBABILITY-GUIDED COT COMPRESSION FOR EFFICIENT REASONING

Anonymous authors
Paper under double-blind review

ABSTRACT

Reasoning models often generate longer chains of thought (CoTs) than a task requires, increasing inference cost. Accuracy-preserving compression requires the current model to absorb the compressed supervision into its reasoning capability. Existing methods choose compression levels from trajectory or task properties without evaluating this model-dependent condition. We introduce **Absorbability-Guided CoT Compression** (AGC^2), which uses local adaptation to select model-absorbable supervision. For each batch, AGC^2 measures reference-trajectory quality and evaluates candidate token budgets through short SFT updates on a fresh copy of the current model. When reference trajectories are reliable, it selects the shortest budget whose adapted accuracy remains within a tolerance of the reference accuracy. When they are unreliable, it increases the budget until the adapted model reaches an absolute accuracy threshold. Across DeepSeek-R1-Distill-Qwen models at 7B, 14B, and 32B scales on GSM8K, MATH-500, and AIME2024, AGC^2 reduces response length by 19.2%–67.3% while matching or exceeding original-model accuracy in seven of nine model–benchmark settings. These results establish model absorbability as the governing criterion for accuracy-preserving CoT compression: the retained budget matches the amount of supervision the current model absorbs into its reasoning capability.

1 INTRODUCTION

Large reasoning models (LRMs) solve difficult problems by generating chains of thought (CoTs) before producing final answers. This test-time computation supports exploration, error correction, and verification (Wei et al., 2022; Snell et al., 2025), but it increases inference cost (Yi et al., 2025; Liu et al., 2026c; Yang et al., 2026). Because inference cost scales with the number of generated tokens, a shorter CoT lowers the cost of every query. Efficient reasoning therefore requires these models to generate concise CoTs that preserve answer accuracy. Training on compressed CoTs provides a direct route to shorter responses, which raises the question of how much of a long trajectory to compress. Existing efficient reasoning methods construct concise CoTs by compression from long trajectories and use them as SFT supervision, either alone or alongside the original long CoTs (Kang et al., 2025; Liu et al., 2024; Li et al., 2025). They choose compression levels from properties of the CoT or task (Feng et al., 2024; Yi et al., 2025; Gao et al., 2025).

Despite promising progress made by the current efficient reasoning methods, they overlook an issue when doing SFT, which is whether the model can truly learn the ability of compressed CoT. Early-stopping methods truncate a CoT at an intermediate point and prompt the same model to produce a trial answer, yielding shorter trajectories (Yang et al., 2026). Complementary work describes conditions under which training moves explicit CoT steps into hidden-state computation (Deng et al., 2023; 2024; Huang et al., 2026). These results show that some reasoning need not remain explicit, but they do not specify which compressed CoT the current SFT model can absorb at its present state. This is the supervision-selection problem addressed in this work.

Removing a removable step leaves the answer correct, whereas removing a step that the computation needs changes it. A high compression ratio alone does not identify an absorbable supervision signal. We illustrate this boundary in Figure 1. We use three sizes of DeepSeek-R1-Distill-Qwen to generate trajectories for AIME2024 and compress these CoTs with GPT, TokenSkip (Xia et al., 2025), and StepEntropy (LI et al., 2026). We then remove the step that derives the answer from each

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.

compressed CoT and let the original model complete it. Every model loses accuracy when it reasons to the answer from the compressed CoT. Therefore, these compressed CoTs are not all absorbable supervision for the evaluated models. The result establishes that the appropriate compression level depends on the model state, not on the compression ratio alone.

To address this issue, we introduce **Absorbability-Guided CoT Compression** (AGC^2), a training method that selects compressed-CoT supervision through model absorbability. For each batch, AGC^2 proposes token budgets, constructs the corresponding compressed CoTs, adapts a fresh copy of the current model with short SFT updates, and measures answer accuracy. High-quality reference batches trigger a search toward shorter budgets, retaining the shortest budget whose adapted accuracy remains within tolerance of the reference accuracy. Low-quality reference batches trigger a search toward longer budgets, retaining the first budget whose adapted model reaches an absolute accuracy threshold. Training continues from the adapted model associated with the selected budget. By testing candidates through local adaptation from a common model state, AGC^2 makes absorbability the criterion that links compression length to the supervision the model absorbs into its reasoning capability.

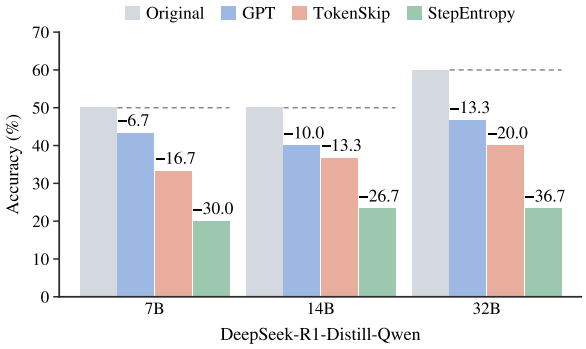


Figure 1: Accuracy (%) of uncompressed chain-of-thought (Original) and three compression methods on Deepseek-R1-Distill-Qwen 7B, 14B and 32B models. The accuracy of the compressed CoTs is calculated with the answer-derivation step removed. The number above each compressed bar is the absolute drop in accuracy relative to the Original baseline. The CoTs compressed with all three methods cause a decrease in accuracy.

We evaluate AGC^2 on DeepSeek-R1-Distill-Qwen models at 7B, 14B, and 32B scales across GSM8K, MATH-500, and AIME2024. Across the nine model–benchmark pairs, the trained models reduce response length by 19.2%–67.3% while matching or exceeding the original-model accuracy in seven settings.

We make three contributions.

- We formulate model absorbability as the criterion for compressed-CoT supervision. Local post-adaptation answer accuracy determines whether a candidate budget retains enough explicit reasoning for the current model, with the criterion conditioned on reference-trajectory quality.
- We propose AGC^2, which searches toward shorter or longer budgets according to reference-trajectory quality, evaluates every candidate through local adaptation from the current model state, and continues training from the selected adapted model.
- We evaluate AGC^2 on three mathematical benchmarks, GSM8K, MATH-500, and AIME2024. The trained models reduce response length by 19.2%–67.3% while matching or exceeding the original-model accuracy in seven of nine settings, with the largest accuracy gains on MATH-500 and AIME2024.

2 Related Work

Compressed CoTs as training signals for efficient reasoning. A line of work constructs concise targets from long trajectories and trains models to emit them. Where compression is performed by a separate teacher, compressor, or larger model, the trained model is a student in the distillation sense. C3oT (Kang et al., 2025) compresses long trajectories, Liu et al. (2024) iteratively shorten them, and CoT-Valve (Ma et al., 2025b) uses a controllable parameter direction; TokenSkip (Xia et al., 2025), SPIRIT (Cui et al., 2025), Skip-Thinking (Chen et al., 2025), TokenSqueeze (Zhang et al., 2025), and Pru-CoT (Liu et al., 2026a) choose retained content with token- or step-level signals.

These methods assess trajectory properties or apply fixed control rules. AGC^2 instead selects compressed candidates through a local update of the model being trained, with the selection anchored to reference-trajectory quality rather than a token-importance score. The criterion is the model's post-adaptation accuracy, which directly measures whether the current model absorbs the compressed supervision. This distinction separates an answer-correct compressed CoT from a compressed CoT that is absorbable as a training target.

Length-aware and adaptive post-training. Length-aware post-training includes ShorterBetter (Yi et al., 2025), REO-RL (Gao et al., 2025), LEASH (Li et al., 2026), LC-R1 (Cheng et al., 2026), and AttnPO (Nie et al., 2026). These methods change the model's preference over response length through rewards, constraints, or step-level credit assignment. Rakotonirina et al. (2026) instead combine supervised fine-tuning with reinforcement learning under an adaptive length penalty. Adaptive-control methods such as Ada-R1 (Luo et al., 2025), TH2T (Liu et al., 2026c), and ACPO (Cheng et al., 2025) condition reasoning depth on an input, an evolving trajectory, or an estimated difficulty; Lin et al. (2025) likewise control a model's thinking speed. AGC^2 adapts the granularity of the SFT target during training according to the supervision that the current model absorbs, rather than selecting a reasoning mode at inference.

CoT truncation and implicit computation. Early-exit methods such as DEER (Yang et al., 2026), NEAT (Liu et al., 2026b), S-GRPO (Dai et al., 2025), and Step-GRPO (Chen et al., 2026) stop an ongoing CoT once a prefix meets their sufficiency criterion. Xiang et al. (2026) similarly use sufficiency assessment to decide when to exit. Distillation studies by Deng et al. (2023), Deng et al. (2024), and Huang et al. (2026) further document cases in which training replaces emitted intermediate computation with computation in model states. These results show that effective reasoning does not require every intermediate step to remain explicit. AGC^2 builds on this finding by using local adaptation to select compressed CoTs that the current model absorbs as effective training targets.

3 Absorbable Compressed-CoT Supervision

Efficient reasoning requires compressed supervision that the current model can absorb into its reasoning capability. We therefore formulate compression as a supervision-selection problem: the compression level follows the model state and the quality of the reference trajectories, not trajectory length or compression ratio alone.

Task and reference trajectories. Let $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$ be a corpus of N problem–gold-answer pairs. A reference generator T returns a long CoT c_i^L and an extracted final answer $\hat{y}_i$ for each problem, $T(x_i) = (c_i^L, \hat{y}_i)$, and the generated trajectories form $\mathcal{D}_{\mathrm{long}} = \{(x_i, c_i^L, \hat{y}_i, y_i)\}$. To keep erroneous data out of training, we retain in $\mathcal{D}_{\mathrm{ref}} = \{(x_i, c_i^L, y_i) \mid \hat{y}_i = y_i\}$ only those generated trajectories whose extracted answer matches the gold answer.

Compressed-CoT candidates. Let $\gamma = (B, b)$ denote a short-CoT transformation, where B is a compression operator and b is its control signal. The operator maps a problem and its long CoT to a candidate $c_{i,\gamma}^S = B(x_i, c_i^L; b)$. The operator B covers compressors, metric-guided token- or step-level pruning rules, and early-stopping truncation rules; the signal b specifies a target token budget, a metric threshold, or a stopping condition. Applying γ to the reference trajectories gives $\mathcal{D}_{\mathrm{short}}^{\gamma} = \{(x_i, c_{i,\gamma}^S, y_i)\}$. The requested budget is a search variable: for a batch $\mathcal{B}_t \subset \mathcal{D}_{\mathrm{ref}}$ it need not equal the realized length, and the realized CoT-token reduction determines the compression that the target actually supplies.

Model-state-conditioned absorbability. Absorbability measures how the supervision carried by a compressed CoT changes the answer accuracy of the current model after local adaptation. A compressed CoT is absorbable when that adaptation restores correct answers, so the score is measured after local adaptation from the current model state. For a model with parameters θ and an SFT loss $\mathcal{L}$, let $\mathcal{U}_m(\theta; \mathcal{L})$ denote m local SFT updates. For a candidate γ, the locally adapted parameters are

$$\theta^{\gamma} = \mathcal{U}_m\big(\theta; \mathcal{L}_{\gamma}^{\mathrm{SFT}}\big)\,, \tag{1}$$

where $\mathcal{L}_{\gamma}^{\mathrm{SFT}}$ is computed from the corresponding compressed targets. On an evaluation set $\mathcal{E}$, let M_θ denote the model with parameters θ and $M_\theta(x_i)$ the final answer that it returns for x_i. We define the absorbability score of γ for model state θ as the answer accuracy of the adapted model,

$$\mathrm{Abs}(\theta, \gamma; \mathcal{E}) = A_{\mathcal{E}}(\theta^{\gamma}) = \frac{1}{|\mathcal{E}|} \sum_{(x_i, y_i) \in \mathcal{E}} \mathbb{I}[M_{\theta^{\gamma}}(x_i) = y_i], \tag{2}$$

where $\mathbb{I}[\cdot]$ denotes the indicator of a true condition, and $L_{\mathcal{E}}(\theta^{\gamma})$ denotes the mean number of generated response tokens.

The criterion connects compression to both supervision and inference: Abs determines whether the current model absorbs a compressed target, and $L_{\mathcal{E}}$ records the resulting inference behavior. AGC^2 selects candidates by this criterion and continues training from the adapted parameters of the selected candidate.

4 Absorbability-Guided CoT Compression

4.1 Overview

AGC^2 repeats the selection of compressed-CoT supervision at every training batch. In each iteration, AGC^2 first measures the quality of the reference trajectories, constructs compressed trajectories at candidate budgets, and adapts the current model with the compressed CoTs as supervision. For each candidate budget, AGC^2 measures the accuracy of the adapted model on the current batch to determine whether the supervision is absorbable. The quality of the reference trajectories determines the direction of the search: reliable trajectories admit shorter budgets, whereas unreliable trajectories require longer budgets before the adapted model reaches the required accuracy. Figure 2 illustrates the pipeline of our approach, and Algorithm 1 states the full procedure.

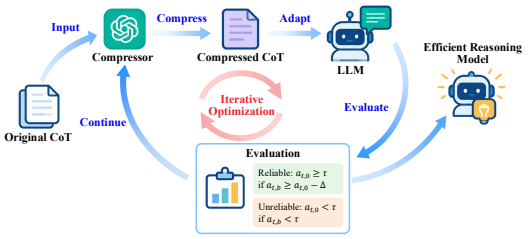


Figure 2: Overview of AGC^2: the compressor proposes a candidate budget, the model adapts to it, and reference quality selects the branch.

4.2 Reference Quality and Candidate Absorbability

Let $\mathcal{B}_t^{\mathrm{long}} = \{(x_i, c_i^L, \hat{y}_i, y_i)\}_{i=1}^{n_t} \subset \mathcal{D}_{\mathrm{long}}$ be the reference-trajectory batch of n_t examples at training step t. AGC^2 first measures the quality of the reference trajectories as

$$a_{t,0} = \frac{1}{|\mathcal{B}_t^{\mathrm{long}}|} \sum_{(x_i, c_i^L, \hat{y}_i, y_i) \in \mathcal{B}_t^{\mathrm{long}}} \mathbb{I}[\hat{y}_i = y_i], \tag{3}$$

where $a_{t,0}$ is the accuracy of the reference trajectories. The answer-verified sub-batch is

$$\mathcal{B}_t^{\mathrm{ref}} = \{(x_i, c_i^L, y_i) \mid (x_i, c_i^L, \hat{y}_i, y_i) \in \mathcal{B}_t^{\mathrm{long}},\ \hat{y}_i = y_i\} \subset \mathcal{D}_{\mathrm{ref}}. \tag{4}$$

Given the compressor B and a requested token budget b, let $\gamma_b = (B, b)$ denote the corresponding compression configuration, and let B produce the compressed CoT c_{i,γ_b}^S for each $(x_i, c_i^L, y_i) \in \mathcal{B}_t^{\mathrm{ref}}$. Let $\theta_{t-1}^{\star}$ denote the parameters selected after batch $t-1$, and $\theta_0^{\star}$ the initial model parameters. We examine in two steps whether the compressed supervision at budget b is absorbable for the current model $M_{\theta_{t-1}^{\star}}$. First, the compressed trajectories of $\mathcal{B}_t^{\mathrm{ref}}$ supervise the current model, with loss

$$\mathcal{L}_{t,b}^{\mathrm{SFT}}(\theta) = -\frac{1}{Z_{t,b}} \sum_{(x_i, c_i^L, y_i) \in \mathcal{B}_t^{\mathrm{ref}}} \sum_{r=1}^{|c_{i,\gamma_b}^S|} \log p_\theta\big(c_{i,\gamma_b,r}^S \mid x_i, c_{i,\gamma_b,<r}^S\big), \tag{5}$$

where p_θ is the model's next-token distribution, r indexes output tokens, $c_{i,\gamma_b,<r}^S$ is the prefix preceding token r, and $Z_{t,b} = \sum_i |c_{i,\gamma_b}^S|$. Every candidate budget is then adapted from the same pre-batch

state $\theta^\star_{t-1}$ by m local SFT updates:

$$\theta_{t,b} = \mathcal{U}_m\big(\theta^\star_{t-1}; \mathcal{L}^{\mathrm{SFT}}_{t,b}\big)\,. \tag{6}$$

Second, the adapted model answers the questions of the full reference batch, and the resulting accuracy gives the absorbability of the compressed supervision at budget b, as defined in Equation 2:

$$\mathrm{Abs}(\theta^\star_{t-1}, \gamma_b; \mathcal{B}^{\mathrm{long}}_t) = a_{t,b} = \frac{1}{n_t} \sum_{(x_i, c^L_i, \hat{y}_i, y_i) \in \mathcal{B}^{\mathrm{long}}_t} \mathbb{I}[M_{\theta_{t,b}}(x_i) = y_i], \tag{7}$$

where $M_{\theta_{t,b}}(x_i)$ is the final answer generated for x_i by the adapted model.

4.3 Branch-Specific Budget Search

The absorbability score of Equation 2 characterizes a single budget; AGC^2 selects the budgets to test from two length scales. The first scale is the typical reference length b_{ref}, the 95th percentile of reference-CoT lengths in the training data, which keeps a few exceptionally long trajectories from setting the global budget. The second scale is the batch length

$$b_{t,0} = \max_{(x_i, c^L_i, \hat{y}_i, y_i) \in \mathcal{B}^{\mathrm{long}}_t} |c^L_i| \tag{8}$$

the maximum long-CoT length in the current batch, which serves as the starting budget of the search.

The reference accuracy $a_{t,0}$, measured against the reference-quality threshold τ, selects the direction of the search: trajectories with $a_{t,0} \geq \tau$ are reliable, and AGC^2 searches toward shorter budgets; trajectories with $a_{t,0} < \tau$ are unreliable, and AGC^2 searches toward longer budgets. At iteration j, the two schedules are

$$b^{\mathrm{short}}_{t,j} = \frac{b_{t,0} j^{\mathrm{short}}_0}{j^{\mathrm{short}}_0 + j}, \qquad b^{\mathrm{long}}_{t,j} = \frac{b_{t,0}(j^{\mathrm{long}}_0 + j)}{j^{\mathrm{long}}_0}, \tag{9}$$

where $j^{\mathrm{short}}_0 = \max(1, b_{\mathrm{ref}}/b_{t,0})$ and $j^{\mathrm{long}}_0 = \max(1, b_{t,0}/b_{\mathrm{ref}})$ set their pace from the ratio between the two scales. The short schedule decays hyperbolically, so its largest cut falls in the first iteration, while the long schedule adds $\min(b_{t,0}, b_{\mathrm{ref}})$ tokens per iteration. The algorithm projects each proposed budget onto the allowed integer range and skips budgets already tried.

Each schedule is paired with an acceptance rule on the absorbability score $a_{t,b}$. In the reliable branch, a candidate is absorbable when its post-adaptation accuracy stays within a tolerance Δ of the reference accuracy, $a_{t,b^{\mathrm{short}}_{t,j}} \geq a_{t,0} - \Delta$; AGC^2 keeps the shortest absorbable budget and stops after $h \geq 1$ consecutive tolerance failures. In the unreliable branch, the reference accuracy is itself below τ, so the criterion is absolute: AGC^2 lengthens the budget until the adapted model reaches $a_{t,b^{\mathrm{long}}_{t,j}} \geq \tau$, and accepts the first budget that does. Both branches use the same absorbability score with different anchors: the reliable branch preserves the quality of a trustworthy reference batch while shortening the supervision, whereas the unreliable branch restores an absolute accuracy level before accepting a candidate.

Training continues from the adapted parameters of the accepted budget, or retains the pre-batch parameters when the search ends without an accepted candidate:

$$\theta^\star_t = \begin{cases} \theta_{t,b^\star_t}, & \text{if the search selects } b^\star_t, \\ \theta^\star_{t-1}, & \text{otherwise.} \end{cases} \tag{10}$$

Appendix A.1 lists the complete batch-wise procedure (Algorithm 1), including candidate construction, independent local adaptation, absorbability evaluation, and parameter selection.

5 Experiments

We evaluate AGC^2 by jointly measuring model response length and answer accuracy, then examine how its search criteria shape the resulting accuracy–length trade-off.

5.1 Experimental Setup

Models. We study DeepSeek-R1-Distill-Qwen models with 7B, 14B, and 32B parameters (DeepSeek-AI et al., 2025).

Data and Metrics. The training corpus combines 7,473 GSM8K training problems (Cobbe et al., 2021) and 8,000 MATH problems (Hendrycks et al., 2021). We reserve 128 examples for validation and train on the remaining 15,345 examples. The original weights of each model generate the long CoTs used as reference trajectories. As described in Section 4, the answer-verified trajectories supply compressed-CoT supervision, while the full reference batch guides budget selection.

We evaluate the final models on three mathematical reasoning benchmarks: GSM8K (Cobbe et al., 2021), MATH-500 (Hendrycks et al., 2021), and AIME2024 (Hugging Face H4, 2024). We report answer accuracy, mean response length, and relative length change against the corresponding original model.

Implementation Details. For all main results, the search uses the reference-quality threshold $\tau = 0.90$, the tolerance $\Delta = 0$, $m = 2$ local SFT updates per candidate, and the tolerance-failure allowance $h = 1$. We use LoRA to finetune the models with the compressed CoTs; the remaining settings, including the budget-proposal cap and the LoRA configuration, are reported in Appendix A.

Baselines. We compare AGC^2 with D-Prompt, TH2T (Liu et al., 2026c), NoThinking (Ma et al., 2025a), TokenSkip (Xia et al., 2025), CoT-Valve (Ma et al., 2025b), and AdaCtrl (Huang et al., 2025). The baseline results are from Liu et al. (2026c).

5.2 Main Results

Table 1 compares AGC^2 with the original models and the baselines at the three model scales. The best and second-best values in each column are shown in bold and underlined, respectively.

Task-dependent accuracy–length trade-offs. AGC^2 shortens mean response length in all nine model–benchmark settings while matching or improving Original accuracy in seven. The resulting trade-offs differ across benchmarks: accuracy declines occur only for the 7B and 14B models on GSM8K, whereas AGC^2 improves accuracy at all three scales on MATH-500 and AIME2024. This pattern reflects the selection rule: AGC^2 retains a shorter candidate only when its locally adapted model satisfies the branch-specific criterion, so explicit reasoning is allocated per task rather than against a uniform response-length target.

More compression is not always better. Shorter responses do not always yield the preferable accuracy–length trade-off: on MATH-500 with the 14B model and AIME2024 with the 32B model, TH2T produces shorter responses while AGC^2 attains higher accuracy. AGC^2 therefore occupies a different point on the accuracy–length frontier, because its objective is the shortest candidate that satisfies the local criterion rather than the fewest tokens.

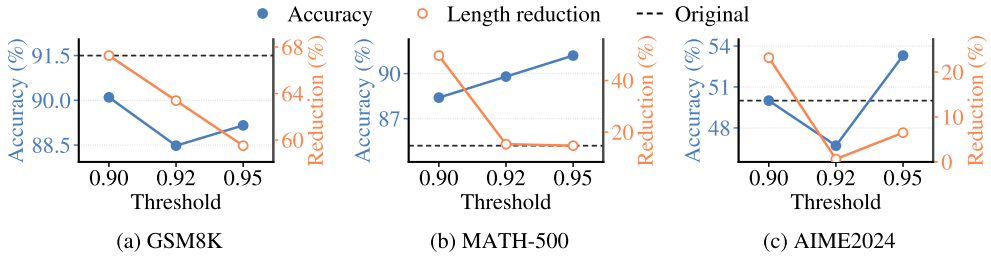
5.3 Analysis of AGC^2

5.3.1 Sensitivity to Search Criteria

Accuracy threshold τ. The threshold τ selects the search branch for each batch: reference accuracy at or above τ triggers shortening; otherwise, AGC^2 lengthens the budget. Figure 3 shows how τ affects response accuracy and length. Raising τ makes AGC^2 more conservative: fewer batches enter the shortening branch, and response length increases on all three benchmarks. This loss of compression does not yield a uniform accuracy benefit: GSM8K accuracy decreases, and $\tau = 0.92$ is both longer and less accurate than $\tau = 0.90$ on AIME2024. Although MATH-500 and AIME2024 improve at $\tau = 0.95$, the gain comes with substantially longer responses. Thus, increasing τ consistently weakens compression without producing a uniformly better accuracy–efficiency trade-off. Appendix B.1 reports results under other Δ settings, which show the same benchmark dependence.

Table 1: Main results for DeepSeek-R1-Distill-Qwen at 7B, 14B, and 32B. Baseline results are from Liu et al. (2026c). Relative length change is computed with respect to the original model at the same scale; the best and second-best values in each column are shown in bold and underlined, and each scale forms its own comparison.

Method	GSM8K			MATH-500			AIME2024		
	Answer Accuracy	Response Length	Length Change	Answer Accuracy	Response Length	Length Change	Answer Accuracy	Response Length	Length Change
DeepSeek-R1-Distill-Qwen-7B									
Original	91.5	1,057	–	85.2	2,857	–	50.0	10,570	–
D-Prompt	90.6	906	-14.3%	86.0	2,726	-4.6%	46.7	10,209	-3.4%
NoThinking	88.5	243	-77.0%	79.4	**699**	**-75.5%**	40.0	**4,134**	**-60.9%**
TokenSkip	88.0	638	-39.6%	80.2	2,109	-26.2%	40.0	5,559	-47.4%
CoT-Valve	87.9	**235**	**-77.8%**	78.6	747	-73.9%	43.3	5,871	-44.5%
AdaCtrl	91.0	349	-67.0%	74.0	3,196	+11.9%	21.3	16,889	+59.8%
TH2T	**92.9**	275	-74.0%	86.8	1,772	-38.0%	**50.8**	7,291	-31.0%
AGC^2	90.1	346	-67.3%	**88.4**	1,438	-49.7%	50.0	8,117	-23.2%
DeepSeek-R1-Distill-Qwen-14B									
Original	93.6	584	–	86.6	2,306	–	50.0	9,900	–
D-Prompt	93.9	606	+3.8%	86.6	2,297	-0.4%	50.0	10,408	+5.1%
NoThinking	88.4	**206**	**-64.7%**	76.0	**612**	**-73.5%**	43.3	**3,735**	**-62.3%**
TokenSkip	89.3	314	-46.2%	73.2	1,356	-41.2%	46.7	6,772	-31.6%
CoT-Valve	92.7	526	-9.9%	84.2	2,111	-8.5%	40.0	7,027	-29.0%
TH2T	**94.1**	246	-57.9%	86.4	1,343	-41.8%	52.5	6,973	-29.6%
AGC^2	93.4	395	-32.4%	**87.6**	1,509	-34.6%	**53.3**	6,292	-36.4%
DeepSeek-R1-Distill-Qwen-32B									
Original	95.6	717	–	87.2	2,357	–	60.0	9,605	–
D-Prompt	95.6	663	-7.5%	86.8	2,198	-6.7%	56.7	9,445	-1.7%
NoThinking	93.9	**228**	**-68.2%**	80.2	**659**	**-72.0%**	53.3	**3,208**	**-66.6%**
TokenSkip	96.5	423	-41.0%	79.8	1,567	-33.5%	50.0	5,856	-39.0%
CoT-Valve	96.1	781	+8.9%	85.3	2,263	-4.0%	53.3	7,997	-16.7%
TH2T	**97.0**	263	-63.3%	88.0	1,733	-26.5%	58.8	7,081	-26.3%
AGC^2	95.9	383	-46.6%	**89.0**	1,441	-38.9%	**66.7**	7,760	-19.2%

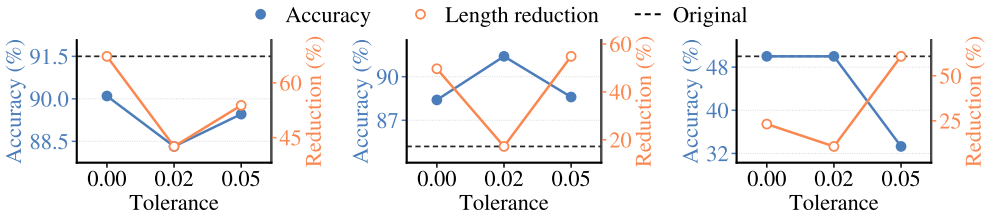


(a) GSM8K (b) MATH-500 (c) AIME2024

Figure 3: Sensitivity to the reference-quality threshold τ under accuracy-constrained search, with $\Delta = 0$ and $h = 1$ fixed. Each panel covers one benchmark for $\tau \in \{0.90, 0.92, 0.95\}$: the left axis reports answer accuracy and the right axis the length reduction relative to the original 7B model, whose accuracy is marked by the dashed line.

Accuracy tolerance Δ. Once a batch enters the shortening branch, Δ sets the largest accuracy drop allowed relative to its reference trajectory. Figure 4 shows how Δ affects response accuracy and length. On AIME2024, $\Delta = 0.05$ gives much shorter responses but lower final accuracy. On MATH-500, the same setting is shorter while accuracy increases slightly, and on GSM8K both looser tolerances lengthen responses and reduce accuracy. Thus, Δ is a local acceptance criterion rather than a global compression target. The results at $\tau = 0.95$ in Appendix B.2 show the same benchmark dependence.

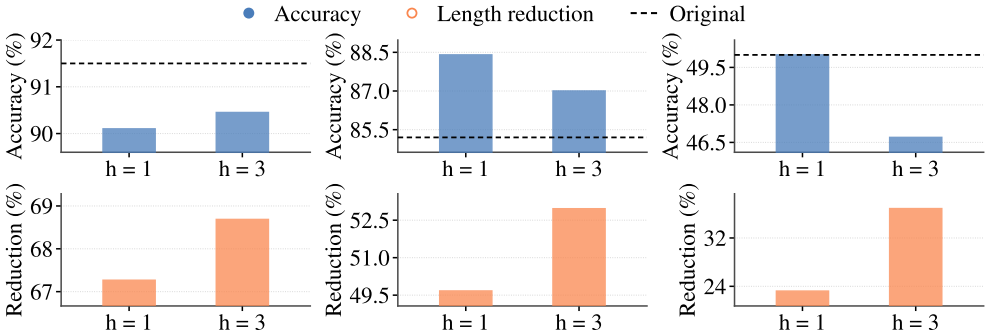
Tolerance-failure patience h. The allowance h determines how many failed short-budget probes are permitted before shortening stops. A larger h lets the search continue beyond its first infeasi-



(a) GSM8K (b) MATH-500 (c) AIME2024

Figure 4: Sensitivity to the accuracy tolerance Δ under accuracy-constrained search, with $\tau = 0.90$ and $h = 1$ fixed. Each panel covers one benchmark for $\Delta \in \{0, 0.02, 0.05\}$: the left axis reports answer accuracy and the right axis the length reduction relative to the original 7B model, whose accuracy is marked by the dashed line.

ble candidate. Figure 5 shows how h affects response accuracy and length. With $h = 3$, response length decreases on all three benchmarks. GSM8K also gains accuracy, whereas MATH-500 and AIME2024 trade accuracy for this extra compression. Thus, h controls search depth rather than enforcing a fixed degree of compression. Appendix B.3 shows that this behavior remains task dependent under other Δ settings.



(a) GSM8K (b) MATH-500 (c) AIME2024

Figure 5: Sensitivity to the tolerance-failure allowance h under accuracy-constrained search, with $\tau = 0.90$ and $\Delta = 0$ fixed. Each column covers one benchmark for $h \in \{1, 3\}$: the upper row reports answer accuracy and the lower row the length reduction relative to the original 7B model; the dashed line marks the accuracy of the original model.

5.3.2 Search Strategy

Accuracy feedback vs. Fixed rounds. Figure 6 compares accuracy-constrained search with Fixed, which applies three shortening rounds to every batch. On MATH-500 and AIME2024, Fixed substantially reduces response length but incurs marked accuracy losses, with the largest degradation on AIME2024. On GSM8K, Fixed is both longer and less accurate. In contrast, accuracy-constrained search selects its final budget using the local acceptance criterion, retaining additional tokens when further shortening violates this constraint. The resulting gap is largest on the two harder benchmarks.

Independent candidate updates vs. cumulative accepted updates. Independent candidate updates evaluate every budget from the same pre-batch model state $\theta^{\star}_{t-1}$, whereas cumulative accepted updates initialize each probe from the last accepted candidate, which makes the within-batch search path dependent. Both rules carry the final accepted state to the next batch.

Under External compression, cumulative updates shift the benchmarks toward different accuracy–length trade-offs: they shorten GSM8K and AIME2024, lengthen MATH-500, and lower accuracy on all three benchmarks (Figure 7). Because each probe starts from an altered state rather than the common pre-batch reference, the search does not follow a uniform shortening direction, and self

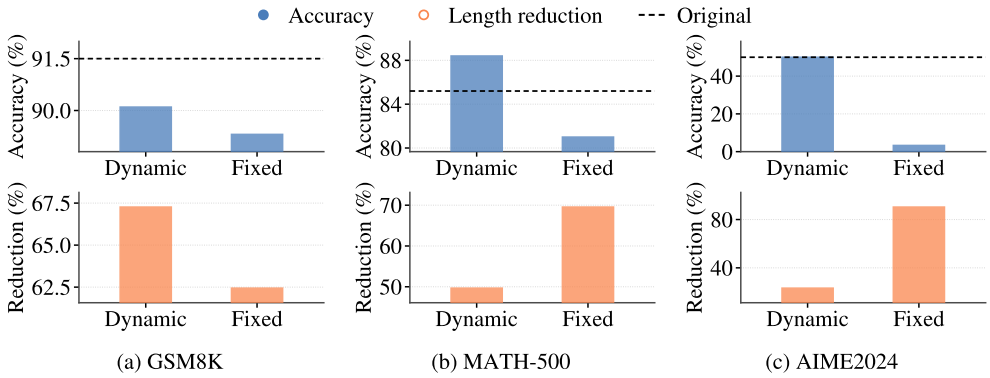


Figure 6: Accuracy-constrained search (Dynamic) versus fixed-round shortening (Fixed) under $(\tau, \Delta, h) = (.90, 0, 1)$. Each column covers one benchmark: the upper row reports answer accuracy and the lower row the length reduction relative to the original 7B model; the dashed line marks the accuracy of the original model.

compression shows the same task-dependent changes. These results favor independent candidate updates as the more stable default.

External vs. Self compressor. External uses GPT-4o-mini to construct candidate CoTs, whereas Self uses the frozen original 7B model. With independent updates, External is shorter on all three benchmarks, more accurate on GSM8K, and less accurate on MATH-500 and AIME2024; the same benchmark-dependent pattern holds with cumulative updates. Compressor choice therefore interacts with both the benchmark and the update rule.

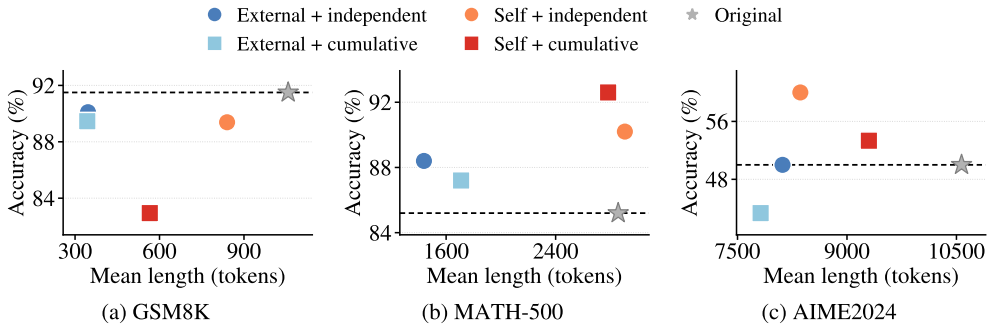


Figure 7: Results on the 7B model for four combinations of compressor choice and within-batch candidate updates under accuracy-constrained search with $(\tau, \Delta, h, m, K) = (.90, 0, 1, 2, 3)$. Each marker is one configuration, placed by its mean response length and answer accuracy: colour and marker shape distinguish the four compressor–update combinations, and the grey star marks the original model.

6 Conclusion

We introduced AGC^2, which selects the compressed-CoT supervision of a batch from the compression that the current model absorbs: it adapts candidate budgets locally from a common model state and keeps a budget only when the adapted model satisfies the corresponding accuracy criterion. Across DeepSeek-R1-Distill-Qwen models at three scales and three mathematical benchmarks, the trained models generate 19.2%–67.3% fewer response tokens while matching or exceeding the original-model accuracy in seven of nine settings, which indicates that effective reasoning compression requires retaining the explicit reasoning that the answer depends on.

AI USE STATEMENT

We did not use generative AI tools for any task with required disclosure: no synthetic data generation, theoretical modelling, formulation of mathematical claims or proofs, hypothesis generation, design of or feedback on the research methodology or experiments, method implementation, dataset cleaning or reformatting, qualitative data analysis, or interpretation of results. We used generative AI tools for two tasks with recommended disclosure: editing the manuscript for clarity, grammar, and style, and searching for relevant literature. Every AI-assisted edit was reviewed and accepted by the authors, and every reference identified through AI-assisted search was verified against its published record before it was cited. We take responsibility for the final content of this work, including its text and claims.

REPRODUCIBILITY STATEMENT

Section 5 describes the training corpus, the three public benchmarks, and the evaluation protocol, and lists the baselines whose reported numbers we compare against. Appendix A gives the hyperparameters of every run, including the LoRA configuration and the search parameters, and Appendix A.1 states the complete batch-wise procedure of AGC^2 (Algorithm 1). The models are the public DeepSeek-R1-Distill-Qwen checkpoints at 7B, 14B, and 32B scales, and GSM8K, MATH-500, and AIME2024 are public datasets.

REFERENCES

Benteng Chen, Weida Wang, Shufei Zhang, Mingbao Lin, and Min Zhang. Step-GRPO: Internalizing dynamic early exit for efficient reasoning. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 21710–21724. Association for Computational Linguistics, 2026. doi: 10.18653/v1/2026.acl-long.990. URL https://aclanthology.org/2026.acl-long.990/.

Xiaoshu Chen, Sihang Zhou, Ke Liang, Xiaoyu Sun, and Xinwang Liu. Skip-thinking: Chunk-wise chain-of-thought distillation enable smaller language models to reason better and faster. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 12142–12157. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.emnlp-main.610. URL https://aclanthology.org/2025.emnlp-main.610/.

Xiaoxue Cheng, Junyi Li, Zhenduo Zhang, Xinyu Tang, Wayne Xin Zhao, Xinyu Kong, and Zhiqiang Zhang. Incentivizing dual process thinking for efficient large language model reasoning. In *Advances in Neural Information Processing Systems*, volume 38, 2025. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/e01d556690c08fed8bd52d786dcef19f-Abstract-Conference.html.

Zhengxiang Cheng, Dongping Chen, Mingyang Fu, and Tianyi Zhou. Optimizing length compression in large reasoning models. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3232–3250. Association for Computational Linguistics, 2026. doi: 10.18653/v1/2026.acl-long.146. URL https://aclanthology.org/2026.acl-long.146/.

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.

Yingqian Cui, Pengfei He, Jingying Zeng, Hui Liu, Xianfeng Tang, Zhenwei Dai, Yan Han, Chen Luo, Jing Huang, Zhen Li, Suhang Wang, Yue Xing, Jiliang Tang, and Qi He. Stepwise perplexity-guided refinement for efficient chain-of-thought reasoning in large language models. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 18581–18597. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.findings-acl.956. URL https://aclanthology.org/2025.findings-acl.956/.

Muzhi Dai, Chenxu Yang, and Qingyi Si. S-GRPO: Early exit via reinforcement learning in reasoning models. In *Advances in Neural Information Processing Systems*, volume 38,

2025. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/44f09d1973a88529b40029e6c3693ded-Abstract-Conference.html.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, et al. DeepSeek-R1 incentivizes reasoning in LLMs through reinforcement learning. *Nature*, 645:633–638, 2025. doi: 10.1038/s41586-025-09422-z. URL https://www.nature.com/articles/s41586-025-09422-z.

Yuntian Deng, Kiran Prasad, Roland Fernandez, Paul Smolensky, Vishrav Chaudhary, and Stuart Shieber. Implicit chain of thought reasoning via knowledge distillation. *arXiv preprint arXiv:2311.01460*, 2023. doi: 10.48550/arXiv.2311.01460. URL https://arxiv.org/abs/2311.01460.

Yuntian Deng, Yejin Choi, and Stuart Shieber. From explicit CoT to implicit CoT: Learning to internalize CoT step by step. *arXiv preprint arXiv:2405.14838*, 2024. doi: 10.48550/arXiv.2405.14838. URL https://arxiv.org/abs/2405.14838.

Kaituo Feng, Changsheng Li, Xiaolu Zhang, Jun Zhou, Ye Yuan, and Guoren Wang. Keypoint-based progressive chain-of-thought distillation for llms. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 13241–13255, 2024. URL https://proceedings.mlr.press/v235/feng24e.html.

Jiaxuan Gao, Shu Yan, Qixin Tan, Lu Yang, Shusheng Xu, Wei Fu, Zhiyu Mei, Kaifeng Lyu, and Yi Wu. How far are we from optimal reasoning efficiency? In *Advances in Neural Information Processing Systems*, volume 38, 2025. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/1022661f3f43406065641f16ce25eafa-Abstract-Conference.html.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*, 2021. URL https://arxiv.org/abs/2103.03874.

Shijue Huang, Hongru Wang, Wanjun Zhong, Zhaochen Su, Jiazhan Feng, Bowen Cao, and Yi R. Fung. AdaCtrl: Towards adaptive and controllable reasoning via difficulty-aware budgeting. *arXiv preprint arXiv:2505.18822*, 2025. doi: 10.48550/arXiv.2505.18822. URL https://arxiv.org/abs/2505.18822.

Yixiao Huang, Hanlin Zhu, Zixuan Wang, Jiantao Jiao, Stuart Russell, Somayeh Sojoudi, and Song Mei. Transformers provably learn to internalize chain-of-thought. In *Workshop on Latent & Implicit Thinking – Going Beyond CoT Reasoning*, 2026. URL https://openreview.net/forum?id=gFdDXKfGbn.

Hugging Face H4. Aime 2024. Hugging Face dataset, 2024. URL https://huggingface.co/datasets/HuggingFaceH4/aime_2024.

Yu Kang, Xianghui Sun, Liangyu Chen, and Wei Zou. C3oT: Generating shorter chain-of-thought without compromising effectiveness. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 24312–24320. AAAI Press, 2025. doi: 10.1609/aaai.v39i23.34608. URL https://ojs.aaai.org/index.php/AAAI/article/view/34608.

Yanhao Li, Lu Ma, Jiaran Zhang, Lexiang Tang, Wentao Zhang, and Guibo Luo. LEASH: Adaptive length penalty and reward shaping for efficient large reasoning model. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 2846–2856. Association for Computational Linguistics, 2026. doi: 10.18653/v1/2026.acl-long.129. URL https://aclanthology.org/2026.acl-long.129/.

Yuetai Li, Xiang Yue, Zhangchen Xu, Fengqing Jiang, Luyao Niu, Bill Yuchen Lin, Bhaskar Ramasubramanian, and Radha Poovendran. Small models struggle to learn from strong reasoners. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 25366–25394. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.findings-acl.1301. URL https://aclanthology.org/2025.findings-acl.1301/.

Zeju LI, Jianyuan Zhong, Ziyang Zheng, Xiangyu Wen, Zhijian Xu, Yingying Cheng, Fan Zhang, and Qiang Xu. Making slow thinking faster: Compressing llm chain-of-thought via step entropy. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 38331–38349, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/40b2844d3c1451a90e0afb193bd735fc-Paper-Conference.pdf.

Zhengkai Lin, Zhihang Fu, Ze Chen, Chao Chen, Liang Xie, Wenxiao Wang, Deng Cai, Zheng Wang, and Jieping Ye. Controlling thinking speed in reasoning models. In *Advances in Neural Information Processing Systems*, volume 38, 2025. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/70d46c12643e8f056a693ed43472dc35-Abstract-Conference.html.

Han Liu, Shuotian Ma, Hui Li, Xiaotong Zhang, Fenglong Ma, and Hong Yu. Pru-CoT: Towards efficient reasoning distillation via pruning chain-of-thought. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 33744–33756. Association for Computational Linguistics, 2026a. doi: 10.18653/v1/2026.findings-acl.1684. URL https://aclanthology.org/2026.findings-acl.1684/.

Kang Liu, YongKang Liu, Xiaocui Yang, Peidong Wang, Wen Zhang, Shi Feng, Yifei Zhang, and Daling Wang. NEAT: Neuron-based early exit for large reasoning models. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 24616–24627. Association for Computational Linguistics, 2026b. doi: 10.18653/v1/2026.findings-acl.1231. URL https://aclanthology.org/2026.findings-acl.1231/.

Tengxiao Liu, Qipeng Guo, Xiangkun Hu, Cheng Jiayang, Yue Zhang, Xipeng Qiu, and Zheng Zhang. Can language models learn to skip steps? In *Advances in Neural Information Processing Systems*, volume 37, 2024. doi: 10.52202/079017-1441. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/504fa7e518da9d1b53a233ed20a38b46-Abstract-Conference.html.

Yongjiang Liu, Haoxi Li, Xiaosong Ma, Jie Zhang, and Song Guo. Think how to think: Mitigating overthinking with autonomous difficulty cognition in large reasoning models. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 38105–38126. Association for Computational Linguistics, 2026c. doi: 10.18653/v1/2026.acl-long.1766. URL https://aclanthology.org/2026.acl-long.1766/.

Haotian Luo, Haiying He, Yibo Wang, Jinluan Yang, Rui Liu, Naiqiang Tan, Xiaochun Cao, Dacheng Tao, and Li Shen. Ada-R1: Hybrid-CoT via bi-level adaptive reasoning optimization. In *Advances in Neural Information Processing Systems*, volume 38, 2025. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/558da25ef295121e2d79209a95e45436-Abstract-Conference.html.

Wenjie Ma, Jingxuan He, Charlie Snell, Tyler Griggs, Sewon Min, and Matei Zaharia. Reasoning models can be effective without thinking, 2025a. URL https://arxiv.org/abs/2504.09858.

Xinyin Ma, Guangnian Wan, Runpeng Yu, Gongfan Fang, and Xinchao Wang. CoT-valve: Length-compressible chain-of-thought tuning. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 6025–6035. Association for Computational Linguistics, 2025b. doi: 10.18653/v1/2025.acl-long.300. URL https://aclanthology.org/2025.acl-long.300/.

Shuaiyi Nie, Dingsiyu, Wenyuan Zhang, Linhao Yu, Tianmeng Yang, Yao Chen, Weichong Yin, Yu Sun, Hua Wu, and Tingwen Liu. AttnPO: Attention-guided process supervision for efficient reasoning. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 39728–39748. Association for Computational Linguistics, 2026. doi: 10.18653/v1/2026.acl-long.1845. URL https://aclanthology.org/2026.acl-long.1845/.

Nathanaël Carraz Rakotonirina, Ren Pang, Neha Anna John, Michael Bohlke-Schneider, and Momchil Hardalov. Correct, concise and complete: Multi-stage training for adaptive reasoning. In

Findings of the Association for Computational Linguistics: ACL 2026, pp. 12798–12810. Association for Computational Linguistics, 2026. doi: 10.18653/v1/2026.findings-acl.622. URL https://aclanthology.org/2026.findings-acl.622/.

Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling llm test-time compute optimally can be more effective than scaling parameters for reasoning. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/1b623663fd9b874366f3ce019fdfdd44-Abstract-Conference.html.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. 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_files/paper/2022/hash/9d5609613524ecf4f15af0f7b31abca4-Abstract-Conference.html.

Heming Xia, Chak Tou Leong, Wenjie Wang, Yongqi Li, and Wenjie Li. TokenSkip: Controllable chain-of-thought compression in LLMs. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 3351–3363. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.emnlp-main.165. URL https://aclanthology.org/2025.emnlp-main.165/.

Yang Xiang, Yixin Ji, Ruotao Xu, Dan Qiao, Zheming Yang, Juntao Li, and Min Zhang. When is thinking enough? early exit via sufficiency assessment for efficient reasoning. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 23541–23556. Association for Computational Linguistics, 2026. doi: 10.18653/v1/2026.acl-long.1080. URL https://aclanthology.org/2026.acl-long.1080/.

Chenxu Yang, Qingyi Si, Yongjie Duan, Zheliang Zhu, Chenyu Zhu, Qiaowei Li, Minghui Chen, Zheng Lin, and Weiping Wang. Dynamic early exit in reasoning models. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 88170–88210, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/8df90a1440ce782d1f5607b7a38f2531-Paper-Conference.pdf.

Jingyang Yi, Jiazheng Wang, and Sida Li. Shorterbetter: Guiding reasoning models to find optimal inference length for efficient reasoning. In *Advances in Neural Information Processing Systems*, volume 38, 2025. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/377ca194437916417e3dbe0dc67254f1-Abstract-Conference.html.

Yuxiang Zhang, Zhengxu Yu, Weihang Pan, Zhongming Jin, Qiang Fu, Deng Cai, Binbin Lin, and Jieping Ye. Tokensqueeze: Performance-preserving compression for reasoning LLMs. In *Advances in Neural Information Processing Systems*, volume 38, 2025. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/8442d394d8266db6fdf92583a4783e8c-Abstract-Conference.html.

A Implementation Details

For all main-result runs, AGC^2 uses the reference-quality threshold $\tau = 0.90$, tolerance $\Delta = 0$, two local SFT updates per proposal ($m = 2$), at most three budget proposals per batch ($K = 3$), and a tolerance-failure allowance of $h = 1$. We use GPT-4o-mini as the compressor.

We use LoRA for finetuning, with rank 8, scaling factor 16, and dropout 0.05. The model parameters are optimized with AdamW. Training examples are grouped into batches with similar reference-CoT lengths, so candidate budgets are compared on a consistent local length scale.

A.1 Batch-Wise Procedure

Algorithm 1 lists the complete batch-wise procedure, including candidate construction, independent local adaptation, absorbability evaluation, and parameter selection.

Algorithm 1 Absorbability-Guided CoT Compression (AGC^2) for training batch t.

Input: pre-batch model parameters $\theta^\star_{t-1}$; long-CoT batch $\mathcal{B}^{\mathrm{long}}_t$; CoT compressor B; global reference-length anchor b_{ref}.
reference-quality and absolute-accuracy threshold τ; tolerance Δ; failure allowance h; local updates m; trial cap K; budget range $[b_{\min}, b_{\max}]$.
Output: model parameters $\theta^\star_t$ selected for the next training batch.

Compute $a_{t,0}$ and form $\mathcal{B}^{\mathrm{ref}}_t$ from $\mathcal{B}^{\mathrm{long}}_t$; set $b_{t,0}$ from the maximum long-CoT length.
$\theta^\star_t \leftarrow \theta^\star_{t-1}$, $\mathcal{S} \leftarrow \varnothing$; **if** $\mathcal{B}^{\mathrm{ref}}_t = \varnothing$ **then return** $\theta^\star_t$.
if $a_{t,0} \geq \tau$ **then** ▷ Reliable reference: search shorter budgets.
  $f \leftarrow 0$.
  for $j \leftarrow 1$ **to** K **do**
    Project $b^{\mathrm{short}}_{t,j}$ to the allowed integer range as b; **continue** if $b \in \mathcal{S}$.
    Add b to $\mathcal{S}$ and compress $\mathcal{B}^{\mathrm{ref}}_t$ at budget b.
    Adapt a fresh copy of $\theta^\star_{t-1}$ for m SFT updates, then evaluate it on $\mathcal{B}^{\mathrm{long}}_t$ to obtain $(\theta_{t,b}, a_{t,b})$.
    if $a_{t,b} \geq a_{t,0} - \Delta$ **then** $(b^\star_t, \theta^\star_t) \leftarrow (b, \theta_{t,b})$; **else** $f \leftarrow f + 1$ and **break if** $f \geq h$.
  end for
else ▷ Unreliable reference: search longer budgets.
  for $j \leftarrow 1$ **to** K **do**
    Project $b^{\mathrm{long}}_{t,j}$ to the allowed integer range as b; **continue** if $b \in \mathcal{S}$.
    Add b to $\mathcal{S}$ and compress $\mathcal{B}^{\mathrm{ref}}_t$ at budget b.
    Adapt a fresh copy of $\theta^\star_{t-1}$ for m SFT updates, then evaluate it on $\mathcal{B}^{\mathrm{long}}_t$ to obtain $(\theta_{t,b}, a_{t,b})$.
    if $a_{t,b} \geq \tau$ **then** $(b^\star_t, \theta^\star_t) \leftarrow (b, \theta_{t,b})$ and **break**.
  end for
end if
return $\theta^\star_t$

B Additional Sensitivity Results

The main text reports the default parameter settings for each sensitivity study. This appendix provides the corresponding results under additional settings of τ, Δ, and h.

B.1 Threshold τ Across Tolerance Settings Δ

Figure 8 reports the threshold sweep at $\Delta = 0.02$.

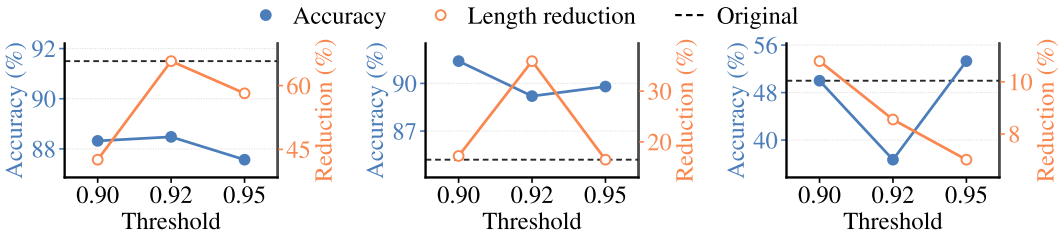


Figure 8: Additional sensitivity to the reference-quality threshold τ under accuracy-constrained search, with $\Delta = 0.02$ and $h = 1$ fixed. Each panel covers one benchmark for $\tau \in \{0.90, 0.92, 0.95\}$: the left axis reports answer accuracy and the right axis the length reduction relative to the original 7B model, whose accuracy is marked by the dashed line.

Figure 9 reports the sweep at $\Delta = 0.05$, where the candidate thresholds are $\tau \in \{0.90, 0.95, 0.96\}$.

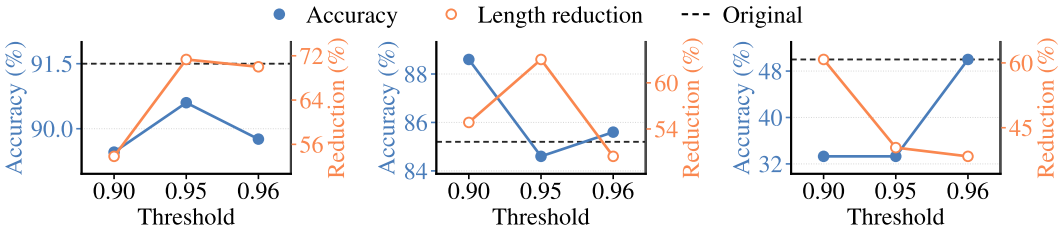


Figure 9: Additional sensitivity to the reference-quality threshold τ under accuracy-constrained search, with $\Delta = 0.05$ and $h = 1$ fixed. Each panel covers one benchmark for $\tau \in \{0.90, 0.95, 0.96\}$, with the same encoding as the main-text threshold figure.

Together with the main-text $\Delta = 0$ setting, these results show that the effect of τ depends on both the benchmark and Δ; no single threshold dominates across all three tasks.

B.2 Tolerance Δ at Fixed Threshold τ

The main-text tolerance analysis fixes $\tau = 0.90$ and $h = 1$. Figure 10 provides results at the additional setting $\tau = 0.95$.

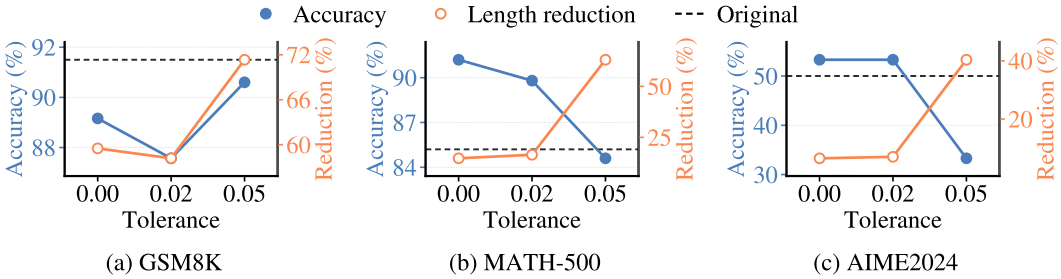
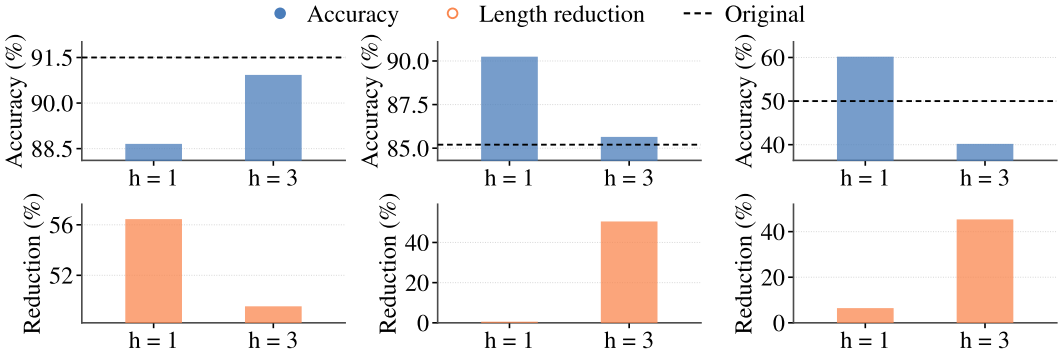


Figure 10: Additional sensitivity to the accuracy tolerance Δ under accuracy-constrained search, with $\tau = 0.95$ and $h = 1$ fixed. Each panel covers one benchmark for $\Delta \in \{0, 0.02, 0.05\}$, with the same encoding as the main-text tolerance figure.

Together with the main-text result at $\tau = 0.90$, this additional setting shows that Δ controls the accuracy allowed during shortening, while the realized trade-off remains task dependent. We therefore use the conservative default $\Delta = 0$ for the unified main-result configuration.

B.3 Tolerance-Failure Patience h Across Tolerance Settings Δ

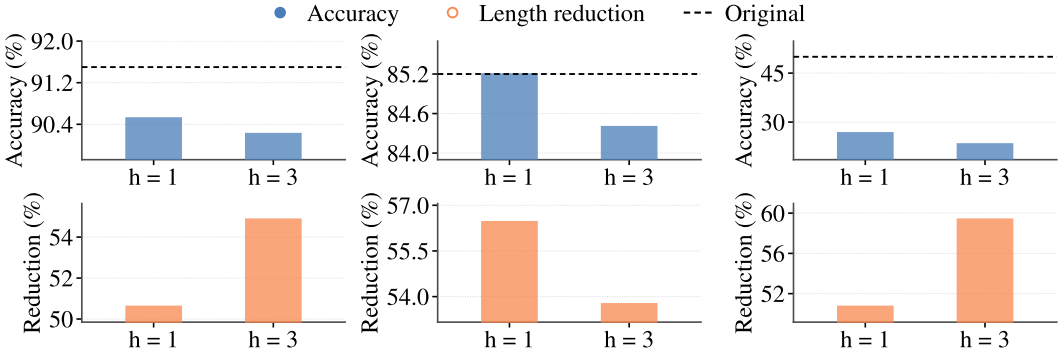
Figure 11 reports the allowance sweep at $\Delta = 0.02$.



(a) GSM8K (b) MATH-500 (c) AIME2024

Figure 11: Additional sensitivity to the tolerance-failure allowance h under accuracy-constrained search, with $\tau = 0.90$ and $\Delta = 0.02$ fixed. Each column covers one benchmark for $h \in \{1, 3\}$: the upper row reports answer accuracy and the lower row the length reduction relative to the original 7B model.

Figure 12 reports the sweep at $\Delta = 0.05$.



(a) GSM8K (b) MATH-500 (c) AIME2024

Figure 12: Additional sensitivity to the tolerance-failure allowance h under accuracy-constrained search, with $\tau = 0.90$ and $\Delta = 0.05$ fixed, with the same encoding as the main-text allowance figure.

Together with the main-text $\Delta = 0$ setting, these results show that increasing h extends the search past an infeasible candidate, but does not produce one fixed accuracy–length direction. Because every extra attempt requires another compression, adaptation, and evaluation cycle, we use $h = 1$ as the cost-conscious default.