LATENT REASONING WORKS: RECOVERING CHAIN-OF-THOUGHT BEHAVIOUR WITH LATENT THOUGHTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Chain of thought (CoT) improves LLM reasoning, but requires a long, autoregressively generated trace, which is subsequently discarded: its value lies in its effect on the distribution over final answers; latent reasoning seeks that effect without the cost. Given the limited success of existing approaches, which do not achieve near-CoT performance at increased efficiency, we ask a more foundational question: do short latent contexts that provide near-CoT performance even exist? We demonstrate that they do: just a single latent token on HumanEval+ recovers 98% of CoT performance and 256 latent tokens even exceeds it on MATH-Hard; by contrast, the median CoT lengths on these datasets are over 1000 and 3000, respectively. We use knowledge distillation, distilling the effect of many CoT rollouts on answers into the latent context in a frozen LLM. Validation loss tracks training loss, indicating that the latents carry some concept of reasoning, not just memorisation.

1 INTRODUCTION

Large language models (LLMs) continue to enjoy performance gains from scaling model size, training data and compute, albeit with diminishing returns (Kaplan et al., 2020). Furthermore, transformers face inherent limitations on tasks requiring serial computation which increasing model size alone cannot resolve. Chain of thought (CoT) offers an additional axis along which computation can be scaled by providing LLMs with a textual scratchpad (Nye et al., 2021; Wei et al., 2022a). This allows the model to explore multiple solution strategies, verify intermediate steps and adapt its approach before committing to a response (Wei et al., 2022b; Li et al., 2024; DeepSeek-AI, 2025).

These benefits come with an increase in inference cost. The generated "reasoning trace" may be orders of magnitude longer than the response itself (Marjanović et al., 2025). Moreover, these tokens are generated auto-regressively, leading to increased response latency.

The trace itself, however, is rarely the object of interest. Production systems rarely expose them (OpenAI, 2024), and most evaluation methods only assess the final answer (Prasad et al., 2023)—indeed, the scratchpad should be akin to 'rough workings', not 'answers'. Even the extent to which traces faithfully represent the model's underlying reasoning remains disputed (Chen et al., 2025). Instead, the trace's value derives from its effect on the distribution over final answers.

Latent representations have long enabled the capture of underlying concepts and abstract features efficiently, beyond human-interpretable forms (Bengio et al., 2013). In this vein, latent reasoning aims to elicit the CoT effect more efficiently (with fewer tokens). Training latent-reasoning models from scratch at a scale comparable to modern LLMs is prohibitively expensive for research, so existing methods adapt off-the-shelf LLMs (Zhu et al., 2025): they are either training-free architectural 'patches' or fine-tune the LLM's weights. The former replace the discrete CoT with a continuous one, but their lengths are the same order of magnitude. The fine-tuning does achieve short sequences, but changing the model itself risks catastrophic forgetting (Xu et al., 2025b), and confounds: is the dataset-specific fine-tuning doing the work or are the latents (Rizvi-Martel et al., 2026)?

Crucially, these methods address *generation*. Shortfalls cannot cleanly be attributed to a generation failure: compact latent thoughts simply may not exist in this case. To our knowledge, no prior work separates these cases. Given the lack of success in generation, we ask the foundational question:

Do latent thoughts which replicate CoT performance actually exist at a short scale?

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.

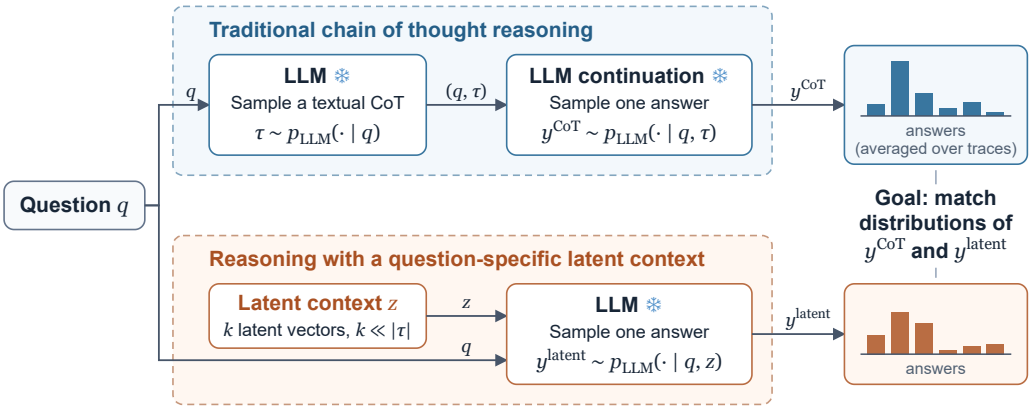


Figure 1.1: For each question q, we desire latent thoughts z such that the LLM answering immediately after q followed by z outputs answers with the same distribution as when a (random) chain of thought follows q. Efficient generation of such z directly from the question is left to future work.

This objective is visualised in Figure 1.1. We emphasise that we use the same, frozen LLM throughout. The qualification is necessary: discrete CoT is a special case of latent reasoning, but not "short".

To demonstrate the existence of such latent thoughts, we use knowledge distillation. The teacher and student use the same frozen LLM backbone: the teacher first generates many, independent answers via textual CoTs; the student has the latent thoughts injected in place of the CoT. These are trained, per question, via a teacher-forced loss; the goal is to reproduce the teacher's *distribution* over answers, not simply the *training bank*. We are not proposing an inference-time algorithm: it requires (many samples of) the very traces that it replaces. Rather, it is a means of disentangling the questions of *existence* and *generation*. Our contributions are positive, indicating that such latents do exist.

- **Existence.** We define precisely what it means for latents to "replicate CoT performance"; see §3. Using frozen, off-the-shelf LLMs (Qwen3), we construct (per-question) latent thoughts which recover, or even exceed, the accuracy gain due to CoT across maths and code, at 1.7B and 8B model sizes and at a fraction of the length; see §4.2. Beyond accuracy, held-out loss tracks training loss, ruling out memorisation of training data; see §4.1.
- **Reasoning capacity.** Utilising question augmentations, we demonstrate that the latent thoughts are not merely storing a compressed version of the answer; see §5. They carry some form of reasoning—eg, how to solve a quadratic, rather than a particular factorisation.

Together, these results establish that latent thoughts which recover the benefit of CoT at a fraction of the length do indeed exist. This supports latent reasoning as a promising direction and provides a reference point for methods that generate such representations at inference time.

2 Related Work

Latent reasoning work has concentrated on how to *generate* latent thoughts. The lack of established definitions and goals limits cross-approach comparison: each emits a slightly different object, embedding in a different pipeline, so end-task accuracy reflects the whole system; isolating the latent-thought contribution is not possible. A definition by effect, specifying what the representation must achieve, is implementation-independent, removing generative versus existence confounding.

Scratchpads and discrete CoT. Giving a model somewhere to write before answering has long helped (Nye et al., 2021; Wei et al., 2022a). Filler tokens (Pfau et al., 2024) supply positions without content; Abstract CoT (Ramji et al., 2026) learns a codebook of M symbols, extending the earlier pause tokens (Goyal et al., 2023) as its $M = 1$ case. Each was trained into the model, so set the scale of what a few discrete tokens buy, not a bound on what continuous ones could.

Latent CoTs. Latent reasoning feeds a continuous vector into the next decoding step in place of a sampled token's embedding; approaches differ in what vector is input. Coconut (Hao et al., 2024),

CODI (Shen et al., 2025) and CCoT (Cheng & Durme, 2024) obtain it by fine-tuning the model or its adapters; Soft Thinking (Zhang et al., 2025a) and Mixture-of-Inputs Zhuang et al. (2025a) do so with no training, simply averaging token embeddings with respect to the LLM's softmax distribution, whilst SoftCoT(++) (Xu et al., 2025b;c) trains only a projection into a frozen backbone.

Another strand trains the models with RL: SofT-GRPO (Zheng et al., 2025) and MoT-G (Jain & Rappazzo, 2025) optimise the policy over soft tokens; HRPO (Yue et al., 2025) gates hidden states into the embedding stream; in "Soft Tokens, Hard Truths" (Butt et al., 2025), noise is added to the mixture to enable RL training, yet hard (discrete) tokens are optimal at inference time. Recurrent-depth models (Geiping et al., 2025), trained from scratch, instead iterate a block in depth.

In each case, success is judged by the end-task accuracy of the whole pipeline. Whether the vectors themselves do the work of reasoning is separate and, when investigated, the answer is often "no"—particularly around the "parallel reasoning" narrative (Rizvi-Martel et al., 2026; Wu et al., 2025).

Furthermore, these methods do not settle whether *compact* latent thoughts exist. The low-/no-training approaches do not reduce substantially compared with discrete CoT—often $10\text{k}+$ tokens. The fine-tuning ones require only a short (latent) CoT, typically 10s of tokens, but the training confounds matters: was the model already able to use latents before training? Implicit CoT (Deng et al., 2023) makes this clear: it answers directly from latents, but only by training a student to read them.

Prompt tuning and existence. Optimising a continuous prefix against a frozen model is prompt tuning Lester et al. (2021); Li & Liang (2021). For off-the-shelf LLMs, (Wang et al., 2025) is the closest to an existence study: that asks which distribution over the next token a soft of hard prompt can elicit from a frozen model. Our question has the same form, but a different, much wider, target: a distribution over entire answers, induced by the model's own reasoning.

Directly optimising the latent. Closest to our approach are methods that freeze the model and optimise a latent (per question) at inference time: LTPO (Ye et al., 2025) maximises the model's internal confidence over the latents alone, in a single forward pass; LatentSeek (Li et al., 2025) and GradCuit (Yu et al., 2026) score a full CoT response. Post-optimisation, each then decodes a full, discrete CoT: this both prohibits substantial efficiency gains, as the CoTs are of similar length to the original traces, and does not test whether the latent context can carry the effect of a CoT on its own.

Further related approaches. Defining a compact representation by its effect is common in context compression: context distillation (Snell et al., 2022) fine-tunes a model to behave as it would with a given context; gist tokens (Mu et al., 2023) and Cartridges (Eyuboglu et al., 2025) instead compress the context into a few soft tokens or a trained prefix; Wingate et al. (2022) optimise a soft prompt by gradient descent to stand in for a single discrete prompt. Each has the same behavioural target, of replicating the effect of a specific prompt.

We apply the same standard to reasoning: the key difference is that, in CoT, each answer is generated with its own trace in context; we target the distribution of answers *marginalised* over traces.

3 Defining and Constructing Latent Thoughts

In this section, we define our exact goal for latent reasoning—in short, replicating the CoT behaviour at a compressed scale—and detail the process that we use to establish existence in this paper.

3.1 Constructing Latent Thoughts by Knowledge Distillation

Latent reasoning allows a more abstract encoding of reasoning. We thus evaluate latent thoughts not by what they contain or by how they are produced, but instead by their effect on the model's output.

Our aim is to establish the existence of latent thoughts which replicate the effect of CoT on the answer distribution. This is distribution learning, for which there are many techniques. We pick (arguably) the simplest: knowledge distillation. This avoids training an auxiliary model for adversarial training or relying on exploration alone in reinforcement learning, instead using teacher guidance.

We freeze an LLM backbone p. For a question q, the *teacher* first samples a CoT τ before answering whilst the *student* answers immediately after a latent context z is injected; $z \in \mathbb{R}^{k \times d}$ is a sequence

of k vectors in the model's embedding space. We say that z *replicates (the effect of) CoT* if

$$\mathrm{S}_{q,z}(y) \coloneqq p(y \mid q \oplus z) \approx \textstyle\sum_\tau p(\tau \oplus y \mid q) =: \mathrm{T}_q(y), \tag{3.1}$$

where $p(x \mid x')$ is the probability of generating x given input x' and $\oplus$ denotes sequence concatenation. The objective is to do so at a scale shorter than the (median) trace that it replaces.

Of course, this will never be exact. The assessment is whether the student, when generating freely, behaves similarly to the teacher; the primary metric of interest is benchmark accuracy.

We emphasise that, in the standard CoT set-up, the goal is not to distil the content of a single trace into the latent context z. Rather, z should encode the change in distribution induced by the CoT. Practically, we do this by performing many (full) rollouts to approximate the teacher distribution T_q.

All LLMs in this paper are frozen. We refer to the LLM into which the (trained) latent context is injected as the *backbone*; the term *student* always refers to this backbone with the latent context.

Shortening a textual trace is an active line of work in its own right (Xu et al., 2025a; Xia et al., 2025). Latent reasoning is more general, replacing the discrete tokens with arbitrary embedding vectors.

3.2 Distillation Loss

Having set the goal, we need a way to find a latent context with the desired property. First, we need to make the "$\approx$" in (3.1) concrete. For a single summary of closeness, we use the KL divergence:

$$\mathcal{D}_q(z) \coloneqq \mathrm{KL}(\mathrm{T}_q \,\big\|\, \mathrm{S}_{q,z}). \tag{3.2}$$

The natural loss would be to estimate $\mathcal{D}_q(z)$ directly using Monte Carlo over CoT rollouts. However, this is computationally intractable. Fortunately, Monte Carlo estimation of an equivalent statistic is tractable: let $\mathcal{R}$ denote a (multi)set of rollouts of the form (τ, y) for question q, and define

$$L_q(z;\mathcal{R}) \coloneqq \tfrac{1}{|\mathcal{R}|} \textstyle\sum_{(\tau,y)\in\mathcal{R}} \sum_{j=1}^{|y|} \mathrm{KL}\big(p^{\text{next}}(\cdot \mid q \oplus \tau \oplus y_{<j}) \,\big\|\, p^{\text{next}}(\cdot \mid q \oplus z \oplus y_{<j})\big); \tag{3.3}$$

the superscript "next" indicates that the distribution is over the next token only. This is precisely the Monte Carlo estimator of the population-level $\mathcal{D}_q(z)$ but shifted by an additive constant, and hence is equivalent from a gradient perspective; a full derivation of the equivalence is given in §B.1.

Note that the loss *sums*, not *averages*, over token positions in the answer. Thus, all tokens in all rollouts are given equal weight, independent of the length of their particular rollout.

3.3 Experiment Set-Up

We evaluate our distillation procedure using Qwen3-{1.7B, 8B} models (Qwen Team, 2025). Qwen3 utilises `<(/)think>` tags: the trace τ is the content between the tags; the answer y is everything generated after the closing tag. Student behaviour is evaluated over a sweep of latent-token lengths: $k \in \{1, 4, 16, 64, 256\}$; the backbone without CoT corresponds to $k = 0$.

Our primary experiments in §4 focus on two datasets: HumanEval+ (Liu et al., 2023) for code writing (all 164 questions) and MATH-Hard (Hendrycks et al., 2021) for mathematical reasoning (128 randomly-sampled questions); for each, 2048 teacher rollouts are sampled per question. In §5, we use GSM-Symbolic (Mirzadeh et al., 2025) and train a single latent context across multiple augmented versions of a question; 1000 rollouts are sampled, generated from different augmentations.

All rollouts—both teacher data and the student's free generation—are sampled at temperature 0.7. Optimisation uses Adam (Kingma & Ba, 2015) with batch size 32, weight decay 10^{-2} and 2000 optimisation steps in §4. A learning-rate sweep was conducted across {HumanEval+, MATH-Hard} × Qwen3-{1.7B, 8B}, and a learning rate of the form $\eta(k) = a + b/k^c$ was fitted. This lead to $a = 0$ (no intercept) and $c = 1$ (linear decay), but b depending on the dataset and model size; see Table A.1.

4 Recovering Chain-of-Thought Behaviour

Our aim is to establish that a latent context z can replicate CoT behaviour. Whilst rollout accuracy is the primary metric of interest to the user, we need to look beyond this: a latent context that

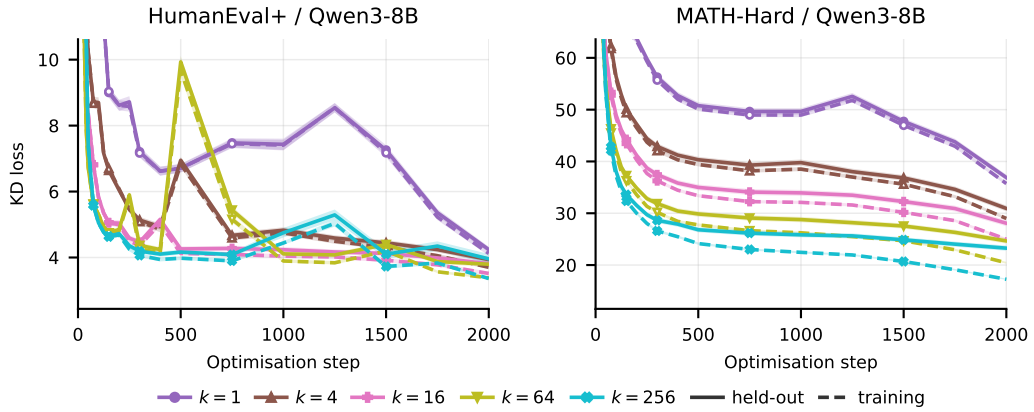


Figure 4.1: Training/validation loss (*dashed*/*solid*) against optimisation step for Qwen3-8B on HumanEval+/MATH-Hard (*left*/*right*). Different colours/markers indicate different numbers k of latent thoughts. Shaded regions indicate ± 2 standard errors, treating the questions as fixed. The analogous results for Qwen3-1.7B are displayed in Figure D.1 in §D.1.

simply encodes "sample a rollout from the training data" trivially reproduce the teacher's (empirical) accuracy exactly; however, this is a degenerate solution, carrying nothing that generalises.

Such 'memorisation' is perfectly acceptable in the case of low-diversity datasets: if the training bank effectively exhausts the teacher's entire support, then there is no space to 'generalise into'; replicating the training data *is* replicating the teacher distribution. This is the case for a substantial proportion of HumanEval+, for which many questions only have 100s of unique answers; the median number per question around 150/300 for Qwen3-1.7B and 70/200 for Qwen3-8B. Conversely, almost all answers are unique in MATH-Hard. Memorisation really is degenerate there: future teacher answers rollouts would be unseen; the student would be stuck on the training bank.

This diversities are perhaps not surprising: MATH-Hard is prompted to generate a worked solution (§C.2), at around 650 tokens; HumanEval+ requests only a Python script (§C.1), at around 50 tokens.

We therefore proceed in two stages. First, §4.1 evaluates the distillation loss on held-out teacher rollouts. This is the key metric for identifying memorisation: such degenerate z will perform poorly on held-out rollouts when the training bank does not exhaust the teacher support. Second, §4.2 looks at the student's own free generation, and is compared with the teacher's via three resolutions.

- Accuracy, which collapses a rollout to a binary via the question's verifier.
- Similarity, which compares pairs of complete rollouts.
- Plagiarism, which compares n-gram character-overlap with the training bank.

The student could match the teacher at a coarse level (eg, same accuracy) but differing at a finer one (eg, minimal n-gram overlap). This could happen if the student merely memorised the training data.

4.1 Optimising Latent Thoughts via Teacher-Forcing

This section evaluates the student on the teacher's rollouts, not what it freely generates itself. Our per-question optimisation process frequently has 10^5 parameters, whilst we sample 2048 rollouts per question; so, overfitting is a real possibility. The held-out loss is precisely the diagnostic for this. Namely, we draw (and fix) two (multi)sets of 128 rollouts: $\mathcal{R}_q^{\text{trn}}$ uniformly from the training set; $\mathcal{R}_q^{\text{val}}$ iid from the target distribution T_q (anew). We evaluate the loss $L_q(z; \mathcal{R}_q^{\text{trn/val}})$ at checkpointed z.

These losses for HumanEval+ and MATH-Hard using Qwen3-8B are displayed in Figure 4.1; the analogous results, with the same trends, for Qwen3-1.7B are displayed in Figure D.1 in §D.1.

We do not observe overfitting: rather, the validation loss follows the shape of the training loss closely throughout optimisation; on MATH-Hard, a gap does start to open near the end of training,

Table 4.2: Failure modes that the student could exhibit, and how each statistic responds. Each entry compares the student with the teacher, except cross-similarity, which is compared with both teacher and student self-similarities. *Italicised* marks highlight particularly insightful signals in each row.

If the student...	**accuracy**	**self-similarity**	**cross-similarity**	**loss gap**
memorises the training answers	matches	matches	equal to both	*large*
replaces teacher answers with new ones	unknown	unknown	*below both*	large
adds new answers on top of teacher's	unknown	*below*	between	small

but the validation loss is still decreasing. This indicates that the latent context genuinely captures the teacher's *distribution*, not just its training data. Accuracy alone cannot determine this.

Overfitting would also cast doubt on other results: no strong conclusion—for example, around mechanism—could be claimed if the objective that our loss encodes (to match the teacher) failed.

For MATH-Hard, the low k-values' validation loss, is still improving, suggesting that more training would help. It is quite possible that this would flatten the MATH-Hard curve in Figure 4.3 below.

All this was evaluation on teacher data, largely to test for memorisation. We now turn to evaluating the student on its own, freely-generated data—namely, its rollout accuracy and similarity metrics.

4.2 Free-Generation Using Latent-Injected Student

We now compare the student's free generation with the teacher's at three resolutions: rollout accuracy; similarity of pairs of rollouts; a plagiarism-type metric comparing with the training bank.

For a question $q \in \mathcal{Q}$ and $p, p' \in \{\mathrm{T}_q, \mathrm{S}_{q,z}\}$, draw $Y \sim p$ and $Y' \sim p'$ independently, and define

$$\mathcal{G}_q(p, p') \coloneqq \mathbb{P}[Y = Y']. \tag{4.1}$$

We define *self-similarity* to be when $p = p'$ and *cross-similarity* when $p \neq p'$. The self-similarity is the complement of the Gini–Simpson index (Österreicher & Casquilho, 2018). We use the following abuse of notation to average this across questions, drawing $Q \sim \mathrm{Unif}(\mathcal{Q})$:

- self-similarity $\to \mathcal{G}(\mathrm{T}, \mathrm{T}) \coloneqq \mathbb{E}[\mathcal{G}_Q(\mathrm{T}_Q, \mathrm{T}_Q)]$ and $\mathcal{G}(\mathrm{S}_z, \mathrm{S}_z) \coloneqq \mathbb{E}[\mathcal{G}_Q(\mathrm{S}_{Q,z}, \mathrm{S}_{Q,z})]$;
- cross-similarity $\to \mathcal{G}(\mathrm{T}, \mathrm{S}_z) \coloneqq \mathbb{E}[\mathcal{G}_Q(\mathrm{T}_Q, \mathrm{S}_{Q,z})] =: \mathcal{G}(\mathrm{S}_z, \mathrm{T})$.

Table 4.2 shows how these statistics are used to diagnose failure modes that the student may exhibit.

It highlights how the accuracy carries no signal to identify these failures. The held-out loss identifies, though cannot distinguish, the first two, but misses the last, being near-blind to mass outside the teacher's support; the cross-similarity is the only metric that can distinguish the first two failures.

Accuracy. Improved accuracy is the primary downstream benefit of CoT, yet it is also the coarsest resolution at which we assess the latent context. It reduces each rollout to a binary pass/fail outcome.

We report the $\mathrm{pass@1}$ metric. The full set {HumanEval+, MATH-Hard} × Qwen3-{1.7B, 8B} of results are displayed in Figure 4.3. The results indicate comprehensive recovery of CoT performance.

- *HumanEval+.* Even $k = 1$ latent thought recovers *almost* the full CoT performance.
- *MATH-Hard.* Conversely, there is a strong upward trend as k increases, with $k = 1$ falling substantially short of CoT performance, yet $k = 256$ actually *exceeding* it.

Although the student beats the teacher on MATH-Hard but not on HumanEval+, it actually wins on more questions for HumanEval+, just by a smaller margin. This is shown in Table D.2. This may be due to the higher diversity of answers in MATH-Hard requiring more capacity to represent.

Therefore, at the coarsest resolution, the effect of CoT is recovered on both datasets and at both model sizes, even exceeded on MATH-Hard at $k = 256$. The finer-resolution investigations ask whether to what level it has successfully recovered the behaviour of the teacher to do so.

Further results, addressing rollouts for which the teacher ran out of context, are displayed in §D.2.

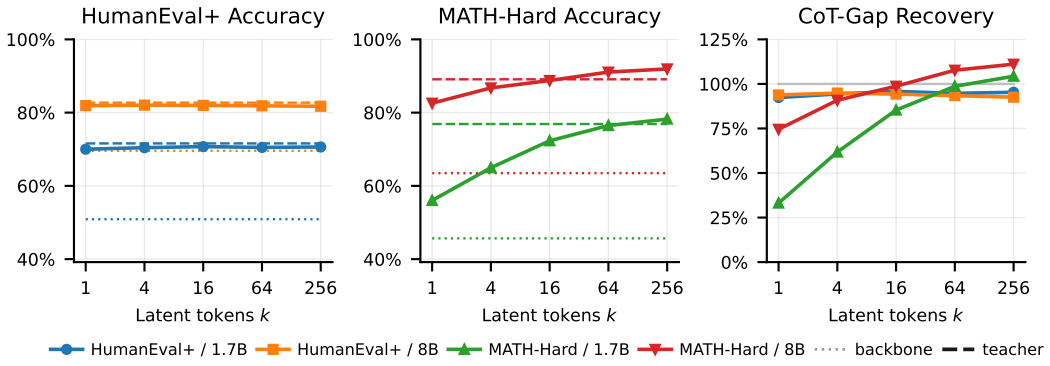


Figure 4.3: Accuracy (pass@1) against number k of latent thoughts for the full set {HumanEval+, MATH-Hard} × Qwen3-{1.7B, 8B}. Horizontal lines indicate baselines: the (no-CoT) backbone and the (CoT) teacher. Shaded regions indicate ± 2 standard errors, treating the questions as fixed; each checkpoint has at least 1024 rollouts per question, rendering the error bars too small to see.

Similarity. Similarity is the intermediate resolution that we consider: self-similarity asks how often two answers generated by the same model (teacher/student) are the same; cross-similarity compares the student's and teacher's. Two students with equal accuracy, equal to the teacher's, may differ here: they may sample from disjoint subsets of the teacher's support, which are equally correct, but one may be substantially more concentrated than the other, giving a higher self-similarity.

The goal is not to *maximise* the student's (self-)similarity, but to *match* the teacher's. This can diagnose two failures modes. First, if the student concentrates on a subset of the training bank, then $\mathcal{G}(\mathrm{S}_z, \mathrm{S}_z) > \mathcal{G}(\mathrm{T}, \mathrm{T})$; this can decrease or increase the accuracy. Second, if it covers the teacher's support, but also extends further, then $\mathcal{G}(\mathrm{S}_z, \mathrm{S}_z) < \mathcal{G}(\mathrm{T}, \mathrm{T})$. The cross-similarity diagnostic is more subtle. It diagnoses when the student *replaces* some of the teacher's answers with others; self-similarities may not detect this, as it changes *which* answers are given, rather than how concentrated they are. We compare cross-similarity $\mathcal{G}(\mathrm{T}, \mathrm{S}_z)$ with *both* self-similarities, $\mathcal{G}(\mathrm{T}, \mathrm{T})$ and $\mathcal{G}(\mathrm{S}_z, \mathrm{S}_z)$.

- *Below both.* The student *replaced* some teacher answers: for both sources, answers match within more often than across. Accuracy and self-similarity are both blind to this.
- *One either side.* For exactly one source, its answers match independent samples from the other more often than from itself; that source is necessarily the more diverse. Whether the student has spread or concentrated is identified by the self-similarities.
- *Above both.* Cauchy–Schwarz excludes this: $\mathcal{G}(\mathrm{T}, \mathrm{S}_z) \leq \sqrt{\mathcal{G}(\mathrm{T}, \mathrm{T})\mathcal{G}(\mathrm{S}_z, \mathrm{S}_z)}$; see §B.2.

The statistics are almost constant in $k \geq 1$. So, for simplicity, we display only its (geometric) average. We also include the no-CoT backbone, denoted B, to compare $k > 0$ versus $k = 0$.

All these data for HumanEval+ with Qwen3-8B are displayed in Figure 4.4. The analogous results for the rest of {HumanEval+, MATH-Hard} × Qwen3-{1.7B, 8B} are in Figure D.5 in §D.3.

The student's self-similarity (0.259) sits just below the teacher's (0.303), with its cross-similarity (0.272) between the two. This is the *added* case from Table 4.2: the student covers the teacher's answers closely, but is slightly *more* diverse. The backbone ($k = 0$), by contrast, is far more repetitive (0.703), yet overlaps the teacher and student ($k > 0$) by only a quarter as much as it could.

Plagiarism. The similarity metric treats the whole answer as a single object. A student that combines memorised fragments of its training rollouts, without repeating any one in full, passes this.

The same issue arises in diagnosing plagiarism in prose. We use the same test: n-gram character-overlap with a corpus, here the training bank of 2048 teacher rollouts. Precisely, within a rollout, we record the proportion of characters which are covered by some n-gram in the training bank.

The goal of a plagiarism checker is to catch overlap. Contrastingly, overlap indicates that the student is learning to mimic the teacher, in a more refine way than crude equality test.

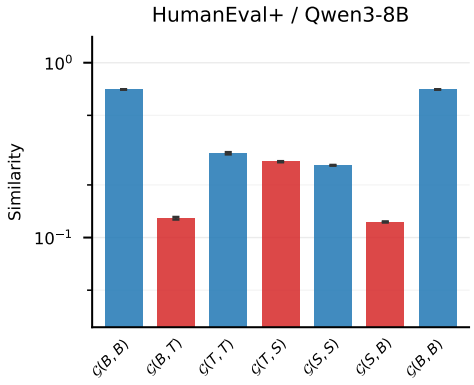


Figure 4.4: Teacher/student/backbone similarities for Qwen3-8B on HumanEval+. Whiskers indicate ± 2 standard errors as usual.

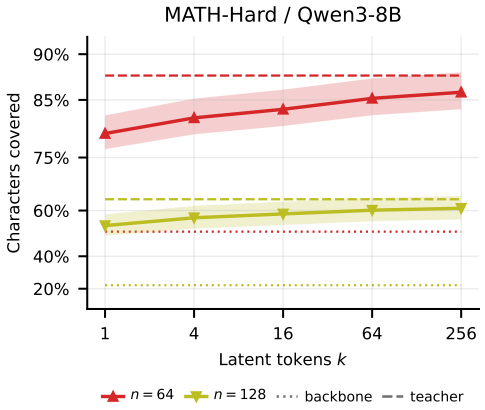


Figure 4.5: 64-/128-gram coverage against k for Qwen3-8B on MATH-Hard. Shaded regions indicate ± 2 standard errors as usual.

The n-gram coverage for MATH-Hard with Qwen3-8B is displayed in Figure 4.5. The analogous results for the rest of {HumanEval+, MATH-Hard} × Qwen3-{1.7B, 8B} are in Figure D.6 in §D.4.

The student is below the teacher, with the backbone substantially further away. This indicates that the student outputs longer (eg, 64-character) snippets that are in the training bank almost as often as (fresh generations from) the teacher, but the backbone (no latent thoughts) does not. Pure memorisation of training rollouts would suggest a *higher* character-overlap than the teacher's—even if it could pass the crude equality test by patching together parts of answers.

5 Latent Thoughts: More Than the Sum of Their Training Parts

The results of §4 definitively establish that a latent context can effectively recover the teacher's behaviour (distribution), and that it can do so without simply memorising its training data.

A discrete CoT answers just the question that was asked; it does not (typically) carry general arguments. Eg, if asked to solve $x^2 - 3x + 3 = 0$, whilst the CoT would likely apply the quadratic formula, it is unlikely to write out the general version. Our latents are trained in the same way.

In this section, we address this more 'general reasoning' concept, using data augmentation. Specifically, we train a single latent across multiple versions of the same question—eg, in Python f-string syntax, `f"Solve ${a} x^2 + {b} x + {c} = 0$"` for multiple choices of (a, b, c)—and evaluated it on *unseen* versions. The hope is that the latent carries some general concepts, rather than just the solution to the special case. Indeed, memorising all training versions wouldn't tell it how to answer the (unseen) evaluation version. This is just the usual concept of generalisation in ML.

We use the GSM-Symbolic (Mirzadeh et al., 2025) family of datasets, run on Qwen3-1.7B. These generate multiple instantiations of the GSM8K (Cobbe et al., 2021) problem templates by varying names, entities and numerical values; its P1 and P2 variants additionally increase the difficulty by introducing one or two clauses (independently), respectively. A concrete example of all four versions (Base, Symbolic, P1 and P2), showing the ramp up, is spelled out in §C.3.

We view each augmentation-type as forming its own dataset. For each type, the latents are trained and evaluated using augmentations of that type. We do not evaluate cross-type performance here.

We select, and fix, 24 questions, each with 50 augmentations of each type; we use 25 augmentations for training and hold out 25 for evaluation. We sweep over the number $a \in \{1, 2, 4, 8, 16\}$ of training augmentations per questions, chosen randomly each time.

The data is plotted in Figure 5.1. The underlying *questions* are fixed—they from the benchmark—but *augmentations* are sampled; hence, per question, the cross-augmentation variance must be taken into account. Each held-out augmentation has 40 rollouts for evaluation, to estimate this.

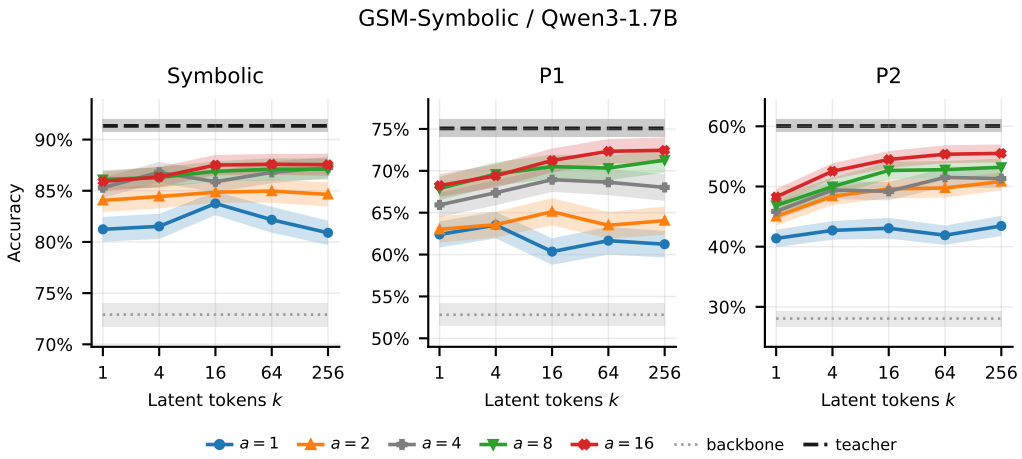


Figure 5.1: Accuracy against number k of latent thoughts for GSM-Symbolic on Qwen3-1.7B. Horizontal lines indicate baselines: the (no-CoT) backbone and the (CoT) teacher. Shaded regions indicate ± 2 standard errors, treating the *questions* as fixed but *augmentations* as sampled.

The results are broadly positive. The clearest trend is that the number $a \in \{1, 2, 4, 8, 16\}$ of training augmentations per question helps. This really suggests that the latents are absorbing more 'context' from exposure to more examples, and that this genuinely enables cross-augmentation learning.

The correlation with the number k of latent tokens is weaker. It can help, but mainly at higher augmentation-per-question counts and for the cases P1 and P2 where extra clause(s) are added. Perhaps the many augmentations are able to utilise the extra capacity for this 'context'.

We plot the final-checkpoint validation losses in Figure D.8. The same trends are exhibited there.

An interpretation that we can't rule out is that the larger capacity is used simply to learn how to solve all 16, say, variants of the question individually. Performance increases is then akin to when an LLM is shown multiple worked examples alongside the question. All these mechanistic interpretations are speculative: they should neither be read as thoroughly investigated nor as concrete conjectures.

Regardless of the interpretation, this is not a property that single, discrete CoTs exhibit: by default, they carry information specific to the prompt provided. Concatenating 16 CoTs, to replicate the "multiple examples" concept, would blow up the context length and destroy inference efficiency.

6 Conclusions, Future Work and Limitations

Latent reasoning promises efficient gains, but is yet to breakthrough. Given the myriad of approaches in recent times, with limited success, we took a step back and asked the foundational question:

> *Is it even possible to replicate CoT behaviour with short latents?*

We answered this positively in §4. On HumanEval+ (code generation), only a single latent token is required to recover CoT accuracy; on MATH-Hard (maths), CoT accuracy is recovered with 64 latent tokens, and even exceeded with 256. Contrastingly, discrete CoTs on these datasets are typically over 1000 and 3000, respectively—at least an order of magnitude larger.

Using data augmentation in §5, we showed that the latent context can carry 'reasoning' that generalises to provide performance gains on unseen augmentations. Standard, discrete CoTs cannot.

We see this as perhaps the most exciting avenue for future work. How far can this generalisation extend? How much are the latents tied to the question style? Can discrete CoT be adjusted (prompted) to carry such information as well? If so, does it (further) bloat the length substantially?

The limitation is that the work establishes existence, but not a (useful) inference-time algorithm. The findings and understanding gleaned from this investigation into existence should be used to inform the design of an inference-time algorithm in the future. This is the natural next major step.

7 AI-USE, ETHICS AND REPRODUCIBILITY STATEMENTS

7.1 AI-USE STATEMENT

In this work, we used generative AI tools to assist with the following tasks:

- provided feedback on research methodology or experiments;
- implement methods (ie, assist with code planning and writing);
- support qualitative and thematic data analysis;
- create or modify scientific figures or images (ie, write code to generate figures/tables);
- draft parts of a research paper (ie, suggest content/direction; rarely copy–paste);
- summarise or analyse existing literature (particularly for "Related Work");
- edit a research paper to improve readability (again, suggest edits, rather than copy–paste);
- identify relevant literature (particularly for "Related Work").

Generative AI was not used for the remaining tasks in the required/recommended sections, some of which are not applicable to this work.

We have reviewed all AI-assisted work, particularly around citation of other works and any load-bearing graphics; whilst all code has some bugs, LLM-generated code was tested for correctness.

We take full responsibility for the final content of this work, including text, claims (about our work and others') and any artefacts produced, whether with the aid of generative AI or not.

7.2 ETHICS STATEMENT

The purpose of this work is to demonstrate the viability of latent reasoning, which has the goal of improving LLM efficiency. The advancement of AI has its own, well-discussed ethical concerns; these broad-strokes concerns apply to our work as much as any other AI-advancement work. Beyond these, the specifics of our work raise no further concerns, such as use of human subjects, releasing datasets, potential conflicts of interest or discrimination.

7.3 REPRODUCIBILITY STATEMENT

All our code is archived on GitHub. The final 'hero run', from which the main results are generated, is documented on a specific branch. This will all be released with the paper, after de-anonymisation. The data (eg, rollouts and checkpoints) was preserved in a clear tree structure to enable auditability.

REFERENCES

Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation Learning: A Review and New Perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8):1798–1828, 2013. doi: 10.1109/TPAMI.2013.50. URL https://doi.org/10.1109/TPAMI.2013.50.

Natasha Butt, Ariel Kwiatkowski, Ismail Labiad, Julia Kempe, and Yann Ollivier. Soft Tokens, Hard Truths, September 2025. URL http://arxiv.org/abs/2509.19170. arXiv:2509.19170.

Mark Chen, Jerry Tworek, Heewoo Jun, et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021. doi: 10.48550/arXiv.2107.03374.

Yanda Chen, Joe Benton, Ansh Radhakrishnan, Jonathan Uesato, Carson Denison, John Schulman, Arushi Somani, Peter Hase, Misha Wagner, Fabien Roger, Vlad Mikulik, Samuel R. Bowman, Jan Leike, Jared Kaplan, and Ethan Perez. Reasoning Models Don't Always Say What They Think, 2025. URL https://arxiv.org/abs/2505.05410.

Jeffrey Cheng and Benjamin Van Durme. Compressed Chain of Thought: Efficient Reasoning Through Dense Representations, December 2024. URL http://arxiv.org/abs/2412.13171. arXiv:2412.13171.

Xinyuan Cheng, Beiduo Chen, Philipp Mondorf, and Barbara Plank. Reasoning that travels: Dissecting how chain-of-thought transfers across models, 2026. URL https://arxiv.org/abs/2605.28913. arXiv preprint arXiv:2605.28913.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021. doi: 10.48550/arXiv.2110.14168.

DeepSeek-AI. DeepSeek-R1: Incentivizing Reasoning Capability in LLM via Reinforcement Learning. *Nature*, 645:633–638, 2025. doi: 10.1038/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, November 2023. URL https://arxiv.org/abs/2311.01460. arXiv:2311.01460.

Sabri Eyuboglu, Ryan Ehrlich, Simran Arora, Neel Guha, Dylan Zinsley, Emily Liu, Will Tennien, Atri Rudra, James Zou, Azalia Mirhoseini, and Christopher Re. Cartridges: Lightweight and general-purpose long context representations via self-study, June 2025. URL https://arxiv.org/abs/2506.06266. arXiv:2506.06266.

Jonas Geiping, Sean McLeish, Neel Jain, John Kirchenbauer, Siddharth Singh, Brian Bartoldson, Bhavya Kailkhura, Abhinav Bhatele, and Tom Goldstein. Scaling up Test-Time Compute with Latent Reasoning: A Recurrent Depth Approach. In *Advances in Neural Information Processing Systems*, volume 38, 2025. doi: 10.52202/085713-1380. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/3b01972cf31e6fa0fe29e4b8b5c2a0a1-Abstract-Conference.html.

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, October 2023. URL https://arxiv.org/abs/2310.02226. arXiv:2310.02226.

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, December 2024. URL https://arxiv.org/abs/2412.06769. arXiv:2412.06769.

Shibo Hao, Sainbayar Sukhbaatar, DiJia Su, Xian Li, Zhiting Hu, Jason E. Weston, and Yuandong Tian. Training large language models to reason in a continuous latent space. In *Second Conference on Language Modeling*, 2025. URL https://openreview.net/forum?id=KrWSrrYGpT.

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.

Adit Jain and Brendan Rappazzo. Learning to reason with mixture of tokens, September 2025. URL https://arxiv.org/abs/2509.21482. arXiv:2509.21482.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling Laws for Neural Language Models, 2020. URL https://arxiv.org/abs/2001.08361.

Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. In *International Conference on Learning Representations*, 2015.

Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning, April 2021. URL https://arxiv.org/abs/2104.08691. arXiv:2104.08691.

Hengli Li, Chenxi Li, Tong Wu, Xuekai Zhu, Yuxuan Wang, Zhaoxin Yu, Eric Hanchen Jiang, Song-Chun Zhu, Zixia Jia, Ying Nian Wu, and Zilong Zheng. Seek in the dark: Reasoning via test-time instance-level policy gradient in latent space, May 2025. URL https://arxiv.org/abs/2505.13308. arXiv:2505.13308.

Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation, January 2021. URL https://arxiv.org/abs/2101.00190. arXiv:2101.00190.

Zhiyuan Li, Hong Liu, Denny Zhou, and Tengyu Ma. Chain of Thought Empowers Transformers to Solve Inherently Serial Problems. In *International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=3EWTEy9MTM.

Bill Yuchen Lin, Ronan Le Bras, Kyle Richardson, Ashish Sabharwal, Radha Poovendran, Peter Clark, and Yejin Choi. ZebraLogic: On the Scaling Limits of LLMs for Logical Reasoning. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 37889–37905. PMLR, 2025. URL https://proceedings.mlr.press/v267/lin25i.html.

Jiawei Liu, Chunqiu Steven Xia, Yuyao Wang, and Lingming Zhang. Is Your Code Generated by ChatGPT Really Correct? Rigorous Evaluation of Large Language Models for Code Generation. In *Advances in Neural Information Processing Systems*, volume 36, 2023. doi: 10.52202/075280-0943. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/43e9d647ccd3e4b7b5baab53f0368686-Abstract-Conference.html.

Sara Vera Marjanović, Arkil Patel, Vaibhav Adlakha, Milad Aghajohari, Parishad BehnamGhader, Mehar Bhatia, Aditi Khandelwal, Austin Kraft, Benno Krojer, Xing Han Lù, Nicholas Meade, Dongchan Shin, Amirhossein Kazemnejad, Gaurav Kamath, Marius Mosbach, Karolina Stańczak, and Siva Reddy. DeepSeek-R1 Thoughtology: Let's Think about LLM Reasoning, 2025. URL https://arxiv.org/abs/2504.07128.

Iman Mirzadeh, Keivan Alizadeh, Hooman Shahrokhi, Oncel Tuzel, Samy Bengio, and Mehrdad Farajtabar. GSM-symbolic: Understanding the limitations of mathematical reasoning in large language models. In *International Conference on Learning Representations*, 2025. URL https://arxiv.org/abs/2410.05229.

Jesse Mu, Xiang Lisa Li, and Noah Goodman. Learning to compress prompts with gist tokens, April 2023. URL https://arxiv.org/abs/2304.08467. arXiv:2304.08467.

Maxwell Nye, Anders Johan Andreassen, Guy Gur-Ari, Henryk Michalewski, Jacob Austin, David Bieber, David Dohan, Aitor Lewkowycz, Maarten Bosma, David Luan, Charles Sutton, and Augustus Odena. Show your work: Scratchpads for intermediate computation with language models, November 2021. URL https://arxiv.org/abs/2112.00114. arXiv:2112.00114.

OpenAI. Learning to Reason with LLMs. https://openai.com/index/learning-to-reason-with-llms/, 2024. Accessed 29 August 2026.

Ferdinand Österreicher and José A. P. Casquilho. On the Gini–Simpson Index and Its Generalisation—A Historic Note. *South African Statistical Journal*, 52(2):129–137, 2018.

Aleksandar Petrov, Philip H. S. Torr, and Adel Bibi. When Do Prompting and Prefix-Tuning Work? A Theory of Capabilities and Limitations. In *International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/18a367688479c1b08b001584218a4443-Abstract-Conference.html. arXiv:2310.19698.

Jacob Pfau, William Merrill, and Samuel R. Bowman. Let's think dot by dot: Hidden computation in transformer language models, April 2024. URL https://arxiv.org/abs/2404.15758. arXiv:2404.15758.

Archiki Prasad, Swarnadeep Saha, Xiang Zhou, and Mohit Bansal. ReCEval: Evaluating Reasoning Chains via Correctness and Informativeness. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 10066–10086, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.622. URL https://aclanthology.org/2023.emnlp-main.622/.

Qwen Team. Qwen3 Technical Report. *arXiv preprint arXiv:2505.09388*, 2025. doi: 10.48550/arXiv.2505.09388. URL https://arxiv.org/abs/2505.09388.

Keshav Ramji, Tahira Naseem, and Ramón Fernandez Astudillo. Thinking without words: Efficient latent reasoning with abstract chain-of-thought, April 2026. URL https://arxiv.org/abs/2604.22709. arXiv:2604.22709.

Michael Rizvi-Martel, Guillaume Rabusseau, and Marius Mosbach. The illusion of superposition? a principled analysis of latent thinking in language models, April 2026. URL https://arxiv.org/abs/2604.06374. arXiv:2604.06374.

Zhenyi Shen, Hanqi Yan, Linhai Zhang, Zhanghao Hu, Yali Du, and Yulan He. CODI: Compressing Chain-of-Thought into Continuous Space via Self-Distillation, February 2025. URL http://arxiv.org/abs/2502.21074. arXiv:2502.21074.

Charlie Snell, Dan Klein, and Ruiqi Zhong. Learning by distilling context, September 2022. URL https://arxiv.org/abs/2209.15189. arXiv:2209.15189.

Yang Sui, Yu-Neng Chuang, Guanchu Wang, Jiamu Zhang, Tianyi Zhang, Jiayi Yuan, Hongyi Liu, Andrew Wen, Shaochen Zhong, Na Zou, Hanjie Chen, and Xia Hu. Stop Overthinking: A Survey on Efficient Reasoning for Large Language Models. *Transactions on Machine Learning Research*, 2025. ISSN 2835-8856. URL https://openreview.net/forum?id=HvoG8SxggZ. arXiv:2503.16419.

Haojin Wang, Zining Zhu, and Freda Shi. Distribution prompting: Understanding the expressivity of language models through the next-token distributions they can produce, May 2025. URL https://arxiv.org/abs/2505.12244. arXiv:2505.12244.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models, January 2022a. URL https://arxiv.org/abs/2201.11903. arXiv:2201.11903.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, 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*, volume 35, pp. 24824–24837, 2022b.

David Wingate, Mohammad Shoeybi, and Taylor Sorensen. Prompt Compression and Contrastive Conditioning for Controllability and Toxicity Reduction in Language Models. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pp. 5621–5634, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.findings-emnlp.412. URL https://aclanthology.org/2022.findings-emnlp.412/.

Chünhung Wu, Jinliang Lu, Zixuan Ren, Gangqiang Hu, Zhi Wu, Dai Dai, and Hua Wu. LLMs Are Single-threaded Reasoners: Demystifying the Working Mechanism of Soft Thinking, August 2025. URL http://arxiv.org/abs/2508.03440. arXiv:2508.03440.

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, Suzhou, China, November 2025. Association for Computational Linguistics. doi: 10.18653/v1/2025.emnlp-main.165. URL https://aclanthology.org/2025.emnlp-main.165/.

Silei Xu, Wenhao Xie, Lingxiao Zhao, and Pengcheng He. Chain of Draft: Thinking Faster by Writing Less, 2025a. URL https://arxiv.org/abs/2502.18600. arXiv:2502.18600.

Yige Xu, Xu Guo, Zhiwei Zeng, and Chunyan Miao. SoftCoT: Soft Chain-of-Thought for Efficient Reasoning with LLMs, May 2025b. URL http://arxiv.org/abs/2502.12134. arXiv:2502.12134.

Yige Xu, Xu Guo, Zhiwei Zeng, and Chunyan Miao. SoftCoT++: Test-Time Scaling with Soft Chain-of-Thought Reasoning, May 2025c. URL http://arxiv.org/abs/2505.11484. arXiv:2505.11484.

Wengao Ye, Yan Liang, and Lianlei Shan. Thinking on the Fly: Test-Time Reasoning Enhancement via Latent Thought Policy Optimization, October 2025. URL http://arxiv.org/abs/2510.04182. arXiv:2510.04182.

Zhaoxin Yu, Qi Shen, Hengli Li, Zhaowei Zhang, Song-Chun Zhu, Chi Zhang, and Zilong Zheng. Gradcuit: Credit-assigned gradient flow enables robust and interpretable test-time latent reasoning, August 2026. URL https://arxiv.org/abs/2608.02585. arXiv:2608.02585.

Zhenrui Yue, Bowen Jin, Huimin Zeng, Honglei Zhuang, Zhen Qin, Jinsung Yoon, Lanyu Shang, Jiawei Han, and Dong Wang. Hybrid latent reasoning via reinforcement learning, May 2025. URL https://arxiv.org/abs/2505.18454. arXiv:2505.18454.

Zhen Zhang, Xuehai He, Weixiang Yan, Ao Shen, Chenyang Zhao, Shuohang Wang, Yelong Shen, and Xin Eric Wang. Soft Thinking: Unlocking the Reasoning Potential of LLMs in Continuous Concept Space, May 2025a. URL http://arxiv.org/abs/2505.15778. arXiv:2505.15778.

Zhen Zhang, Xuehai He, Weixiang Yan, Ao Shen, Chenyang Zhao, and Xin Wang. Soft Thinking: Unlocking the Reasoning Potential of LLMs in Continuous Concept Space. In *Advances in Neural Information Processing Systems*, volume 38, 2025b. doi: 10.52202/085713-5629. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/f7396d1c54d51416958d63e285377103-Abstract-Conference.html.

Zhi Zheng, Yu Gu, Wei Liu, Yee Whye Teh, and Wee Sun Lee. Soft-grpo: Surpassing discrete-token llm reinforcement learning via gumbel-reparameterized soft-thinking policy optimization, November 2025. URL https://arxiv.org/abs/2511.06411. arXiv:2511.06411.

Rui-Jie Zhu, Tianhao Peng, Tianhao Cheng, Xingwei Qu, Jinfa Huang, Dawei Zhu, Hao Wang, Kaiwen Xue, Xuanliang Zhang, Yong Shan, Tianle Cai, Taylor Kergan, Assel Kembay, Andrew Smith, Chenghua Lin, Binh Nguyen, Yuqi Pan, Yuhong Chou, Zefan Cai, Zhenhe Wu, Yongchi Zhao, Tianyu Liu, Jian Yang, Wangchunshu Zhou, Chujie Zheng, Chongxuan Li, Yuyin Zhou, Zhoujun Li, Zhaoxiang Zhang, Jiaheng Liu, Ge Zhang, Wenhao Huang, and Jason Eshraghian. A Survey on Latent Reasoning, 2025. URL https://arxiv.org/abs/2507.06203.

Yufan Zhuang, Liyuan Liu, Chandan Singh, Jingbo Shang, and Jianfeng Gao. Text Generation Beyond Discrete Token Sampling, May 2025a. URL http://arxiv.org/abs/2505.14827. arXiv:2505.14827.

Yufan Zhuang, Liyuan Liu, Chandan Singh, Jingbo Shang, and Jianfeng Gao. Mixture of Inputs: Text Generation Beyond Discrete Token Sampling. In *Advances in Neural Information Processing Systems*, volume 38, 2025b. doi: 10.52202/085713-0889. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/25c61012c845908bff2ea6223cdc6192-Abstract-Conference.html.

Table A.1: Learning rate at $k = 1$ as a function of the dataset and the model size. Roughly, the larger model wants a learning rate 2–3$\times$ that of the smaller and MATH-Hard wants 3–4$\times$ HumanEval+.

$b = \eta(1)$	1.7B	8B	Ratio
HumanEval+	0.70	0.25	2.8
MATH-Hard	2.0	1.0	2.0
Ratio	2.9	4.0	—

A Optimisation Procedure

The paper focuses on the *existence* of high quality latent thoughts. The specifics of the procedure used to find them are secondary. Nevertheless, we detail them precisely here.

The distillation scheme, described in §3, could be performed *online*: at every step, generate (anew) a batch of rollouts from the teacher; calculate the teacher-forced loss against these rollouts. However, this would be prohibitively expensive: teacher-rollouts involve 1000s of forward passes, versus only one for the loss calculation. As such, we generate a large training bank, *offline*. In §4/§5, we sample 2048/1024 teacher rollouts and perform 2000/1000 optimisation steps with batch size 32, leading to each rollout being used roughly 30 times. The difference is due to empirically observed faster convergence for GSM-Symbolic (§5) than HumanEval+ or MATH-Hard (§4). Moreover, we can use the same rollouts across all variants of our experiemnts, amortising the cost substantially.

It is not merely the specific tokens in these rollouts that matter, but also the conditional distributions; these are required for a KL, rather than negative log-likelihood loss. Qwen3's vocabulary contains over 150k tokens (Qwen Team, 2025), the vast majority of which will have vanishingly small probability assigned to them. As is standard in the literature, we truncate: we take a generous truncation, keeping the top 256, and aggregate the remainder into a tail probability.

For the main runs on HumanEval+ and MATH-Hard, we conduct an exploratory learning-rate sweep at 500 steps on 16 questions, from which we selected $\eta(k) = b/k$ with b given in Table A.1; see Figures A.2 (sweep) and A.4 (fit). An 8-question learning-rate sweep for GSM-Symbolic, including varying the number of augmentations per question whilst keeping the total training rollouts at 1024 and evaluating on held-out augmentations, yielded $\eta(k) = 1/k$; see Figures A.3 (sweep) and A.5 (fit). Importantly, these 8 questions are distinct from the 24 used for the full run shown in Figure 5.1.

The GSM-Symbolic sweep is only exploratory: the majority of its cases are censored (ie, the optimum is at the end of the grid) and is run on a smaller set of learning rates.

Batch size 32 and weight decay 10^{-2} and are used. No signal across batch sizes $\{32, 128\}$ or weight decays $\{10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 1\}$ was observed; the values were picked somewhat arbitrarily.

Optimisation uses Adam (Kingma & Ba, 2015), whose approximate invariance to gradient scale reduces sensitivity to the substantial variation in absolute magnitude of the loss. The loss is over *complete answers*, which vary subsantially in length, not per token. The learning rate uses a linear warm-up/cool-down schedule over the first/last 10%/35% of steps.

B Derivations

§B.1. The gradient-equivalence of the objective (3.2) and loss (3.3) is derived.

§B.2. The similarity inequality $\mathcal{G}(\mathrm{T}, \mathrm{S}_z) \leq \sqrt{\mathcal{G}(\mathrm{T}, \mathrm{T})\mathcal{G}(\mathrm{S}_z, \mathrm{S}_z)}$ is derived.

B.1 Loss Derivation

The objectives and loss used for distillation are described in Section 3, particularly (3.2, 3.3). The justification that loss $L_q(z)$ is appropriate for the objective $\mathcal{D}_q(z)$ is described here.

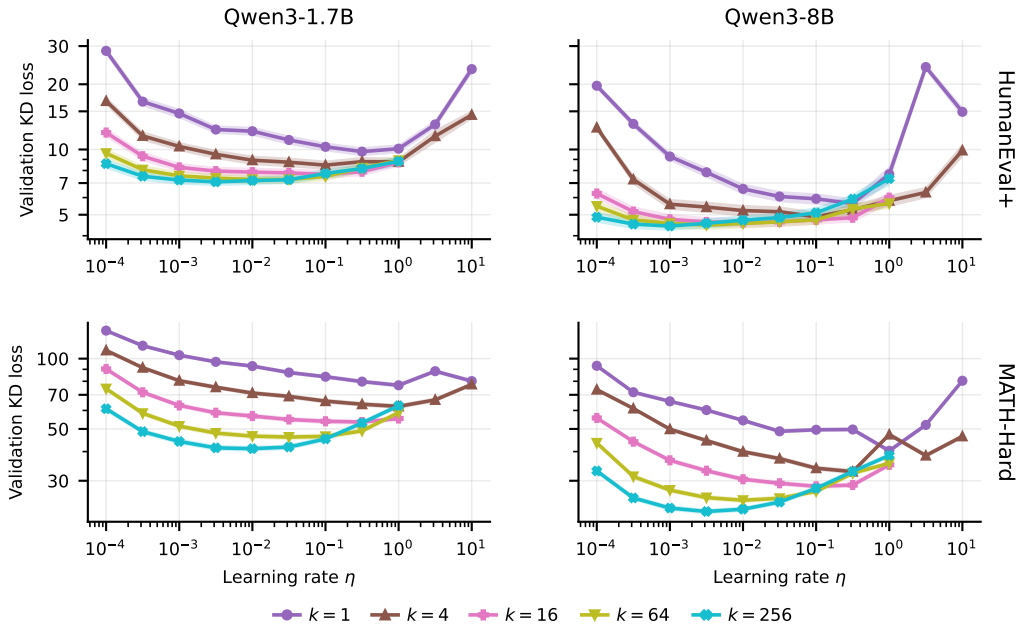


Figure A.2: Learning-rate sweep for {HumanEval+, MATH-Hard} × Qwen3-{1.7B, 8B} on a fixed set of 16 questions, run for 500 optimisation steps. Shared regions indicate ±2 standard errors.

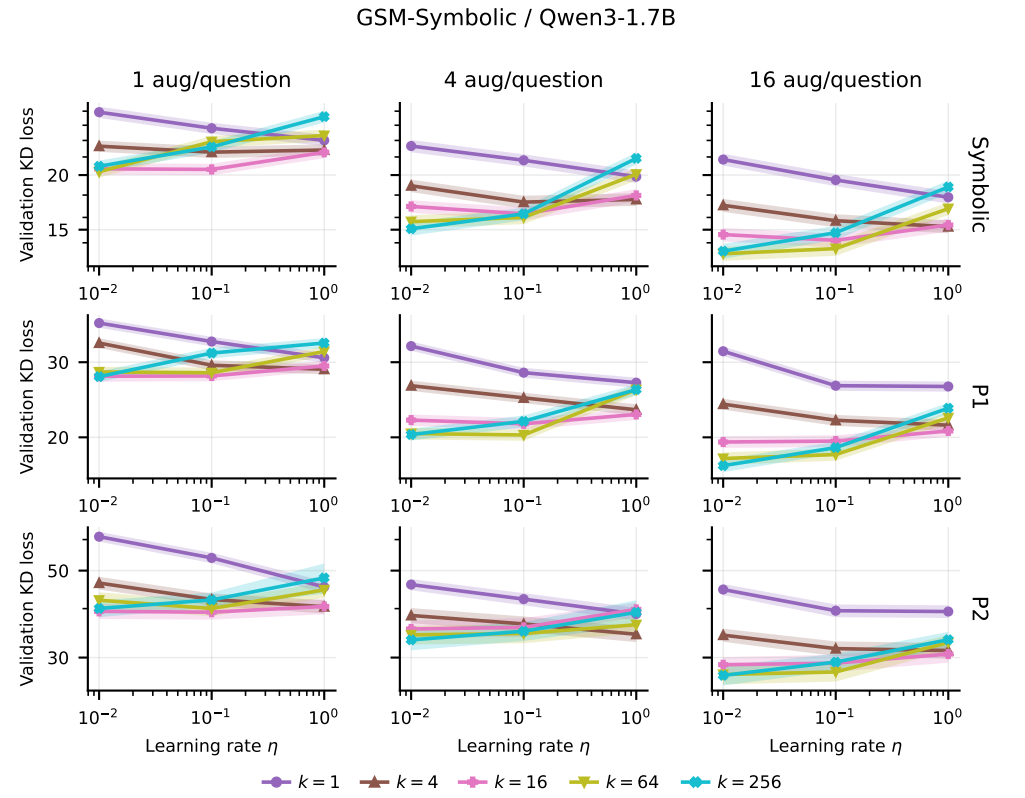


Figure A.3: Learning-rate sweep for GSM-Symbolic with Qwen3-1.7B on a fixed set of 8 questions, run for 500 optimisation steps. Shaded regions indicate ±2 standard errors.

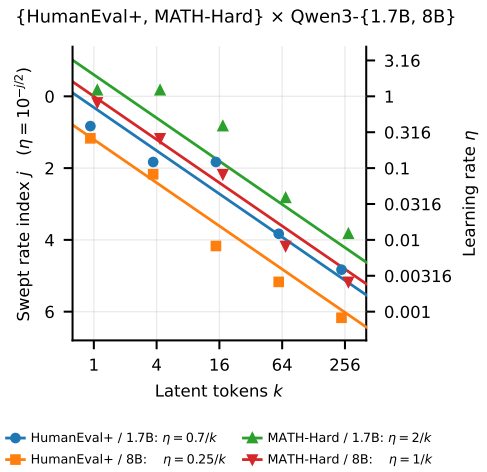


Figure A.4: Learning-rate fit: {HumanEval+, MATH-Hard} × Qwen3-{1.7B, 8B}.

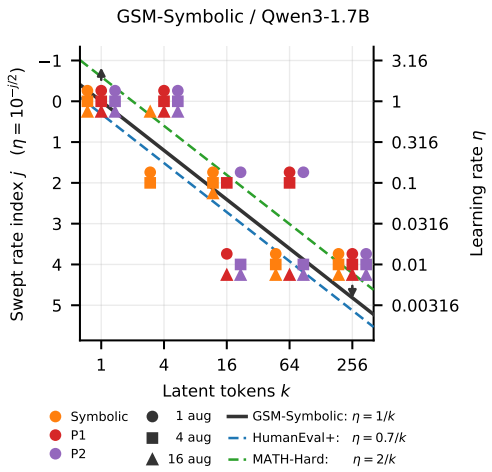


Figure A.5: Learning-rate fit: GSM-Symbolic with Qwen3-1.7B; arrows indicate censoring.

We can decompose the objective $\mathcal{D}_q(z)$ as

$$\mathcal{D}_q(z) = \mathbb{E}_{Y\sim \mathrm{T}_q}\big[\log\big(\mathrm{T}_q(Y) \,/\, \mathrm{S}_{q,z}(Y)\big)\big] \tag{B.1}$$

$$= \mathbb{E}_{(\tau,Y)\sim p(\cdot|q)}\left[\log \frac{\sum_\tau p(Y \mid q\oplus\tau)p(\tau \mid q)}{p(Y \mid q\oplus\tau)} \frac{p(Y \mid q\oplus\tau)}{p(Y \mid q\oplus z)}\right] \tag{B.2}$$

$$= -I(\tau; Y \mid q) + \mathcal{L}_q(z), \tag{B.3}$$

where $I(\tau; Y \mid q)$ is the mutual information between the teacher's answer and reasoning trace and

$$\mathcal{L}_q(z) \coloneqq \mathbb{E}_{(\tau,\cdot)\sim p(\cdot|q)}\big[\mathrm{KL}\big(p(\cdot \mid q\oplus\tau) \,\big\|\, p(\cdot \mid q\oplus z)\big)\big] \tag{B.4}$$

$$= \mathbb{E}_{(\tau,Y)\sim p(Y|q)}\big[\log\big(p(Y \mid q\oplus\tau) \,/\, p(Y \mid q\oplus z)\big)\big]. \tag{B.5}$$

Since the values $\mathcal{D}_q(z)$ and $\mathcal{L}_q(z)$ differ by an additive constant, gradients are unaffected:

$$\nabla_z \mathcal{D}_q(z) = \nabla_z \mathcal{L}_q(z) \quad \forall z \in \mathbb{R}^{k\times d}. \tag{B.6}$$

It remains to estimate the population-level loss $\mathcal{L}_q(z)$. Expanding answers into sums over tokens,

$$\mathcal{L}_q(z) = \mathbb{E}_{(\tau,Y)\sim p(\cdot|q)}\big[\textstyle\sum_{j=1}^{|Y|} \log\big(p^{\text{next}}(Y \mid q\oplus\tau\oplus Y_{<j}) \,/\, p^{\text{next}}(Y \mid q\oplus z\oplus y_{<j})\big)\big] \tag{B.7}$$

$$= \mathbb{E}_{(\tau,Y)\sim p(\cdot|q)}\big[\textstyle\sum_{j=1}^{|Y|} \mathrm{KL}\big(p^{\text{next}}(\cdot \mid q\oplus\tau\oplus y_{<j}) \,\big\|\, p^{\text{next}}(\cdot \mid q\oplus z\oplus y_{<j})\big)\big], \tag{B.8}$$

where the "next" superscript indicates that the distribution is over the next token only. The loss $L_q(z; \mathcal{R})$ is exactly the Monte Carlo estimator: it replaces the expectation over (τ, Y) with samples.

B.2 Cauchy–Schwarz Similarity Inequality

§4.2 claimed that Cauchy–Schwarz established the inequality

$$\mathcal{G}(\mathrm{T}, \mathrm{S}_z) \le \sqrt{\mathcal{G}(\mathrm{T}, \mathrm{T})\mathcal{G}(\mathrm{S}_z, \mathrm{S}_z)}, \tag{B.9}$$

and deferred the verification to here. Since $\mathcal{G}$ averages twice, Cauchy–Schwarz is applied twice.

For a question q, let p_q and p'_q be two answer distributions for it. First, Cauchy–Schwarz gives

$$\mathcal{G}_q(p_q, p'_q) = \textstyle\sum_y p_q(y)p'_q(y) \le \sqrt{\sum_y p_q(y)^2}\sqrt{\sum_y p'_q(y)^2} = \sqrt{\mathcal{G}_q(p_q, p_q)\,\mathcal{G}_q(p'_q, p'_q)}. \tag{B.10}$$

Second, average over $Q \sim \mathrm{Unif}(\mathcal{Q})$: Cauchy–Schwarz again gives

$$\mathbb{E}\Big[\sqrt{\mathcal{G}_Q(p_Q, p_Q)\,\mathcal{G}_Q(p'_Q, p'_Q)}\Big] \le \sqrt{\mathbb{E}[\mathcal{G}_Q(p_Q, p_Q)]\mathbb{E}[\mathcal{G}_Q(p'_Q, p'_Q)]}. \tag{B.11}$$

This gives (B.9) upon taking $p_q \coloneqq \mathrm{T}_q$ and $p'_q \coloneqq \mathrm{S}_{q,z}$. The same holds for the backbone B_q too.

C Dataset Details

Each benchmark dataset $\mathcal{Q}$ comprises a fixed number of questions. For each benchmark $\mathcal{Q}$ and question $q \in \mathcal{Q}$, we sample 2048 rollouts to train on and 128 for validation using CoT.

A (very low) proportion of these fail to generate any answer, almost exclusively because they hit the LLM's sequence-length cap before emitting the `</think>` tag; we refer to these as *null rollouts*. This causes the number of *usable* rollouts to vary slightly across settings. We do not (iteratively) resample these. Valid rollout rates for HumanEval+/MATH-Hard are given in Figure C.1.

C.1 HumanEval+

HumanEval+ is a program-synthesis benchmark, comprising of 164 Python programming problems. It extends HumanEval (Chen et al., 2021) with substantially expanded test suites and correct reference implementations, enabling more rigorous evaluation of generated solutions (Liu et al., 2023).

Our experiments use all 164 questions. To verify correctness, we take the complete generated text after the final closing `</think>` tag, append it verbatim to the benchmark's function prompt. The resulting function is then executed against both the original HumanEval tests and the additional EvalPlus tests. A rollout is correct if and only if it passes all tests.

To reduce the overall proportion of null rollouts, we discourage overthinking via the prompt.

> Complete the following Python function. If necessary, reason first, in as few sentences as the problem needs. Keep it to the logic of this specific completion; do not enumerate edge cases or compare alternative implementations. Your entire reply must be the function body and nothing else: no prose, no explanation, no markdown fences, no restated signature. If you have any reasoning to do, keep it inside the think tags only — never in the reply.
>
> [question inserted]

We started with the 1.7B model. When extending to the 8B model, we found that it did not indent the code—which, to be fair, it had not been asked to do, but the 1.7B did of its own accord—leading to worse performance than the 1.7B model. This was rectified by specifying indentation in the prompt.

> Complete the following Python function. If necessary, reason first, in as few sentences as the problem needs. Keep it to the logic of this specific completion; do not enumerate edge cases or compare alternative implementations. Your entire reply must be the *correctly indented* function body *using at least 4 spaces to indent lines of code, ready to append directly to the provided function,* and nothing else: no prose, no explanation, no markdown fences, no restated signature. If you have any reasoning to do, keep it inside the think tags only — never in the reply.
>
> [question inserted]

The addition is highlighted in *italics*. We acknowledge that an identical prompt would make for a fairer comparison. However, intuitively, it is minor, and unlikely to affect the overall behaviour.

The median/mean percentage of null rollouts per question is 0.0%/3.6% and 0.0%/2.5% for the 1.7B and 8B model sizes, respectively. The distributions of the proportion of valid—ie, non-null—rollouts across HumanEval+ questions are shown in Figure C.1.

C.2 MATH-Hard

MATH-Hard is a subset of the MATH benchmark (Hendrycks et al., 2021), consisting of questions assigned its highest difficulty rating. It comprises competition-level questions spanning seven mathematical categories: algebra; geometry; number theory; counting and probability; intermediate algebra; prealgebra; precalculus. Each question is accompanied by a worked solution and final answer expressed wrapped in `\boxed{·}`.

For our experiments, we uniformly sample 128 questions from MATH-Hard across all seven categories; see Table C.2 for the original/sampled distributions across categories. To verify correctness,

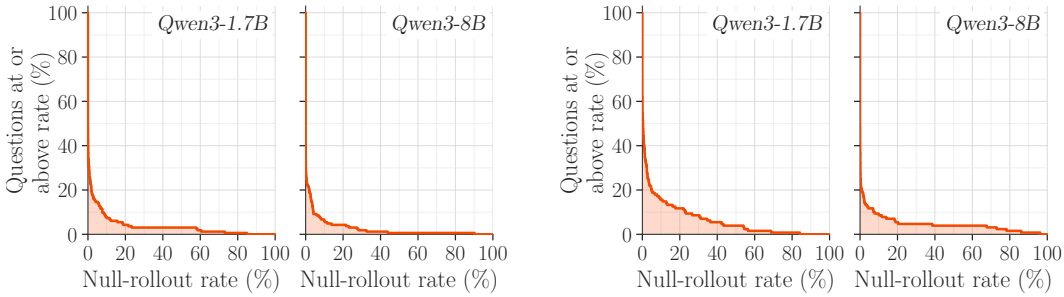


Figure C.1: Distribution of null rollouts across HumanEval+/MATH-Hard questions (*left*/*right*): a point (x, y) on the curve means that the proportion of questions whose null rollouts comprise at least $x\%$ of their rollouts is $y\%$. This automatically provides a non-increasing curve.

Table C.2: Distribution of question categories used across the MATH-Hard experiments.

Dataset Split	Algebra	Counting & Probability	Geometry	Intermediate Algebra	Number Theory	Pre-Algebra	Pre-Calculus
"Train"	436	276	421	429	313	252	177
Ours	24	14	23	27	19	14	7

first extract the contents from the `\boxed{·}` wrapper, then pass it through the `math_verify` package, as is the recommended procedure.

To address lack of diversity in answers—eg, if the question asks for an answer which is an integer, the answers will likely be single-token—we prompt the model to generate a full solution to the question, not merely output the final value.

> Write a concise model answer with working to the following maths problem. If necessary, reason first, in as few sentences as the problem needs. Keep any exploratory reasoning, checking, backtracking, or alternative approaches inside the think tags only. After the think tags end, begin the model answer immediately. Include only clean, solution-grade working, and finish with the final answer inside `\boxed`.
>
> [question inserted]

The entire evaluation of the answer rests on the `\boxed{·}` content. So, hypothetically, the model could have incorrect working, yet give the correct answer by chance. Ideally, we would have an exam-type mark scheme, but we leave this for future work. In the end, the verifier doesn't affect the distillation scheme, and it is consistent across the teacher and student.

The median/mean percentage of null rollouts per question is $0.3\%/6.4\%$ and $0.0\%/4.6\%$ for the 1.7B and 8B model sizes, respectively. The distributions of the proportion of valid rollouts across MATH-Hard questions are shown in Figure C.1.

C.3 GSM-Symbolic

GSM-Symbolic (Mirzadeh et al., 2025) is derived from the GSM8K benchmark (Cobbe et al., 2021), and consists of templated augmentations of GSM8K's grade-school arithmetic word problems. We use the three pre-generated datasets released with the paper. The first, "Symbolic", merely resamples the names, entities and numerical values. The others increase the difficulty by adding one or two clauses, and are called "P1" and "P2", respectively. To make this concrete, we give an example of the different levels of augmentation, using GSM8K #799; *italics* highlights (a subset of) changes.

Base. A loaf of bread at the bakery costs \$2. Bagels cost \$1 each. How much more do 3 loaves of bread cost than 2 bagels?

Sym. A loaf of bread at the store costs *€3*. *Croissants* cost *€1* each. How much more do *5* loaves of bread cost than *4 croissants*?

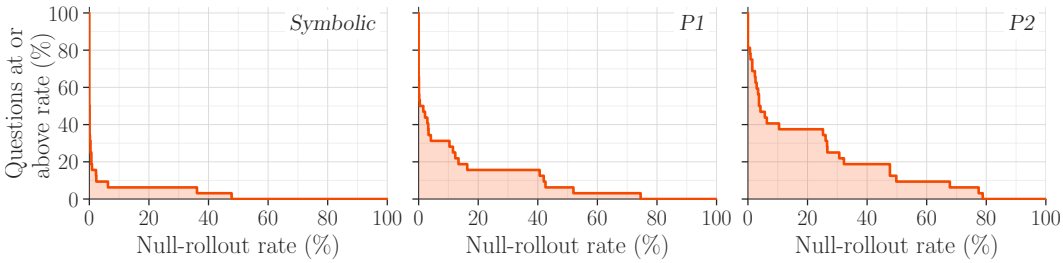


Figure C.3: Distribution of null rollouts across the 32 randomly sampled GSM-Symbolic questions: a point (x, y) on the curve means that the proportion of questions whose null rollouts comprise at least $x\%$ of their rollouts is $y\%$. This automatically provides a non-increasing curve.

P1. A loaf of bread at the store costs €13. *The cost of a croissant is one-fourth of a loaf of bread.* How much more do 6 loaves of bread cost than 8 croissants?

P2. A loaf of sourdough at the cafe costs ¥15. The cost of a muffin is one-fourth of a loaf of sourdough. *The cafe has a promotion that if one buys 3 muffins or more, they get a 20% discount on muffins.* How much more do 10 loaves of sourdough cost than 8 muffins?

Counting solution steps confirms that the additional clauses increase the number of reasoning steps required: Symbolic preserves the depth of the base problems (mean 4.58 versus 4.59 steps); P1 adds approximately one step (5.55) and P2 adds approximately 1.7 steps (6.22).

There are 50 questions which have all three types of augmentation, and each has (at least) 50 of each type. We take these as our dataset. Again, we use `math_verify` to assess correctness.

We use the same prompt as for MATH-Hard, requesting a clean, show-your-workings solution.

> Write a concise model answer with working to the following maths problem. If necessary, reason first, in as few sentences as the problem needs. Keep any exploratory reasoning, checking, backtracking, or alternative approaches inside the think tags only. After the think tags end, begin the model answer immediately. Include only clean, solution-grade working, and finish with the final answer inside `\boxed`.
>
> [question inserted]

The median/mean percentage of null rollouts per question is 0.0%/3.1%, 0.9%/10% and 4.8%/18% for the Symbolic, P1 and P2 variants, respectively. This indicates a very heavy tail for P1 and P2; further, the questions contributing to the tail are ones where the teacher struggles, even conditional on outputting an answer. The distributions of the proportion of valid rollouts across the GSM-Symbolic questions are shown in Figure C.3.

D Further Results

The main body selected certain dataset–model pairs to display; the deferred figures are given here. The overall interpretation is the same, so narration is kept to a minimum. The one exception is accuracy when null rollouts are removed from the teacher.

D.1 Loss

Figure 4.1 displayed both HumanEval+ and MATH-Hard, but only for Qwen3-8B. The analogous data for Qwen3-1.7B is displayed in Figure D.1. The behaviour is also analogous.

D.2 Accuracy

Figure 4.3 covered accuracy for the full set {HumanEval+, MATH-Hard} × Qwen3-{1.7B, 8B}. We deferred one table showing the teacher-to-student head-to-head.

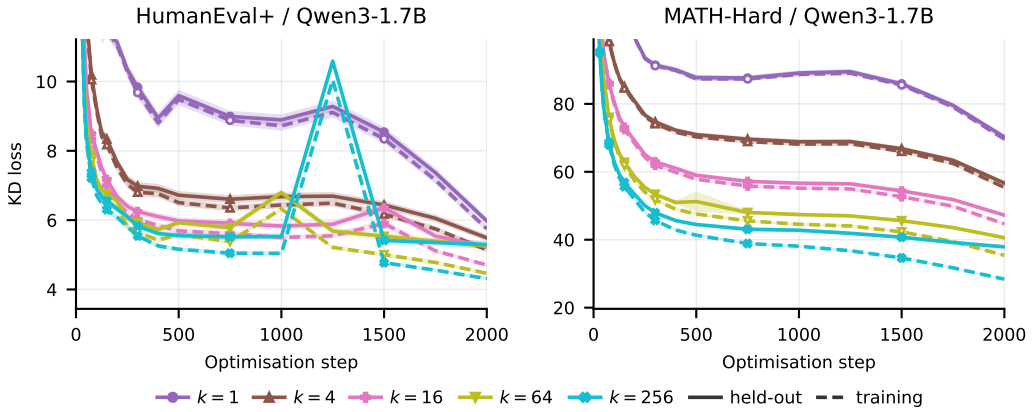


Figure D.1: Training/validation loss (*dashed*/*solid*) against optimisation step for Qwen3-1.7B on HumanEval+/MATH-Hard (*left*/*right*). Different colours/markers indicate different numbers k of latent thoughts. Shaded regions indicate ± 2 standard errors, treating the questions as fixed. The analogous results for Qwen3-8B are displayed in Figure 4.1 in §4.1.

Table D.2: Teacher versus student ($k = 256$) head-to-head comparisons across {HumanEval+, MATH-Hard} × Qwen3-{1.7B, 8B}: left-hand table shows the percentage-point difference "student minus teacher"; right-hand table shows how many questions where the student beats the teacher.

	HumanEval+	**MATH-Hard**
1.7B	-0.97 ± 0.17	$+1.35 \pm 0.18$
8B	-0.98 ± 0.14	$+2.84 \pm 0.11$

	HumanEval+	**MATH-Hard**
1.7B	$71/164$ (43%)	$50/128$ (39%)
8B	$79/164$ (48%)	$45/128$ (35%)

We also consider a variant here which accounts for null rollouts differently. Namely, here, we discard all the teacher's null rollouts—ie, the ones where it failed to answer before hitting its cap. The motivation for this is that these are the only rollouts that the student *actually saw*. So, technically, our loss did not target the true teacher distribution, but the teacher distribution *conditional on being non-null*. Taken to the extreme, suppose that the teacher runs out of context 90% of the time, but the 10% of times that it does not is gets the answer right. Then, 2048 rollouts lead to roughly 200 non-null rollouts for the student to train on—but, they're all correct. So, it seems plausible that the student could get above 10% accuracy.

Now, it turns out that this is not the case for us: the questions which the teacher hits the cap on are difficult questions for which its non-null rollouts are typically incorrect; see Table D.3. To emphasise, weight *questions*, not *rollouts*, equally: when the (non-null) rollout-count per questions is heterogeneous, we average first over all rollouts for each question then over the questions.

Changes in accuracy are tabulated in Table D.3. The analogous plot to Figure 4.3, with this new baseline, is given as Figure D.4. The key qualitative difference is that MATH-Hard/1.7B no longer exceeds the teacher under this metric, but MATH-Hard/8B still does—now by 1%, down from 11%.

Table D.3: Accuracy comparison of teacher with null rollouts filtered against teach without filtering and student at $k = 256$. All values are percentages. All absolute figures are accurate to $\pm 0.1\%$.

Dataset/Model	**Teacher** (all)	**Difference**	**Teacher** (non-null)	**Difference**	**Student** ($k = 256$)
HumanEval+/1.7B	71.6	$+0.67 \pm 0.03$	72.3	-1.6 ± 0.2	70.6
HumanEval+/8B	82.7	$+0.80 \pm 0.03$	83.5	-1.8 ± 0.1	81.7
MATH-Hard/1.7B	76.9	$+2.58 \pm 0.05$	79.5	-1.2 ± 0.2	78.2
MATH-Hard/8B	89.1	$+2.50 \pm 0.07$	91.6	$+0.3 \pm 0.1$	91.9

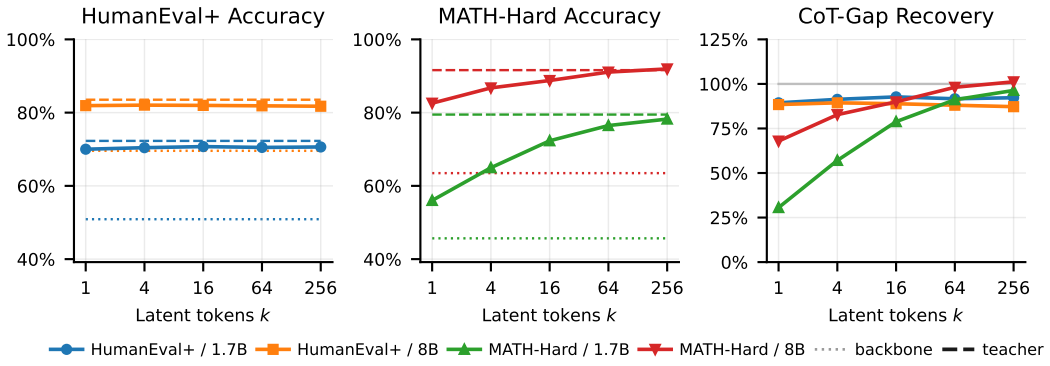


Figure D.4: Accuracy against number k of latent thoughts for the full set {HumanEval+, MATH-Hard} × Qwen3-{1.7B, 8B} with null rollouts removed from the teacher's computation and questions weighted equally. Horizontal lines indicate baselines: the (no-CoT) backbone and the (CoT) teacher. Shaded regions indicate ± 2 standard errors, treating the questions as fixed; each checkpoint has at least 1024 rollouts per question, rendering the error bars too small to see.

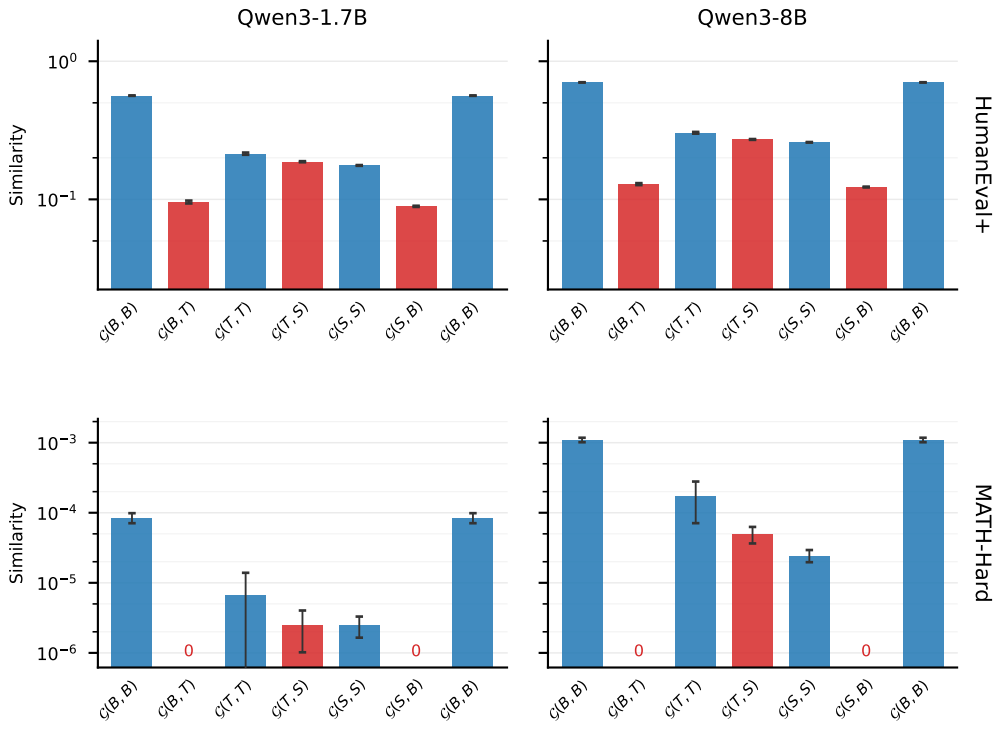


Figure D.5: Similarities for {HumanEval+, MATH-Hard} × Qwen3-{1.7B, 8B}. Horizontal bars indicate the upper bound $\mathcal{G}(X,Y) \leq \sqrt{\mathcal{G}(X,X)\mathcal{G}(Y,Y)}$ for $X, Y \in \{\mathrm{T}, \mathrm{S}_z, \mathrm{B}\}$, achieved only when $X = Y$. Whiskers indicate ± 2 standard errors, treating the questions as fixed.

D.3 Similarity

Figure 4.4 displayed the similarity metrics only for (HumanEval+, Qwen3-8B). The full set of {HumanEval+, MATH-Hard} × Qwen3-{1.7B, 8B} is displayed in Figure D.5.

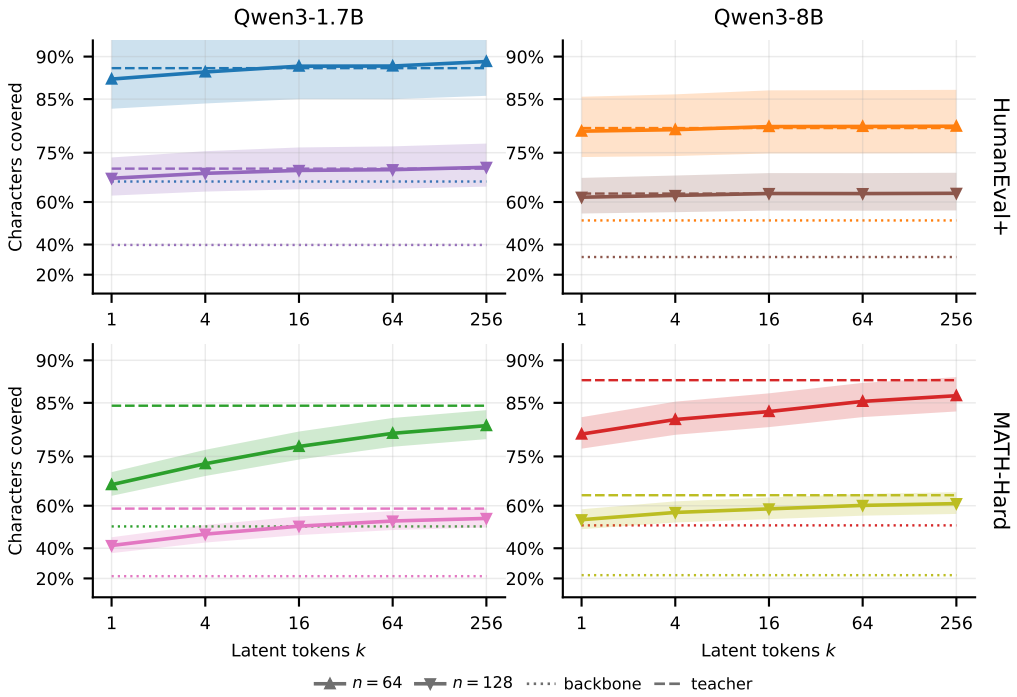


Figure D.6: 64-/128-gram coverage against number k of latent thoughts for {HumanEval+, MATH-Hard} × Qwen3-{1.7B, 8B}. Shaded regions indicate ± 2 standard errors as usual.

The behaviour of Qwen3-1.7B on HumanEval+ is almost identical to that of Qwen3-8B. The MATH-Hard behaviour is more extreme: the teacher–student relationship is similar, albeit a less tight fit; the backbone is still more diverse, yet has *no* intersection with the teacher or student. The self-similarity is far lower for MATH-Hard than HumanEval+, leading to substantially more noisy estimates.

D.4 Plagiarism

Figure 4.5 displayed the similarity metrics only for (MATH-Hard, Qwen3-8B). The full set of {HumanEval+, MATH-Hard} × Qwen3-{1.7B, 8B} is displayed in Figure D.6.

The behaviour of Qwen3-1.7B on MATH-Hard is analogous to that of Qwen3-8B. The HumanEval+ behaviour is substantially different: the curves are much flatter in k. This reflects the previous scenarios in which the (final-checkpoint) loss and accuracies were also somewhat indifferent to k; see Figures 4.1/D.1 and 4.3. In fact, the student may even exceed the teacher for HumanEval+, but the error bars are large so we do not draw any strong conclusion here.

D.5 GSM-Symbolic

The accuracy against number k of latent tokens for Qwen3-1.7B on GSM-Symbolic for different counts a of augmentations per question. The 'transposed' plot, in which accuracy is plotted against a and individual curves represent different values of k, is given here, as Figure D.7.

This clearly highlights the positive trend from increasing the augmentation-per-question count a. The substantial overlap of curves for Symbolic and, at low a, indicates the indifference to k.

Whilst accuracy is the most important downstream metric from a user's perspective, the (final-checkpoint) validation loss is a better indicator of fit to the teacher distribution. This is plotted in Figure D.8; both the against-k and against-a variants are displayed. The trends are broadly similar to those for accuracy—although, there is a definite plateau then up-tick from $k = 16$ to $k = 256$. This may be due to slight overfitting at this capacity, or it may simply be converging more slowly.

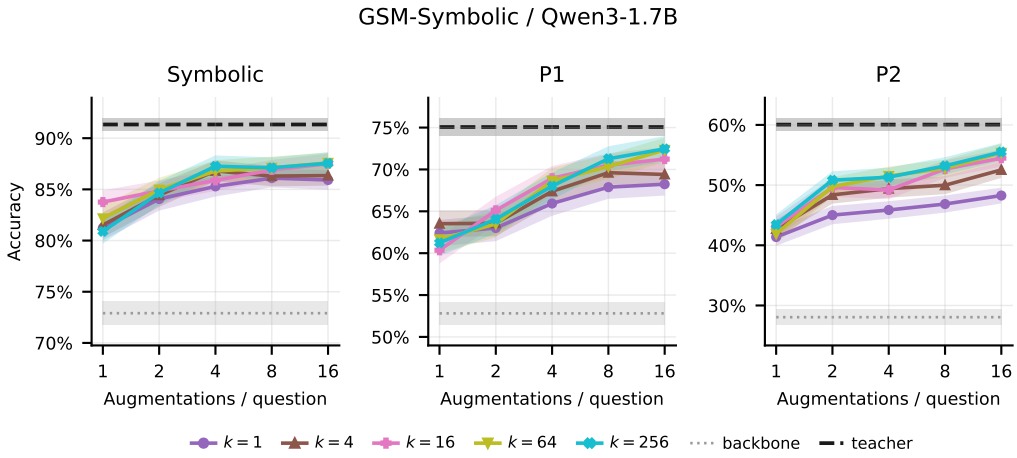


Figure D.7: Accuracy against number of augmentations per question for GSM-Symbolic on Qwen3-1.7B. Horizontal lines indicate baselines: the (no-CoT) backbone and the (CoT) teacher. Shaded regions indicate ± 2 standard errors, treating the *questions* as fixed but *augmentations* as sampled.

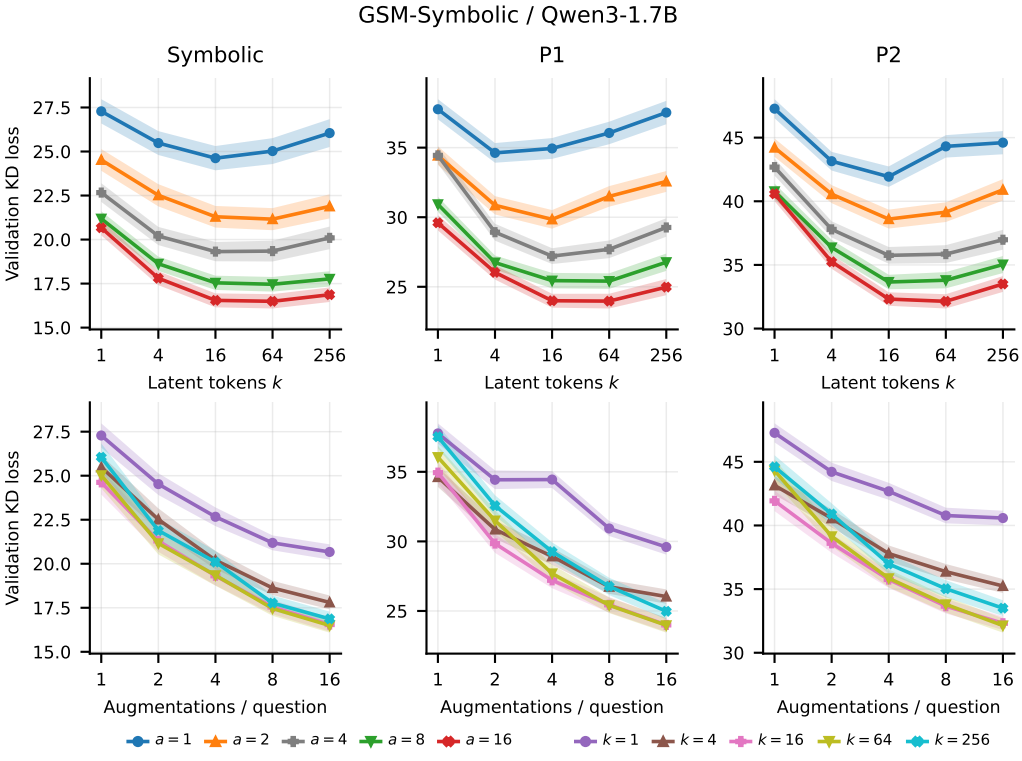


Figure D.8: Validation loss against k/a (*top*/bottom) for GSM-Symbolic on Qwen3-1.7B. Shaded regions indicate ± 2 standard errors, treating the *questions* as fixed but *augmentations* as sampled.