HAIR: HIDDEN-STATE ANCHORED IMPLICIT REASONING WITH COMPACT TEXTUAL CHECKPOINTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Chain-of-thought (CoT) reasoning is accurate but slow, because every intermediate step is decoded as text. Latent reasoning moves these steps into hidden states, but without dense step-level supervision it stays below CoT beyond 1B parameters, it never commits an intermediate result, and it is compared with CoT baselines that differ in trace format, objective and decoding at once. We present HAIR, a streaming reasoning framework that commits each intermediate result as a one-line textual checkpoint, such as `G1=G0*2=18`, and places a block of b latent positions before each checkpoint in one append-only KV cache. Goal-boundary distillation (GBD) anchors the hidden state at each checkpoint to a shared-weight explicit path that reads the teacher's rationale for that step. Because the latent blocks and GBD can be removed while the checkpoint stream, data and decoding stay the same, HAIR also tests whether latent computation adds anything beyond the checkpoints. On GSM8K with Qwen3-1.7B, HAIR retains 97.5–99.1% of CoT accuracy at 42–59% of its latency from 18k training records, while CODI and Coconut trained on the same records score 30 or more points lower. The checkpoints carry this accuracy: latent blocks and GBD change it by less than the seed-to-seed spread, although GBD lowers its loss by three orders of magnitude. The savings carry over to SVAMP, GSM-Hard and MATH.

1 INTRODUCTION

Chain-of-thought (CoT) lets a language model solve multi-step problems by writing out its intermediate steps (Wei et al., 2022), but every step is decoded token by token, and most of these tokens make the text readable rather than carry the computation (Hao et al., 2025). Latent reasoning moves the steps into hidden states: Coconut feeds the last hidden state back as the next input (Hao et al., 2025), CODI distills an explicit chain into a few continuous tokens (Shen et al., 2025), and later work compresses, parallelizes or supervises such chains (Tan et al., 2025; Wu et al., 2025; Wei et al., 2026).

Three problems remain open. First, beyond 1B parameters latent methods trail CoT unless every step is supervised: with Llama-3.2-1B and 385k equation traces, CODI trails CoT by 6 points and Coconut by 16 (Shen et al., 2025), and closing the gap takes extra machinery such as the step decoder of SIM-CoT (Wei et al., 2026) or the looped Transformer of LOTUS (Fan et al., 2026). Second, a latent chain never commits an intermediate result, so an early error propagates unseen and supervision must pass through the whole chain. Third, a latent method and its CoT baseline differ in trace format, objective and decoding at once, so the comparison cannot tell whether accuracy comes from the latent positions; hybrid methods that keep some steps as text (Su et al., 2025; Shi et al., 2026) are compared with fully latent baselines, not with the same stream without its latent parts.

HAIR addresses the three problems with one design. A reasoning step has a *result*, which later steps need, and a *justification*, which they do not. HAIR writes the result as a one-line checkpoint such as `G1=G0*2=18` and leaves the justification to a block of b latent positions before it, all in one append-only KV cache; the model decides how many checkpoints to write (Figure 1). Each checkpoint ends a step and so gives the latent block a supervision point: goal-boundary distillation (GBD) anchors the latent state there to the state that a shared-weight explicit path reaches after reading the teacher's rationale. Setting $b = 0$ or turning off GBD removes the latent parts while the checkpoint stream, data, prompt and decoding stay the same, so each ablation changes one component.

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.

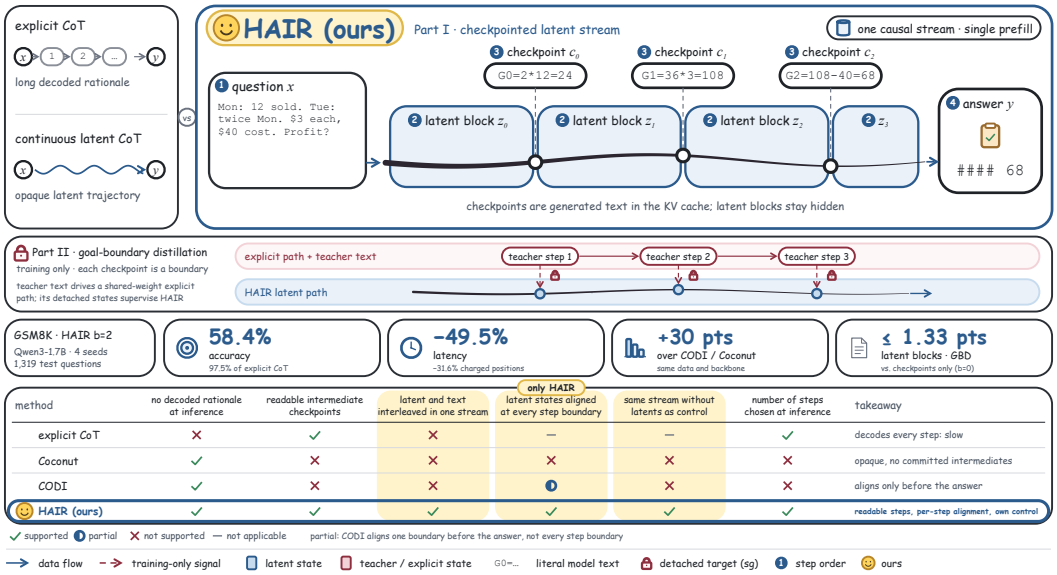

Figure 1: **HAIR at a glance.** *Top:* explicit CoT decodes a long rationale and latent CoT keeps an opaque trajectory; HAIR prefills the question once and alternates latent blocks z_k with generated checkpoints c_k in one causal stream. *Middle:* in training, GBD aligns the latent state at each checkpoint with a shared-weight explicit path that reads the teacher step. *Bottom:* GSM8K results and design properties.

Our contributions are:

- HAIR, a checkpointed latent stream in which the model writes its own goal checkpoints and a latent block precedes each one in a single append-only KV cache; every intermediate result is a line that later steps read and whose arithmetic can be checked.
- Goal-boundary distillation, a per-step objective that takes its targets from the model's own explicit path at the checkpoint boundaries, with no extra module and no inference cost.
- An evaluation with up to six training seeds per setting on GSM8K, two shifted test sets and MATH. HAIR retains 97.5–99.1% of CoT accuracy at 42–59% of its latency with 18k training records and exceeds CODI and Coconut trained on the same records by 30 or more points; paired ablations show that the checkpoints carry this accuracy, while latent blocks and GBD change it by less than the seed-to-seed spread.

2 Related Work

Implicit and continuous chain-of-thought. Early implicit CoT distills a teacher's hidden states so that a student answers without steps (Deng et al., 2023), or deletes CoT tokens during fine-tuning (Deng et al., 2024). Coconut feeds continuous thoughts back as inputs (Hao et al., 2025), which can hold several search frontiers at once (Zhu et al., 2025); CODI aligns an explicit and an implicit task at the token before the answer (Shen et al., 2025) and matches CoT at GPT-2 scale but not with larger backbones (Wei et al., 2026; Fan et al., 2026). Other methods compress latent embeddings (Tan et al., 2025), update them in parallel (Wu et al., 2025), evolve them with a dedicated module (Jiang et al., 2025) or project them from an assistant model (Xu et al., 2025b). Pause tokens and Quiet-STaR add hidden computation between generated tokens (Goyal et al., 2024; Zelikman et al., 2024); a $b = 1$ block of HAIR is one learned pause position before each checkpoint. None of these methods emits intermediate results; HAIR commits each one and uses it as a supervision boundary.

Step-level supervision and interleaving. SIM-CoT reconstructs each explicit step from its latent token with a training-time decoder (Wei et al., 2026), and LOTUS supervises every position of a looped Transformer (Saunshi et al., 2025) and closes the gap to CoT at 3B (Fan et al., 2026); both keep the steps implicit at inference. GBD is also step-level, but its target is the model's own state

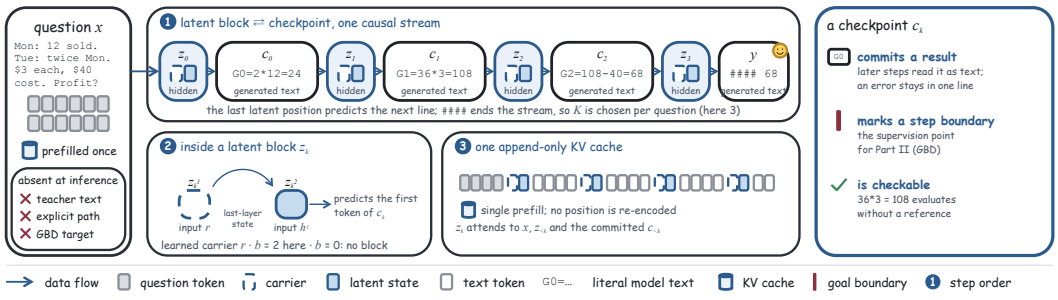

Figure 2: **The checkpointed latent stream at inference.** The question is prefilled once; a block of b latent positions precedes each generated checkpoint, and all positions share one append-only KV cache.

after reading the rationale. Token Assorted replaces the start of a trace with discrete latent codes (Su et al., 2025), SwiReasoning switches between latent and explicit decoding by entropy (Shi et al., 2026), and Self-SoftCoT alternates soft thinking with speaking (Dong et al., 2026); HAIR writes only a one-line result per step and keeps everything else latent.

Compact traces and analyses of latent steps. Program-of-thought separates what to compute from how (Chen et al., 2023; Gao et al., 2023), Chain of Draft prompts for minimal drafts (Xu et al., 2025a), and TokenSkip and C3oT fine-tune on compressed CoT (Xia et al., 2025; Kang et al., 2025); HAIR uses such a compact trace as the anchor of a latent stream and as its control. Coconut's latents can act as placeholders (Zhang et al., 2025), latent positions take distinct roles rather than adding uniform depth (Li et al., 2026), and latent CoT trades exploration for execution accuracy (Zou et al., 2026). Our ablations extend these analyses to a stream in which committed results and latent computation vary independently.

3 METHOD

A teacher model decomposes each training question x into goals $g_0, \ldots, g_{K-1}$. Goal g_k has a description d_k, a rationale r_k, an expression e_k that may refer to earlier goals, and its value v_k. HAIR learns to write, from x alone, the result (e_k, v_k) of each goal and then the answer y; d_k and r_k are used in training only. Appendix A gives prompts, decoding rules and loss details.

Checkpoints. Goal k is written as one line c_k = `Gk=`e_k`=`v_k, e.g. `G0=16-3-4=9` and then `G1=G0*2=18`, and the stream ends with `#### 18`. A checkpoint keeps the operation and its result and drops the explanation; at the median it has 14.5% of the characters of the corresponding CoT (Appendix B). It *commits* a result that later steps read as text, so a wrong value is written where it first appears; it *marks the boundary* of a step, which GBD uses; and its arithmetic can be *checked* without a reference. For MATH, a checkpoint is a compact statement `Gk:` *result*.

Latent blocks. For a question x, HAIR generates $x, z_0, c_0, z_1, c_1, \ldots, z_K, y$ (Figure 2), where each z_k is a block of b latent positions. A block starts from a learned carrier vector r, and inside it the last-layer state of each position is the input of the next, as in Coconut (Hao et al., 2025). After each block the model writes a checkpoint or the answer marker, so it chooses K per question. The question is prefilled once and all positions share one append-only KV cache, so a question costs $\sum_k |c_k| + |y|$ decoded tokens and $(K+1)\, b$ latent positions.

Goal-boundary distillation. Training unrolls the stream with teacher-forced checkpoints along two paths that share all weights (Figure 3). The latent path runs z_k and is trained with cross-entropy on the checkpoint ($\mathcal{L}_{\text{ckpt}}$) and the answer ($\mathcal{L}_{\text{ans}}$). The explicit path starts from the same prefix and writes d_k, r_k and then the same checkpoint ($\mathcal{L}_{\text{aux}}$). Both reach a position that predicts the first token

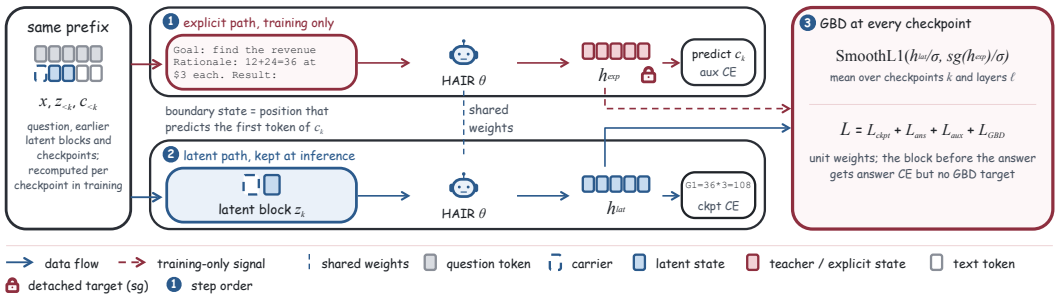

Figure 3: **Goal-boundary distillation at one checkpoint.** From the same prefix, the explicit path (training only) reads the teacher rationale and supplies a detached layer-wise target at the position that predicts c_k; the latent path reaches that position through z_k.

of c_k, and GBD anchors the latent state $h^{\text{lat}}_{k,\ell}$ there to the explicit state $h^{\text{exp}}_{k,\ell}$ in every layer ℓ:

$$\mathcal{L}_{\text{GBD}} = \frac{1}{KN} \sum_{k=0}^{K-1} \sum_{\ell=1}^{N} \text{SmoothL1}\left(\frac{h^{\text{lat}}_{k,\ell}}{\sigma_{k,\ell}}, \frac{\text{sg}\left(h^{\text{exp}}_{k,\ell}\right)}{\sigma_{k,\ell}} \right), \tag{1}$$

where N is the number of layers, $\sigma_{k,\ell}$ the standard deviation of the detached explicit state and sg the stop-gradient. The objective is $\mathcal{L} = \mathcal{L}_{\text{ckpt}} + \mathcal{L}_{\text{ans}} + \mathcal{L}_{\text{aux}} + \mathcal{L}_{\text{GBD}}$ with unit weights. Unlike CODI, which aligns one state before the answer (Shen et al., 2025), GBD gives every block its own target, taken from the same weights in the same forward pass. At inference the explicit path is removed.

4 Experiments

4.1 Setup

Datasets. For training, we sample 20,000 questions from GSM8K-Aug (Deng et al., 2023) and keep the 18,394 whose teacher decomposition reaches the reference answer; 500 further GSM8K-Aug questions form the validation set. For in-distribution (IID) evaluation we use the 1,319 GSM8K test questions (Cobbe et al., 2021). For out-of-distribution (OOD) evaluation we apply the same models, without further training, to SVAMP (Patel et al., 2021), whose problems vary wording and structure, and GSM-Hard (Gao et al., 2023), the GSM8K test questions with larger numbers. For MATH (Hendrycks et al., 2021), we train on the 5,577 of 7,000 training problems with an accepted decomposition and test on MATH-500 (Lightman et al., 2024) and 500 further test problems (Fresh-500). Appendices A and B describe the teacher, the filtering and the data statistics.

Models and metrics. All systems fine-tune Qwen3-1.7B-Base (Yang et al., 2025) with LoRA (Hu et al., 2022) for the same 4,600 updates, about one epoch, and decode greedily (Table 8). Baselines are explicit CoT trained on the same records, and CODI (Shen et al., 2025) and Coconut (Hao et al., 2025) trained from their official code on the same backbone and records (Appendix G). Accuracy compares the final number with the reference. Charged positions count the prompt, decoded tokens and latent positions per question, and latency is measured on one RTX 4090D at batch size 1 (Appendix D). GSM8K results are means $\pm$ SD over up to six training seeds, and paired comparisons use exact McNemar tests. In total we train 30 GSM8K models, 27 for the six settings of Table 1 (Table 11) plus answer-only, CODI and Coconut, and four MATH models.

4.2 Main results

Accuracy and cost. On GSM8K, HAIR$_{b=2}$ reaches 58.43% against 59.93% for CoT, 97.5% of its accuracy, with 31.6% fewer charged positions and 49.5% lower latency, and no seed shows a significant paired difference ($p \geq .14$; Table 1). HAIR$_{b=0}$ retains 98.7% of CoT accuracy at 42.3% of its latency, and all five settings lie within 1.5 points of CoT. The savings come from shorter sequential decoding: CoT writes 107.5 tokens per question and HAIR 41.5–42.3 tokens plus 4–16 latent positions, and each decoded token or latent position costs one sequential forward pass of about

Table 1: **GSM8K accuracy and inference cost** (1,319 test questions). Accuracy: mean ± SD over 3–6 training seeds (Table 11); CODI and Coconut: one seed, latency not profiled. Charged positions per question = prompt (≈80) + decoded tokens + latent positions.

System	Accuracy (%)	% of CoT	Decoded tokens	Charged positions	Latency (ms)
Explicit CoT	**59.93** ± 0.59	100.0	107.5	188.1	2,385
CODI	18.95	31.6	8.6	≈95	–
Coconut	28.28	47.2	62.7	≈149	–
$\text{HAIR}_{b=0}$	59.14 ± 1.12	98.7	41.6	**121.0**	**1,008**
$\text{HAIR}_{b=1}$	59.38 ± 1.06	99.1	42.3	125.6	1,115
$\text{HAIR}_{b=2}$ (default)	58.43 ± 1.04	97.5	41.5	128.6	1,204
$\text{HAIR}_{b=4}$	58.45 ± 0.87	97.5	41.8	136.8	1,396
$\text{HAIR}_{b=2}$ w/o GBD	58.96 ± 0.04	98.4	42.0	129.2	1,193

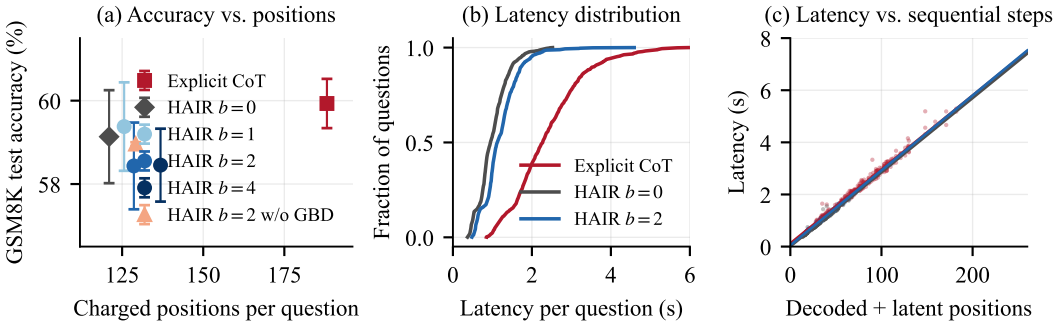

Figure 4: **Where the savings come from.** **(a)** Accuracy (mean ± SD over seeds) against charged positions. **(b)** Per-question latency on 300 validation questions (seed 29). **(c)** Latency is linear in the number of sequential positions (≈28.5 ms each, $r \geq 0.99$).

28.5 ms (Figure 4c). The shift is uniform across questions, with median latency falling from 2.27 s to 1.12 s (Figure 4b).

Comparison with latent baselines. CODI and Coconut trained on the same records reach 18.95% and 28.28%, 30–41 points below every HAIR setting. Their published results keep 73.5% and 90.3% of CoT with 385k equation-only traces, twenty times our data, whereas HAIR keeps 97.5–99.1% (Table 2); backbone, data and trace style all differ, so this comparison only places our results in context.

4.3 IID AND OOD GENERALIZATION

GSM8K to SVAMP and GSM-Hard. Table 3 evaluates the same models IID on GSM8K and OOD on SVAMP, which varies wording and structure, and GSM-Hard, which enlarges the numbers. On GSM8K all settings lie within 1.5 points of CoT with these seeds. Under the shift, the gap to CoT widens on SVAMP to 0.6–5.4 points (93–99% of CoT accuracy), whereas on GSM-Hard every setting matches or exceeds CoT, on the full set and on the clean subset. No latent budget is consistently better than $\text{HAIR}_{b=0}$: at seed 29, where all settings were run, $\text{HAIR}_{b=0}$ reaches 80.4% on SVAMP and 46.1% on GSM-Hard, against 77.2–80.2% and 43.2–46.0% with latent blocks.

A harder domain: MATH. Trained and tested on competition mathematics, where solutions are longer, HAIR saves more (Table 4): $\text{HAIR}_{b=0}$ and $\text{HAIR}_{b=2}$ use 45.5% and 43.9% fewer charged positions than CoT and stay within 1.2 points of it on MATH-500 ($p = .62$ for $\text{HAIR}_{b=2}$); on Fresh-500, $\text{HAIR}_{b=2}$ is 4.2 points below CoT. As on GSM8K, $b = 4$ adds cost without adding accuracy.

Table 2: **Accuracy relative to CoT in two data settings** (GSM8K). Each method is divided by the CoT baseline of its own setting. The published rows come from Shen et al. (2025) and are not controlled against ours.

Method	Method (%)	CoT (%)	% of CoT
Published: Llama-3.2-1B-Instruct, 385k equation-only traces			
Coconut	45.3	61.6	73.5
CODI	55.6	61.6	90.3
Ours: Qwen3-1.7B-Base, 18k natural-language traces			
CODI	19.0	59.9	31.6
Coconut	28.3	59.9	47.2
$\text{HAIR}_{b=0}$	59.1	59.9	98.7
$\text{HAIR}_{b=2}$	58.4	59.9	97.5

Table 3: **IID and OOD accuracy** (%) of the GSM8K-trained models, with no further training. CoT, b=0, b=2: mean over seeds 11, 29, 71; others seed 29. Clean: 900 GSM-Hard problems with a non-negative integer reference.

	IID	OOD		
System	GSM8K	SVAMP	GSM-Hard	Clean
Explicit CoT	60.0	80.8	43.1	47.9
$\text{HAIR}_{b=0}$	59.8	78.2	44.6	51.0
$\text{HAIR}_{b=1}$	58.6	80.2	46.0	51.3
$\text{HAIR}_{b=2}$	58.8	75.4	43.7	49.2
$\text{HAIR}_{b=4}$	58.8	79.4	43.2	49.8
w/o GBD	59.0	78.2	44.3	49.0

Table 4: **MATH** (accuracy, %; seed 11, 1,750 updates on 5,577 records). Charged positions per question on MATH-500. Fresh-500 was evaluated for CoT and $\text{HAIR}_{b=2}$ only.

System	MATH-500	Fresh-500	Charged
Explicit CoT	37.0	35.4	433.2
$\text{HAIR}_{b=0}$	36.4	–	236.1
$\text{HAIR}_{b=2}$	35.8	31.2	242.9
$\text{HAIR}_{b=4}$	35.4	–	261.2

Table 5: **What each change does to GSM8K accuracy.** Δ: accuracy of the second setting minus the first, paired by seed on the same 1,319 questions (mean $\pm$ SD over seeds). The last two columns count seeds on which $\Delta > 0$ and on which the difference is significant (exact McNemar, $p < .05$).

Change	From $\rightarrow$ to	Seeds	Δ (points)	$\Delta > 0$	Significant
CoT trace $\rightarrow$ checkpoints	CoT $\rightarrow$ $\text{HAIR}_{b=0}$	6	-0.80 ± 1.28	2	1, lower
Add latent blocks	$\text{HAIR}_{b=0}$ $\rightarrow$ $\text{HAIR}_{b=2}$ w/o GBD	3	-0.71 ± 0.29	0	0
Add GBD	$\text{HAIR}_{b=2}$ w/o GBD $\rightarrow$ $\text{HAIR}_{b=2}$	3	-0.63 ± 1.22	1	0
Add both	$\text{HAIR}_{b=0}$ $\rightarrow$ $\text{HAIR}_{b=2}$	4	-1.33 ± 0.91	0	1, lower
Latent budget $1 \rightarrow 2$	$\text{HAIR}_{b=1}$ $\rightarrow$ $\text{HAIR}_{b=2}$	4	-1.04 ± 1.60	1	1, lower
Latent budget $2 \rightarrow 4$	$\text{HAIR}_{b=2}$ $\rightarrow$ $\text{HAIR}_{b=4}$	3	$+0.13 \pm 1.43$	1	0

4.4 Ablations

Settings. Starting from $\text{HAIR}_{b=2}$, we remove GBD ($\text{HAIR}_{b=2}$ w/o GBD), remove the latent blocks and with them GBD ($\text{HAIR}_{b=0}$, which keeps $\mathcal{L}_{\text{aux}}$), or vary the budget $b \in \{1, 2, 4\}$. All keep the checkpoint stream, records, prompt, backbone and decoding, so neighbouring settings differ in one component. Two references change more: explicit CoT, and answer-only, which writes the answer directly (Table 9).

Checkpoints. Committing intermediate results matters first. Answer-only reaches 20.32% on seed 11, close to CODI and Coconut, whereas $\text{HAIR}_{b=0}$ reaches 59.36% and CoT 59.06% with the same seed. Answer-only also drops the explicit path, so this 39-point gap covers the checkpoints together with the rationale supervision that comes with them. Replacing the CoT trace by checkpoints then changes accuracy by only -0.80 ± 1.28 points while removing 61% of the decoded tokens (Table 5).

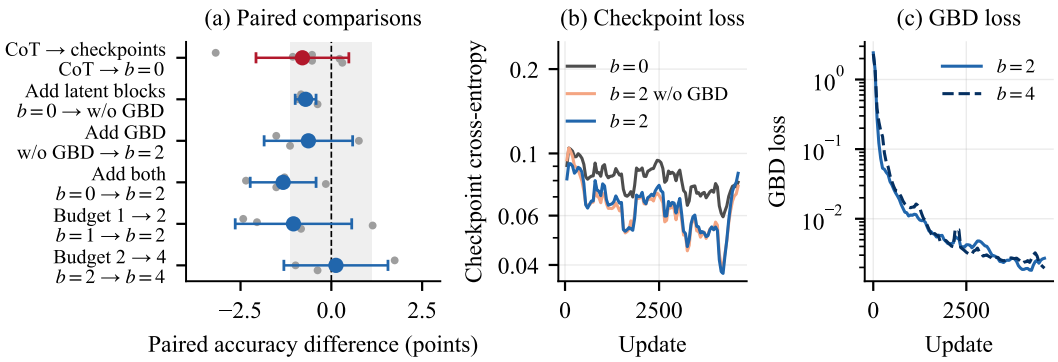

Figure 5: **Ablations and training dynamics.** **(a)** Paired differences of Table 5: seeds (grey) and mean ± SD; the band is the seed SD of $\text{HAIR}_{b=0}$. **(b)** Training cross-entropy of the checkpoints (seed 29). **(c)** GBD loss.

Table 6: **Changing b only at test time.** Accuracy (%) of the four $\text{HAIR}_{b=2}$ models, which were trained with $b = 2$; columns are training seeds.

Test b	11	29	47	71
0 (no block)	0.00	0.00	0.00	0.00
1	58.07	59.44	4.02	58.76
2 (trained)	57.85	59.74	57.39	58.76
4	1.52	0.00	0.15	0.30

Table 7: **Quality of the written checkpoints** (%, mean over seeds; 1,236 questions with a verified teacher decomposition). Consistent: the expression, with the model's earlier values, gives the stated value (per line, or all lines of a question). The last two rows compare with the teacher.

	b=0	b=1	b=2	b=4	w/o GBD
Consistent lines	94.6	94.4	93.9	94.0	94.3
Consistent questions	85.2	84.6	83.9	83.4	84.6
Teacher step count	67.5	67.4	67.9	67.7	67.3
All teacher values	54.3	54.4	53.2	53.4	54.5

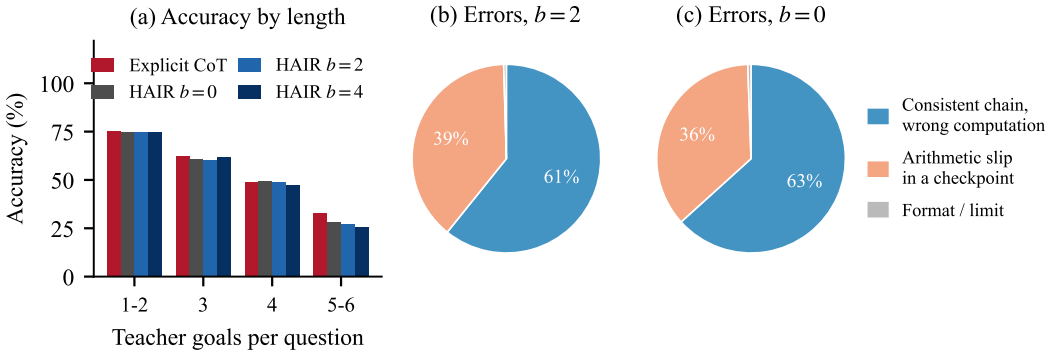

Figure 6: **Error analysis on GSM8K. (a)** Accuracy by the number of teacher goals (1,236 questions with a verified decomposition). **(b, c)** Wrong answers of $\text{HAIR}_{b=2}$ (2,193, four seeds) and $\text{HAIR}_{b=0}$ (3,234, six seeds): all checkpoints consistent but the computation wrong, an arithmetic slip in a checkpoint, or a format or length failure.

Latent blocks and GBD. Adding latent blocks changes accuracy by -0.71 ± 0.29 points and adding GBD by -0.63 ± 1.22. Every mean difference involving them lies between -1.33 and $+0.13$ points, within the seed SD of $\text{HAIR}_{b=0}$ (± 1.12; Figure 5a), and $b = 1$ gives the highest mean. With the same seed, $\text{HAIR}_{b=2}$ and $\text{HAIR}_{b=0}$ differ in correctness on 14.4% of questions, fewer than two $\text{HAIR}_{b=0}$ models trained with different seeds (17.9%). Both objectives are optimized: the GBD loss falls from about 3 to 0.002 (Figure 5c), and the blocks lower the training cross-entropy of the checkpoints from 0.064 to 0.047–0.048 (Figure 5b), yet test accuracy does not improve.

Test-time block length. Changing b only at inference shows what the trained block does (Table 6). Without the block, accuracy drops to 0.00% on every seed because the model writes a goal description and rationale, as the explicit path does from the same prefix; with $b = 4$ it also collapses, while $b = 1$ stays within 0.3 points on three of four seeds. The block thus acts as a switch that tells the shared weights to write a checkpoint.

4.5 Analysis

Checkpoint quality. In every setting, 93.9–94.6% of the written lines are arithmetically consistent (Table 7). All settings match the teacher's step count on about 67.5% of questions and write every teacher value on 53–55%, and in that case the answer is correct 99.2% of the time; the first missed teacher value falls at the same step in every setting (Table 17). These measures miss valid alternative decompositions (Appendix C), so they compare settings rather than measure correctness. By them, latent blocks change neither consistency nor coverage, plausibly because the expression already states the operation that the rationale justifies.

Errors. On the 1,132 of 1,236 verified test questions with up to four goals, all systems lie within 2.2 points of each other; on the 104 with five or six goals, CoT reaches 33.0% against 28.2% for $\text{HAIR}_{b=0}$ and 26.9% for $\text{HAIR}_{b=2}$, a third of the gap for $\text{HAIR}_{b=2}$, and $b = 4$ does not close it (25.6%; Figure 6a). Of the wrong answers, 61–63% contain only consistent checkpoints, so the error lies in what was computed; 36–39% contain an arithmetic slip, which can be detected without a reference; fewer than 1% break the format (Figure 6b,c).

Further analyses. Appendix D reports per-seed accuracy and the latency profile, and Appendix E per-seed paired tests, answer agreement between settings, accuracy by difficulty and the position of the first missed teacher value. Appendices F and H give training cost and termination statistics.

5 Conclusion

HAIR is a streaming reasoning framework in which the model writes its own goal checkpoints, with latent blocks between them trained by goal-boundary distillation at each checkpoint. On GSM8K it stays within 1.5 points of explicit CoT at about half the latency, while CODI and Coconut trained on the same data fall 30 or more points below. Comparisons within the stream show that the committed checkpoints carry this accuracy; at this scale, latent blocks and GBD give no consistent gain, even though GBD aligns the blocks closely with explicit reasoning. Committing compact results is the component to keep when reasoning is made cheaper.

Limitations. We use one 1.7B backbone and blocks of at most four positions; larger backbones may use longer blocks differently (Fan et al., 2026). Checkpoint granularity follows one teacher's decompositions. The extra explicit path makes training costlier than CoT training (Appendix F), but not inference. Our domains are arithmetic and mathematics, where a step has a checkable result.

Reproducibility statement

Section 4.1 and Appendix A specify data, splits, training and decoding. We will release the code, the accepted teacher records and their identifiers, all predictions, the cost-accounting and latency-profiling scripts, and the scripts that produce every number and figure in this paper from the saved predictions and training logs.

LLM USAGE

A DeepSeek model generated the goal decompositions used as training data (Section 4.1). LLM assistants helped with implementation, result inspection and editing of the manuscript. The authors checked all numbers against the saved predictions and are responsible for the content.

REFERENCES

Wenhu Chen, Xueguang Ma, Xinyi Wang, and William W. Cohen. Program of thoughts prompting: Disentangling computation from reasoning for numerical reasoning tasks. *Transactions on Machine Learning Research*, 2023.

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.

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.

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.

Liangliang Dong, Lianlei Shan, and Shuaimin Li. Self-SoftCoT: A self-consistent framework via position-aware latent space reinforcement learning. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 32393–32414, 2026. doi: 10.18653/v1/2026.acl-long.1496. URL https://aclanthology.org/2026.acl-long.1496/.

Ying Fan, Anej Svete, and Kangwook Lee. Bridging the gap between latent and explicit reasoning with looped transformers. *arXiv preprint arXiv:2606.31779*, 2026.

Luyu Gao, Aman Madaan, Shuyan Zhou, Uri Alon, Pengfei Liu, Yiming Yang, Jamie Callan, and Graham Neubig. PAL: Program-aided language models. In *International Conference on Machine Learning*, 2023. PMLR 202.

Sachin Goyal, Ziwei Ji, Ankit Singh Rawat, Aditya Krishna Menon, Sanjiv Kumar, and Vaishnavh Nagarajan. Think before you speak: Training language models with pause tokens. In *International Conference on Learning Representations*, 2024.

Shibo Hao, Sainbayar Sukhbaatar, DiJia Su, Xian Li, Zhiting Hu, Jason Weston, and Yuandong Tian. Training large language models to reason in a continuous latent space. In *Conference on Language Modeling (COLM)*, 2025.

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. In *Advances in Neural Information Processing Systems Datasets and Benchmarks Track*, 2021.

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.

Nan Jiang, Ziming Wu, De-Chuan Zhan, Fuming Lai, and Shaobing Lian. DART: Distilling autoregressive reasoning to silent thought. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 5100–5108, 2025.

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*, 2025.

Zirui Li, Xuefeng Bai, Kehai Chen, Yizhi Li, Jian Yang, Chenghua Lin, and Min Zhang. Dynamics within latent chain-of-thought: An empirical study of causal structure. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026. PMLR 306; arXiv:2602.08783.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison 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.

Arkil Patel, Satwik Bhattamishra, and Navin Goyal. Are NLP models really able to solve simple math word problems? In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 2080–2094, 2021.

Nikunj Saunshi, Nishanth Dikkala, Zhiyuan Li, Sanjiv Kumar, and Sashank J. Reddi. Reasoning with latent thoughts: On the power of looped transformers. In *International Conference on Learning Representations*, 2025. arXiv:2502.17416.

Zhenyi Shen, Hanqi Yan, Linhai Zhang, Zhanghao Hu, Yali Du, and Yulan He. CODI: Compressing chain-of-thought into continuous space via self-distillation. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 677–693, 2025. doi: 10.18653/v1/2025.emnlp-main.36. URL https://aclanthology.org/2025.emnlp-main.36/.

Dachuan Shi, Abedelkadir Asi, Keying Li, Xiangchi Yuan, Leyan Pan, Wenke Lee, and Wen Xiao. SwiReasoning: Switch-thinking in latent and explicit for pareto-superior reasoning LLMs. In *International Conference on Learning Representations*, 2026. arXiv:2510.05069.

DiJia Su, Hanlin Zhu, Yingchen Xu, Jiantao Jiao, Yuandong Tian, and Qinqing Zheng. Token assorted: Mixing latent and text tokens for improved language model reasoning. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 57144–57163. PMLR, 2025. URL https://proceedings.mlr.press/v267/su25g.html.

Wenhui Tan, Jiaze Li, Jianzhong Ju, Zhenbo Luo, Ruihua Song, and Jian Luan. Think silently, think fast: Dynamic latent compression of LLM reasoning chains. In *Advances in Neural Information Processing Systems*, 2025.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc V. Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, 2022.

Xilin Wei, Xiaoran Liu, Yuhang Zang, Xiaoyi Dong, Yuhang Cao, Jiaqi Wang, Xipeng Qiu, and Dahua Lin. SIM-CoT: Supervised implicit chain-of-thought. In *International Conference on Learning Representations*, 2026.

Haoyi Wu, Zhihao Teng, and Kewei Tu. Parallel continuous chain-of-thought with jacobi iteration. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 914–926, 2025. doi: 10.18653/v1/2025.emnlp-main.47. URL https://aclanthology.org/2025.emnlp-main.47/.

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*, 2025.

Silei Xu, Wenhao Xie, Lingxiao Zhao, and Pengcheng He. Chain of draft: Thinking faster by writing less. *arXiv preprint arXiv:2502.18600*, 2025a.

Yige Xu, Xu Guo, Zhiwei Zeng, and Chunyan Miao. SoftCoT: Soft chain-of-thought for efficient reasoning with LLMs. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 23336–23351, 2025b. doi: 10.18653/v1/2025.acl-long.1137. URL https://aclanthology.org/2025.acl-long.1137/.

An Yang et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Eric Zelikman, Georges Harik, Yijia Shao, Varuna Jayasiri, Nick Haber, and Noah D. Goodman. Quiet-STaR: Language models can teach themselves to think before speaking. In *First Conference on Language Modeling (COLM)*, 2024.

Yuyi Zhang, Boyu Tang, Tianjie Ju, Sufeng Duan, and Gongshen Liu. Do latent tokens think? a causal and adversarial analysis of chain-of-continuous-thought. *arXiv preprint arXiv:2512.21711*, 2025.

Hanlin Zhu, Shibo Hao, Zhiting Hu, Jiantao Jiao, Stuart Russell, and Yuandong Tian. Reasoning by superposition: A theoretical perspective on chain of continuous thought. In *Advances in Neural Information Processing Systems*, 2025.

Jiaxuan Zou, Yaozhong Xiong, and Yong Liu. Capabilities and fundamental limits of latent chain-of-thought. *arXiv preprint arXiv:2602.01148*, 2026.

A Implementation details

Prompt and output format. The GSM8K prompt is `Question: {question}` followed by the instruction `Write G0=expression=value, one result per line. End with #### answer.` For competition mathematics, where an intermediate state is often not a single expression, a checkpoint is a compact statement `G`k`:` *result*, such as a formula, a condition or a derived fact, and the MATH prompt asks for such lines instead. Teacher goals, rationales and reference answers never enter the inference prompt. Explicit CoT uses its own prompt (`Solve step by step`) and is trained on the concatenation of each goal's rationale followed by <<*expression*=*value*>>, then `#### answer`. Answer-only is trained on `#### answer` after the question. $\text{HAIR}_{b=0}$ keeps the explicit path and $\mathcal{L}_{\text{aux}}$ but has no block, so GBD is undefined.

Latent block. The carrier is a learned vector r of the hidden size H, initialized from $\mathcal{N}(0, 0.02^2)$ and normalized as $\sqrt{H}\, r/(\|r\|_2 + 10^{-6})$ before each block. Each block starts from this carrier, and the last-layer hidden state of each latent position is the input embedding of the next; the input recurrence restarts at every block, while attention covers the whole cache. A $b = 1$ block is therefore a single learned pause position (Goyal et al., 2024), and each further position adds one recurrent step.

Decoding. The last latent position of z_k predicts the first token of c_k, which is decoded greedily until a newline. If the model writes the answer marker `####` instead, it decodes the answer and stops. A checkpoint line ends at a generated newline and must match `G`k`=...=...` (GSM8K) or `G`k`: ...` (MATH); otherwise the rollout stops and is scored as wrong.

Losses. The explicit path writes d_k and r_k, then the marker `Result:` and c_k. The block z_K before the answer receives answer cross-entropy but no GBD target, because the answer has no rationale. The checkpoint cross-entropy is the token mean within each checkpoint and the auxiliary cross-entropy the token mean over the goal description, rationale and checkpoint, both averaged over checkpoints; the answer cross-entropy is the token mean over the answer and the end-of-sequence token. GBD uses all Transformer layers (not the embedding output); $\sigma_{k,\ell}$ is the standard deviation of the detached explicit state over hidden coordinates plus 10^{-6}. The four terms are summed with unit weights, which were not tuned.

Table 8: Training and decoding settings.

Setting	GSM8K	MATH
Backbone	Qwen3-1.7B-Base, bf16, LoRA rank 16, $\alpha = 32$, dropout 0	
Training records	18,394 accepted of 20,000	5,577 accepted of 7,000
Updates $\times$ accumulation	4,600 $\times$ 4	1,750 $\times$ 4
Learning rate	8×10^{-5}	8×10^{-5}
Input / training-target limit	2,048 / 512 tokens	4,096 / 4,096 tokens
Output limit / block limit	256 tokens / 8 blocks	1,024 tokens / 16 blocks
Decoding	greedy, batch size 1	
Training seeds	11, 29, 47, 53, 71, 89 (Table 11)	11

Ablation settings. Table 9 lists the settings of Section 4.4. $\text{HAIR}_{b=0}$ removes the latent blocks and therefore GBD, but keeps $\mathcal{L}_{\text{aux}}$ so that the shared weights still see the rationales. Answer-only

is trained on the same records to write the answer directly, without checkpoints or an explicit path. Explicit CoT is trained on the rationales and expressions of the same records with its own prompt.

Table 9: **Ablation settings.** All rows use the same 18,394 training records, backbone, LoRA configuration, 4,600 updates and greedy decoding. All HAIR settings also train the shared-weight explicit path, which is dropped at inference.

Setting	Model writes per step	Latent b	GBD	Purpose
References				
Explicit CoT	rationale + expression	–	–	full-rationale reference
Answer-only	nothing (answer only)	–	–	no-checkpoint reference
HAIR				
$\text{HAIR}_{b=0}$	checkpoint	0	–	checkpoints alone
$\text{HAIR}_{b=2}$ w/o GBD	checkpoint	2	–	adds latent blocks
$\text{HAIR}_{b=2}$ (full)	checkpoint	2	✓	adds GBD
$\text{HAIR}_{b=1}$, $\text{HAIR}_{b=4}$	checkpoint	1, 4	✓	varies the latent budget

Teacher labels. GSM8K decompositions come from `deepseek-flash` and MATH decompositions from `deepseek-v4-flash-0731`, both at temperature zero with thinking disabled. A record is accepted if it follows the schema and its final value agrees with the reference answer. GSM8K answers are scored numerically; MATH answers are scored with `math-verify` 0.8.0. All configurations were fixed before the multi-seed runs reported here.

Splits. The 500 validation questions are GSM8K-Aug questions outside the 20,000 sampled for training; they serve for configuration decisions and latency profiling. Fresh-500 consists of 500 MATH test problems outside MATH-500, selected by a fixed hash. Outputs that hit a length or block limit or break the checkpoint format are scored as wrong (Appendix H).

Transfer sets. SVAMP uses all 1,000 problems of the public release, none of which appear in our training data; the question is the problem body followed by the question sentence. GSM-Hard uses all 1,319 problems. Its references come from executing program-of-thought solutions, and 419 of them are negative or non-integer, often because the enlarged numbers make the original question ill-posed; we therefore also report the 900 problems with a non-negative integer reference. Both sets use the same verifier and decoding settings as GSM8K.

B TRAINING DATA STATISTICS

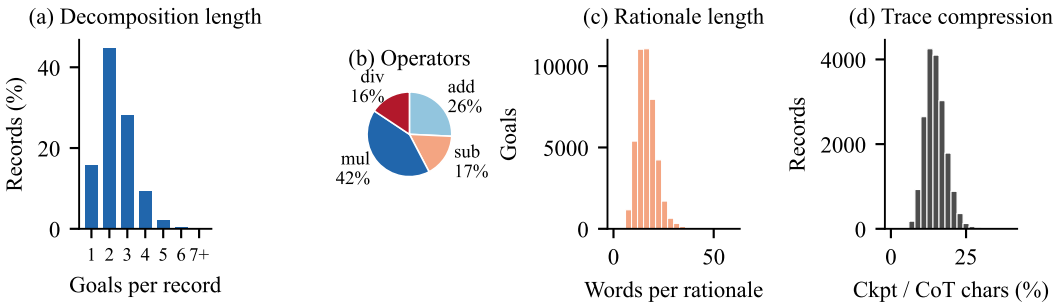

Figure 7: **GSM8K training records** (18,394 records, 43,886 goals). **(a)** Goals per record. **(b)** Operators in checkpoint expressions. **(c)** Words per teacher rationale, which the checkpoint drops. **(d)** Characters of the checkpoint lines relative to the CoT trace of the same record.

Table 10: Summary statistics of the GSM8K training records.

Statistic	Value
Accepted records	18,394 of 20,000 (92.0%)
Goals in total / per record (mean)	43,886 / 2.39
Records with 1 / 2 / 3 / $\geq$4 goals	15.6 / 44.6 / 28.2 / 11.6%
Expressions citing an earlier goal	47.5%
Operators: mul / add / sub / div	42 / 26 / 17 / 16%
Question length (median words)	33
Rationale length (median words)	16
Checkpoint chars relative to CoT (median)	14.5%

Most records have two or three goals, and fewer than 3% have five or more (Figure 7a), which matches the scarcity of long test questions in Figure 6a. Nearly half of the expressions cite an earlier goal, so many steps read a committed result. A rationale has 16 words at the median, while the checkpoint line that replaces it states only the operation and the value; checkpoints have 14.5% of the CoT characters at the median (Figure 7d).

C Examples

A training record. *Question:* Sabrina buys 10 packs of pencils, each containing 12 pencils, for the school year. If she loses 15% of the pencils before the end of the school year, how many pencils does she have left?

- Goal G0. Description: *Find the total number of pencils Sabrina buys.* Rationale: *Multiply the number of packs by the number of pencils per pack to get the total pencils.* Checkpoint: `G0=10*12=120`.
- Goal G1. Description: *Find the number of pencils Sabrina has left after losing some.* Rationale: *She loses 15% of the total pencils, so she keeps 85% of them. Multiply the total by 0.85.* Checkpoint: `G1=G0*0.85=102`, then `#### 102`.

The latent path is trained on the checkpoint lines; the explicit path on descriptions, rationales and checkpoints; explicit CoT on the rationales with `<<10*12=120>>`-style annotations.

Outputs on a test question (seed 29). *Question:* Terry eats 2 yogurts a day. They are currently on sale at 4 yogurts for $5.00. How much does he spend on yogurt over 30 days?

- *Explicit CoT* (105 decoded tokens, 2.81 s): The sale offers 4 yogurts for $5.00, so dividing the total cost by the number of yogurts gives the price per yogurt. `<<5/4=1.25>>` Terry eats 2 yogurts per day, so multiply the daily amount by the number of days. `<<2*30=60>>` Multiply the number of yogurts needed by the cost per yogurt to get the total spending. `<<G1*G0=75>>` `#### 75`
- *HAIR$_{b=0}$ and HAIR$_{b=2}$* (identical text; HAIR$_{b=2}$ adds two latent positions before each line): `G0=2*30=60` / `G1=G0/4=15` / `G2=G1*5=75` / `#### 75`

Both HAIR settings choose a different but valid decomposition from the teacher's (`G1=5/4=1.25`): they count packs rather than computing a unit price. The teacher-value measures of Tables 7 and 16 count this correct output as missing the teacher's `1.25`.

A typical error (HAIR$_{b=2}$, seed 29). *Question:* Judy teaches 5 dance classes, every day, on the weekdays and 8 classes on Saturday. If each class has 15 students and she charges $15.00 per student, how much money does she make in 1 week? *Output:* `G0=5+8=13` / `G1=G0*15=195` / `G2=G1*15=2925` / `#### 2925` (reference 7425, teacher `G0=5*5+8=33`). Every line is arithmetically consistent; the first line computes the wrong quantity, the dominant error type in Figure 6, and the later lines build on it.

D PER-SEED RESULTS AND LATENCY

Table 11: GSM8K test accuracy (%) per training seed. – marks seeds not trained for that setting.

System	11	29	47	53	71	89	Mean
Explicit CoT	59.06	60.42	59.51	59.74	60.58	60.27	59.93
HAIR$_{b=0}$	59.36	59.89	59.74	58.68	60.05	57.09	59.14
HAIR$_{b=1}$	58.68	58.61	59.44	58.98	61.18	–	59.38
HAIR$_{b=2}$	57.85	59.74	57.39	–	58.76	–	58.43
HAIR$_{b=4}$	57.47	58.76	59.14	–	–	–	58.45
HAIR$_{b=2}$ w/o GBD	58.98	58.98	58.91	–	–	–	58.96

Table 12: Same-GPU latency profile (seed 29; 300 validation questions; three counterbalanced rounds).

System	Mean (ms)	SD over rounds	Median (ms)	p95 (ms)
Explicit CoT	2,385	101	2,275	4,050
HAIR$_{b=0}$	1,008	13	953	1,730
HAIR$_{b=1}$	1,115	9	1,051	1,878
HAIR$_{b=2}$	1,204	20	1,123	1,980
HAIR$_{b=4}$	1,396	27	1,298	2,336
HAIR$_{b=2}$ w/o GBD	1,193	15	1,129	2,033

Latency uses the seed-29 models on 300 validation questions after 20 warm-up questions, at batch size 1 in bf16, over three rounds with counterbalanced system order. Validation accuracy on the 500 held-out questions ranges from 77.2% to 81.0% across all cells and does not separate the systems (seed means 78.4–79.9%). Latency is stable across rounds, and a linear fit of per-question latency on sequential positions gives 28.4, 28.5 and 28.8 ms per position for CoT, HAIR$_{b=0}$ and HAIR$_{b=2}$ ($r = 0.991, 0.991, 0.996$).

E ADDITIONAL ANALYSES

All numbers in this section are computed from the saved test predictions of the runs in Table 11. Tables 14 and 15 use the four seeds on which CoT, HAIR$_{b=0}$ and HAIR$_{b=2}$ were all trained (11, 29, 47, 71).

Table 13: **Per-seed version of Table 5** (GSM8K, 1,319 questions). Each cell: accuracy of the second setting minus the first, in points, with the exact McNemar p-value in parentheses; – marks seeds not trained for both settings.

From → to	s11	s29	s47	s53	s71	s89
CoT → HAIR$_{b=0}$	+0.30 (.87)	−0.53 (.74)	+0.23 (.91)	−1.06 (.47)	−0.53 (.74)	−3.18 (.02)
CoT → HAIR$_{b=2}$	−1.21 (.38)	−0.68 (.65)	−2.12 (.14)	–	−1.82 (.19)	–
HAIR$_{b=0}$ → HAIR$_{b=2}$ w/o GBD	−0.38 (.76)	−0.91 (.41)	−0.83 (.45)	–	–	–
HAIR$_{b=2}$ w/o GBD → HAIR$_{b=2}$	−1.14 (.28)	+0.76 (.51)	−1.52 (.17)	–	–	–
HAIR$_{b=0}$ → HAIR$_{b=2}$	−1.52 (.16)	−0.15 (.94)	−2.35 (.03)	–	−1.29 (.26)	–
HAIR$_{b=1}$ → HAIR$_{b=2}$	−0.83 (.44)	+1.14 (.27)	−2.05 (.07)	–	−2.43 (.02)	–
HAIR$_{b=2}$ → HAIR$_{b=4}$	−0.38 (.77)	−0.99 (.39)	+1.74 (.12)	–	–	–

Table 14: **Answer agreement between settings** trained with the same seed (%). "Correctness differs" counts questions that exactly one of the two systems answers correctly. For reference, two $\text{HAIR}_{b=0}$ models with different seeds give the same answer on 59.7% of questions (range 58.1–61.0) and two $\text{HAIR}_{b=2}$ models on 58.3%.

Pair	Measure	s11	s29	s47	s71
$\text{HAIR}_{b=2}$ vs. $\text{HAIR}_{b=0}$	Same final answer	65.7	66.3	64.1	64.4
	Identical generated text	56.3	55.9	53.8	52.7
	Correctness differs	13.6	14.1	14.8	15.1
$\text{HAIR}_{b=2}$ vs. CoT	Same final answer	52.7	52.8	50.4	53.2
	Correctness differs	22.0	24.2	25.3	23.2

With the same seed, $\text{HAIR}_{b=2}$ and $\text{HAIR}_{b=0}$ produce byte-identical outputs on more than half of the questions, so the two latent positions before each checkpoint leave the text unchanged in most cases. Where the outputs differ, $\text{HAIR}_{b=0}$ is right slightly more often (94–113 against 80–92 questions per seed).

Table 15: **Accuracy by the number of calculator steps** in the original GSM8K solution (%, mean over seeds 11, 29, 47, 71). Figure 6a gives the same breakdown by teacher goals.

Calculator steps	Questions	Explicit CoT	$\text{HAIR}_{b=0}$	$\text{HAIR}_{b=2}$
≤ 2	440	76.1	75.4	74.7
3	364	66.8	66.7	65.0
4	290	49.2	49.8	48.5
5	138	37.3	38.6	37.3
≥ 6	87	20.4	18.4	15.5

No stratum shows an advantage of $\text{HAIR}_{b=2}$ over $\text{HAIR}_{b=0}$: of 28 difficulty strata tested (by calculator steps, goals, question length and empirical difficulty), 25 have a negative difference and none is significant after Holm correction over 49 tests. The largest nominal differences are on the longest questions and on solutions with six or more calculator steps, both against $\text{HAIR}_{b=2}$.

Table 16: **Agreement with the teacher decomposition** (%, mean over seeds; 1,236 questions). Intermediates: every teacher value except the final one is written. Last column: answer accuracy when all teacher values are written.

Setting	Step count = teacher	All values written	Inter-mediates	Correct if all values
$\text{HAIR}_{b=0}$	67.5	54.3	62.2	99.0
$\text{HAIR}_{b=1}$	67.4	54.4	62.2	98.8
$\text{HAIR}_{b=2}$	67.9	53.2	61.6	99.2
$\text{HAIR}_{b=4}$	67.7	53.4	61.8	98.8
$\text{HAIR}_{b=2}$ w/o GBD	67.3	54.5	63.1	99.3

Table 17: **Where the first teacher value is missed** (% of the 1,236 questions).

Setting	G0	G1	G2+
$\text{HAIR}_{b=0}$	31.9	12.8	8.7
$\text{HAIR}_{b=1}$	31.9	13.2	8.5
$\text{HAIR}_{b=2}$	31.5	12.9	8.8
$\text{HAIR}_{b=4}$	31.5	13.0	9.1
$\text{HAIR}_{b=2}$ w/o GBD	31.5	12.8	9.0

Most mismatches with the teacher start at the first goal, where the model chooses what to compute, and the distribution of the first mismatch is the same in every setting. The difference between $\text{HAIR}_{b=2}$ and $\text{HAIR}_{b=0}$ comes from emitting all teacher values slightly less often (53.2% against 54.3%), not from converting them into an answer.

F TRAINING COST

Table 18: Training wall-clock time on one RTX 4090D (median over seeds; approximate for resumed runs).

	CoT	$\text{HAIR}_{b=0}$	$\text{HAIR}_{b=1}$	$\text{HAIR}_{b=2}$	$\text{HAIR}_{b=2}$ w/o GBD	$\text{HAIR}_{b=4}$
Hours	2.1	5.7	9.8	14.9	14.8	19.1

The explicit path accounts for most of the difference between CoT and $\text{HAIR}_{b=0}$; the latent blocks, which require recomputing the prefix to propagate gradients through earlier blocks, account for the rest. GBD itself adds no measurable time (14.9 versus 14.8 h), since it reuses states that both paths already compute. $\text{HAIR}_{b=0}$, the most accurate setting per unit of inference cost, is also the cheapest HAIR setting to train.

G EXTERNAL BASELINES

Both baselines use their official repositories with the smallest changes needed to run Qwen3-1.7B-Base on one 24 GB GPU and to read the 18,394 training records; the same numeric verifier scores all systems. **CODI**: six continuous tokens with a projection layer, LoRA rank 128 and $\alpha = 32$, two epochs (about 36.8k example exposures, twice that of our systems), learning rate 8×10^{-4}, distillation factor 20 with standard-deviation normalization, and a 320-token decoding limit. **Coconut**: one epoch of CoT followed by three latent stages of one epoch each, two continuous thoughts per replaced step, LoRA rank 16 and $\alpha = 32$ plus trainable embeddings for the three new special tokens; steps beyond the third remain in text, which explains its 62.7 decoded tokens. Both runs use seed 11. They measure how fully latent methods fare with the same data and backbone; the published numbers use the 385k-example GSM8K-Aug set. Published accuracies are rounded to one decimal; percentages of CoT are computed by us from the unrounded values.

H TERMINATION STATISTICS

Across all GSM8K seeds, HAIR ends with the end-of-sequence token in more than 99.6% of questions. The remaining rollouts hit the 8-block limit (5 of 5,276 for $\text{HAIR}_{b=2}$; 14 of 7,914 for $\text{HAIR}_{b=0}$) or produce a line that does not match the checkpoint format (5 and 4, respectively). Explicit CoT is truncated at 256 tokens in 62 of 7,914 rollouts. On MATH-500, CoT is truncated in 31 of 500 rollouts; $\text{HAIR}_{b=0}$ hits the block or token limit in 25 and $\text{HAIR}_{b=2}$ in 24. All such rollouts are counted as wrong.