

MIXREASONING: SWITCHING MODES TO THINK

Anonymous authors

Paper under double-blind review

ABSTRACT

Reasoning models enhance performance by tackling problems in a step-by-step manner, decomposing them into sub-problems and exploring long chains of thought before producing an answer. However, applying extended reasoning to every step introduces substantial redundancy, as sub-problems vary widely in difficulty and complexity: a small number of pivotal steps are genuinely challenging and decisive for the final answer, while many others only involve straightforward revisions or simple computations. Therefore, a natural idea is to endow reasoning models with the ability to adaptively respond to this variation, rather than treating all steps with the same level of elaboration. To this end, we propose MixReasoning, a framework that dynamically adjusts the depth of reasoning within a single response. The resulting chain of thought then becomes a mixture of detailed reasoning on difficult steps and concise inference on simpler ones. Experiments on GSM8K, MATH-500, and AIME show that MixReasoning shortens reasoning length and substantially improves efficiency without compromising accuracy.

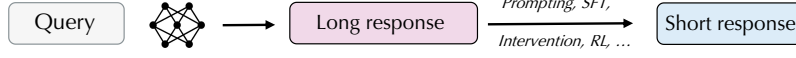
1 INTRODUCTION

Large Reasoning Models (LRMs) such as DeepSeek-R1 (Guo et al., 2025) and QwQ3 (Yang et al., 2025a) have achieved state-of-the-art results on a wide range of complex tasks, spanning arithmetic, commonsense, and scientific reasoning. A key driver of these gains is the use of long chains of thought (CoTs) (Wei et al., 2022) that externalize intermediate computations before arriving at a final answer. However, uniformly applying elaborate reasoning throughout the entire solution path induces substantial inference cost since thinking sequences become verbose, and autoregressive decoding time scales roughly linearly with sequence length. The resulting latency and compute overhead are prohibitive for interactive applications and degrade user experience (Fu et al., 2024); in addition, excessively verbose traces hurt readability by inserting coherence filler and redundant self-checks that humans typically avoid, significantly degrading user experience (Fu et al., 2025).

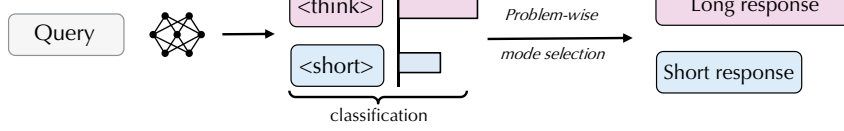
To mitigate these costs, recent work largely follows two lines. The first compresses the entire reasoning process via prompting (Ma et al., 2025a), fine-tuning (Ma et al., 2025b; Chen et al., 2025), token-budget constraints (Sun et al., 2025), or penalizing thinking tokens (Wang et al., 2025a), thereby making models “think less.” While effective in lowering latency, such global compression can inadvertently truncate pivotal reasoning steps, making it challenging to preserve accuracy and to maintain a favorable accuracy–efficiency balance. The second line adopts hybrid reasoning (Fang et al., 2025; Yang et al., 2025a), routing between reasoning and non-reasoning modes based on problem-level difficulty or user’s tolerance. This improves the trade-off in some regimes, yet it assumes that a problem admits a clean binary partition (reason vs. no-reason) and leaves long-reasoning trajectories rife with redundancy: many tokens still articulate routine manipulations (mostly complete ongoing linguistic or deterministic structures) (Wang et al., 2025b) that do not require detailed thought.

Since global length compression and problem-level long/short switching can diminish accuracy, we pursue a finer-grained alternative that preserves multi-step reasoning while controlling cost. Our method builds on three properties of LRMs: **(1) reasoning complexity of different substeps is heterogeneous within a CoT.** A small set of pivotal steps, such as initial analysis, decomposition, and key derivations, largely determines downstream progress and final answers, whereas many other steps (e.g., arithmetic carry-outs, case enumeration, or straightforward transformations) are comparatively easy. Consequently, the goal is not whether to think, but how to allocate detail-expanding at pivotal parts and remaining concise elsewhere within a CoT. **(2) Integrate thinking and non-**

(a) Long-to-short Compression



(b) Hybrid Reasoning



(c) MixReasoning

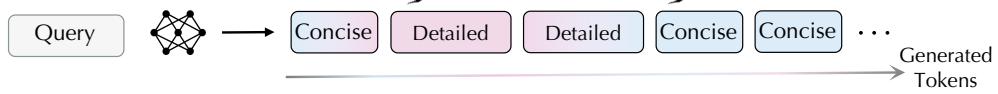


Figure 1: The comparison among Long-to-short compression, Hybrid reasoning, and MixReasoning. MixReasoning adaptively switches between thinking modes during inference based on local uncertainty, achieving a balance of conciseness and detail.

thinking modes without degrading the base model. Current methods (Fang et al., 2025; Yang et al., 2025a) to achieve this in one model typically involve retraining (e.g., via SFT/RL), which can inevitably introduce forgetting. Instead, we attach lightweight LoRA adapters trained to elicit non-thinking behavior while freezing base weights. By on-the-fly scaling strength of the LoRA adapter on a single served base model, we can easily switching thinking modes during reasoning process, thereby integrating thinking and non-thinking ability without sacrificing the capability of base model. (3) **Reasoning tokens are disproportionately consequential for trajectory formation:** tokens with the lowest next-token entropy mostly complete ongoing linguistic or deterministic structures, whereas the highest-entropy tokens actually facilitate reasoning and steer the model to explore plausible reasoning pathways (Wang et al., 2025b). In MixReasoning, we therefore treat local peaks in token-level entropy as decision points to which detail should be allocated: when entropy spikes, we temporarily diminish the strength of LoRA adapter to expand a short window into long-form reasoning, and then anneal back to the concise mode once uncertainty subsides. This uncertainty-aware allocation of long-reasoning prioritizes critical decision points, improving accuracy at a reduced token budget.

Motivated by these insights, we propose MixReasoning, an inference-time framework that adaptively balances conciseness and detail by switching between short-form and long-form reasoning during inference. MixReasoning employs a lightweight LoRA-based distillation to obtain the concise reasoning ability and exposes a single served base model with dynamic adapter strength based on local uncertainty. The design is memory-friendly (only LoRA weights are added, avoid loading multiple models), requires no architectural changes, and allows seamless KV-cache reuse, with only small, bounded prefill overhead when expanding. For inference-time thinking mode switching, MixReasoning monitors token-level uncertainty to decide when detailed reasoning is warranted. When the model is highly uncertain at a given point, the method expands a local window of steps into detailed (long-form) reasoning and then anneals back to the concise (short-form) reasoning for subsequent portions. This yields responses that are both efficient and readable, concise where possible and detailed where necessary. We evaluate MixReasoning across a wide range of reasoning workloads (GSM8K (Cobbe et al., 2021), Math-500 (Lightman et al., 2023) and AIME’24 (Veeraboina, 2023)), spanning tasks of varying complexity and find that CoTs consistently contain substantial redundancy. MixReasoning can compresses reasoning traces without sacrificing accuracy and, in most cases, improves overall accuracy by avoiding verbosity-induced errors. Figure 1 illustrates the comparison among Long-to-short compression, Hybrid reasoning and MixReasoning.

Multi-model speculation vs. MixReasoning. Another acceleration line, multi-model speculative decoding (Pan et al., 2025; Liao et al., 2025; Xia et al., 2024; Yang et al., 2025c), uses a small draft model to propose future tokens that a stronger verifier then accepts or refines. While both methods interleave “modes” during inference, there are key distinctions between the two. Speculative decoding typically loads multiple models and maintains separate KV-caches, primarily reducing per-token

latency by fast-tracking tokens on which the draft and verifier agree; it does not shorten the chain of thought and thus does not target redundancy within long-form reasoning. In contrast, MixReasoning runs a single served base model augmented with lightweight LoRA adapters and scales adapter strength on-the-fly in response to token-level uncertainty, thereby switching between concise and detailed reasoning within one model. This design eliminates multi-model coordination/memory overhead and, crucially, reduces CoT length where appropriate by allocating detail to important segments while keeping easy and routine spans brief, thinking where it matters and yielding responses that are both efficient and more human-readable, so even improve accuracy.

In conclusion, we demonstrate that substantial portions of elaborate reasoning are redundant and inefficient. By switching thinking modes to think based on local uncertainty, MixReasoning can substantially reduce reasoning cost while maintaining capability and, in many cases, improving overall performance.

2 RELATED WORK

Long-to-short Reasoning. LRMs introduce a structured problem-solving approach that breaks down complex problems into multiple simpler reasoning steps, commonly referred to as a long CoT (Wei et al., 2022). This enables the model to generate intermediate reasoning steps before producing a final answer, which can significantly scale inference-time compute. To mitigate this, many works focus on compressing reasoning paths to reduce token generation. Training-free methods prompt models to “think less” (Renze & Guven, 2024; Ma et al., 2025a), intervene during decoding (Wang et al., 2025a;c; Tang et al., 2025), early stopping (Yang et al., 2025b) once answer confidence is high (Yang et al., 2025b), or enforcing hard token budgets to bound rationale length (Sun et al., 2025). Training-based approaches include SFT on synthetic concise traces to teach models shorter explanations (Ma et al., 2025b), and RL with length-aware rewards that penalize long chains (Aggarwal & Welleck, 2025). While effective at lowering latency, these methods apply uniform compression across all problems and steps, which often truncates pivotal reasoning and yields suboptimal accuracy–efficiency trade-offs.

Hybrid Reasoning. An alternative path to efficient reasoning is hybrid reasoning (Fang et al., 2025; Zhang et al., 2025; Anthropic, 2025; Yang et al., 2025a), which routes by instance difficulty: based on problem-wise uncertainty or model confidence, easy cases receive short answers while difficult ones trigger long-form reasoning. This reduces redundancy when many queries admit straightforward solutions and can maintain accuracy on truly hard problems. However, it does not address redundancy within long chains, models tend to remain verbose even across routine substeps, and instance classification is itself hard, since seemingly simple problems may contain localized challenging parts.

Speculative Decoding and Reasoning. Due to the memory-bound nature of LLM decoding, recent work has also leveraged the technique of speculation to accelerate model reasoning (Pan et al., 2025; Liao et al., 2025; Xia et al., 2024; Yang et al., 2025c). Speculation interleaves a fast drafting step with verification by a larger target model, enforcing token-level or semantic-level agreement between a lightweight “draft” model and the base model. These methods reduce time per output token without necessarily shortening the CoT itself, and typically require co-serving both models, increasing memory footprint and operational complexity; consequently, rationales may remain verbose. This line of work is orthogonal to ours: MixReason shortens rationales via intra-CoT adaptive detail selection and can be combined with speculative decoding for additional speedups.

3 METHOD

In this section, we provide an in-depth discussion of our method. Subsection 3.1 introduces a simple yet effective approach that enables the reasoning model to generate concise responses, and then can be seamlessly used to switch thinking mode during inference. Next, in Subsection 3.2, we introduce how we adaptively choose the switching point based on token uncertainty. In Subsection 3.3, we show our method is hardware-friendly with only a single model served.

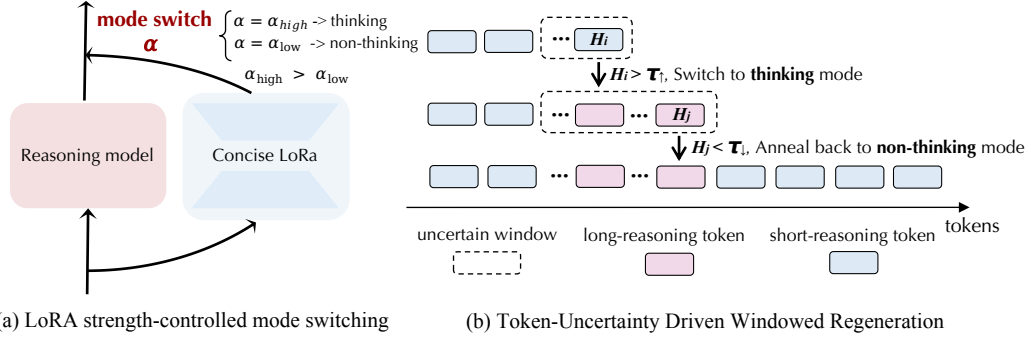


Figure 2: MixReasoning use a single base model served together with a concise LoRA; during decoding we modulate the adapter strength to switch between short-form and long-form reasoning. When token-level uncertainty exceeds a threshold, we expand a local “uncertain window” and regenerate it in long-form mode; once confidence recovers, adapter strength is annealed back and decoding proceeds in the concise mode.

3.1 LoRA-BASED CONTROL OF THINKING MODES

To adaptively vary thinking ability during inference, we need a flexible and reliable control that elicits concise responses without sacrificing accuracy or consistency. Following (Ma et al., 2025b), we obtain such a control via lightweight LoRA SFT that distills a concise-reasoning behavior into the base model. Formally, let θ denote model parameters. For a question q with latent thoughts $\mathbf{t}_{1:n} = \{t_i\}_{i=1}^n$ and final answer a , the original response by the reasoning model $\mathbf{t}_{1:n}$ may contain errors or unnecessary details. Given short (synthesized or human-annotated) explanations $\mathbf{t}_{1:m}$ with $m < n$, we fine-tune to prefer short yet accurate and consistent chains:

$$\max_{\Delta\theta} \mathbb{E}_{(q,a,\mathbf{t}_{1:m}) \sim \mathcal{D}} \left[\log p_{\theta+\Delta\theta}(a \mid \mathbf{t}_{1:m}, q) + \sum_{i=1}^m \log p_{\theta+\Delta\theta}(t_i \mid \mathbf{t}_{<i}, q) \right], \quad (1)$$

where $\Delta\theta$ is a small LoRA update. Because multiple reasoning paths of different lengths can yield the same a , $\Delta\theta$ acts like a task vector that controls CoT length (Ilharco et al., 2022).

At inference (Fig. 2a), we **scale the LoRA adapter strength** α to interpolate between the base model’s longer-chain reasoning and the non-thinking mode, enabling on-the-fly switching within a single served model.

3.2 TOKEN UNCERTAINTY-BASED MODE SWITCHING.

To decide when to expand detail, we monitor token-level uncertainty during decoding and switch modes only at locally pivotal steps. As suggested by Fig. 2(b) and our token entropy insights, reasoning complexity is heterogeneous within a CoT: most tokens are routine low-entropy completions, whereas occasional high-entropy tokens act as decision forks that steer the trajectory among plausible pathways (Wang et al., 2025b). We therefore use next-token probability distribution to detect these forks and allocate long-form reasoning locally. Let $x_{1:t}$ be the partial output and $p_t(v) = p_\theta(v \mid x_{1:t})$ the next-token distribution; the normalized uncertainty score is

$$H_t = - \sum_{v \in \mathcal{V}} p_t(v) \log p_t(v) / \log |\mathcal{V}|. \quad (2)$$

When local uncertainty is high ($H_t \geq \tau_\uparrow$), we open an *uncertainty window*

$$W_t = [t-B, t+F],$$

roll back to its left boundary $t-B$, and re-decode all tokens in W_t under **thinking mode** by setting the LoRA strength to a higher value α_{high} (trained in Sec. 3.1). Outside windows we default to **non-thinking (concise) mode** with a lower adapter strength α_{low} . To avoid oscillations, we employ a hysteresis schedule with $\tau_\downarrow < \tau_\uparrow$: after finishing a window, we keep **thinking** as long as uncertainty

remains above the lower threshold, and only anneal back to non-thinking mode (Concise reasoning) when $H_t \leq \tau_\downarrow$.

Formally, We maintain $S_t \in \{\alpha_{\text{low}}, \alpha_{\text{high}}\}$ (α_{low} : thinking; α_{high} : concise). The mode follows a hysteresis rule:

$$S_{t+1} = \begin{cases} \alpha_{\text{low}}, & (S_t = \alpha_{\text{high}} \wedge H_t \geq \tau_\uparrow) \vee (S_t = \alpha_{\text{low}} \wedge H_t > \tau_\downarrow), \\ \alpha_{\text{high}}, & \text{otherwise.} \end{cases} \quad (3)$$

When the first branch applies with $S_t = \alpha_{\text{high}}$ (i.e., we enter thinking), we perform windowed regeneration: set $W_t = [t-B, t+F]$, roll back to $t-B$, and decode all $u \in W_t$ with $S_u = \alpha_{\text{low}}$; outside W_t , decoding proceeds under the current S .

This single-model, windowed regeneration concentrates long-form reasoning on high-uncertainty forks while keeping low-uncertainty spans brief. Crucially, the ratio of modes, and thus the overall response length, can be explicitly controlled by (B, F) , the trigger thresholds $(\tau_\uparrow, \tau_\downarrow)$, and the anneal rate η : larger windows or more sensitive triggers yield more thinking tokens (longer, more detailed outputs), whereas tighter windows or stricter triggers favor conciseness. Empirically, this MixReasoning scheme improves efficiency at a reduced token budget and mitigates verbosity-induced errors and overthinking common in pure long-form decoding. Moreover, it remedies a key limitation of long-to-short compression, which applies a uniform shortening policy across problems and steps and thus often truncates decision-critical reasoning by allocating detail only when uncertainty is high; as a result, routine spans are compressed while pivotal tokens are preserved, yielding superior accuracy-efficiency trade-offs.

This single-model, windowed regeneration concentrates expansion on high-entropy(uncertain) forks while keeping low-entropy spans brief, and thus improves efficiency at a reduced token budget by thinking where it matters. Empirically, this conciseness-detail balancing MixReasoning not only makes reasoning more efficient but also effectively mitigates verbosity-induced errors and overthinking commonly observed in long-form reasoning.

3.3 KV-CACHE REUSE AND PREFILL OVERHEAD.

A practical advantage of our approach is that it serves a single model and toggles a lightweight LoRA adapter at inference time, instead of coordinating multiple models as in multi-model speculation (Pan et al., 2025; Liao et al., 2025; Yang et al., 2025c). We maintain a scalar LoRA; the base model weights stay fixed and only the adapter strength is switched, so memory footprint and scheduling remain comparable to standard single-model decoding. When switching from non-thinking to thinking, we perform a one-time prefill over the existing prefix to seed the thinking KV states, then continue decoding. When switching back, we reuse the concise KV states built before the switch and prefill only the new tokens produced in the thinking segment. This avoids recomputing attention on already processed tokens, so switching cost scales only with the switched span prefilling and remains a small fraction of end-to-end latency. In practice, prefill is highly efficient, being parallelizable and memory-bound, so a long prefilling often takes roughly the wall-clock time of generating only 1–2 tokens in auto-regressive manner (Pan et al., 2025).

As a further optimization (not the focus of this paper), placing LoRA only on MLP layers while leaving attention k/v untouched would allow full KV-cache reuse across mode switches, making runtime behavior essentially identical to a single fixed- α decode; see Sec. 4.4 for analysis motivating this choice.

Overall, MixReasoning is runtime-efficient: it avoids multi-model residency and cross-model context shuttling, reuses most of the KV cache, and confines extra compute to small, uncertainty-localized rollbacks, preserving high throughput while enabling detail where it matters.

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUPS

Models. We test our method across multiple model families and scales. In addition to QwQ-32B-Preview (Team, 2024), a common baseline in prior work, we evaluate two recent state-of-the-art

open-source models, Qwen-3-14B and Qwen-3-8B (Alibaba, 2025), to examine generality across parameter counts and architectures.

Benchmarks. We evaluate on widely used reasoning benchmarks: GSM8K (Cobbe et al., 2021) (Grade School Math; test split with 1,319 word problems), AIME’24 (Veeraboina, 2023) (30 problems from the 2024 American Invitational Mathematics Examination), and Math-500 (Lightman et al., 2023) (a 500-problem subset of the MATH benchmark). This suite spans a broad difficulty range, from elementary word problems to competition-level mathematics, providing a comprehensive assessment of both accuracy and efficiency.

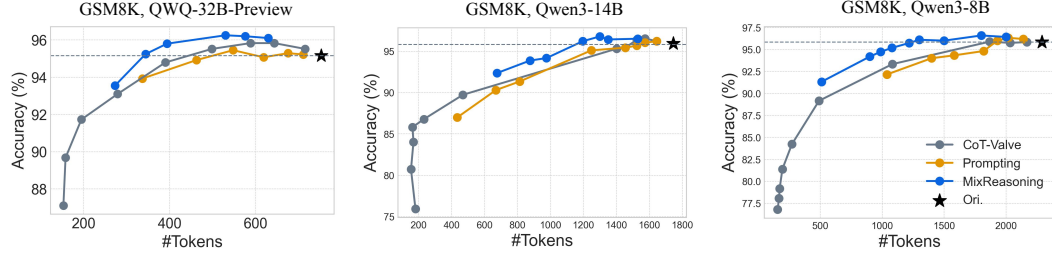


Figure 3: MixReasoning and Long-to-short reasoning(prompting, finetuning(CoT-Valve)) results on GSM8K dataset with QwQ-32B-Preview, Deepseek-R1-14B and Qwen3-8B. MixReasoning can achieve a better trade-off between efficiency and accuracy.

4.2 MAIN RESULTS ON MIXREASONING

Finding 1. MixReasoning improves the accuracy–efficiency Pareto frontier.

Across GSM8K, MATH-500, and AIME’24 (Fig. 3; Tab. 1), MixReasoning yields shorter traces at equal or higher accuracy by allocating thinking only at high-uncertainty steps. At matched accuracy, it cuts tokens substantially (e.g., QwQ-32B-Preview on GSM8K: -47% ; Qwen-3-8B on GSM8K: -45% ; MATH-500 on QwQ-32B-Preview: -26%). At matched budget, it matches or exceeds the accuracy of long-to-short compression baselines.

In experiments, we compare against representative length-control methods that uniformly shorten chains: (i) Prompting family (Han et al., 2024), which steers toward brevity via instruction templates; (ii) CoT-Valve (Ma et al., 2025b), LoRA-based SFT on synthetic concise traces with length modulated by α ; (iii) DEER (Yang et al., 2025b), dynamic early stopping during reasoning; (iv) NoWait (Wang et al., 2025a), removal/pruning of thinking tokens; and (v) ConciseHint (Tang et al., 2025), inference-time interventions to reduce output length.

A key advantage is controllability: sweeping the uncertainty threshold and window size moves points smoothly along Fig. 3, enabling per-request budget control without model swapping. By preserving decision-critical segments that uniform compression often truncates, MixReasoning attains superior accuracy–efficiency trade-offs and more readable traces at the same budget.

The gains stem from selective detail: MixReasoning expands reasoning only at pivotal steps (identified by local uncertainty) and stays concise on routine spans, directly addressing redundancy within long trajectories. The uncertainty threshold τ and window size W smoothly trade accuracy for cost (larger W or higher τ engages detailed mode more often), giving practitioners budget-aware control without retraining. The design is training-light and memory-friendly (LoRA weights only) and composes with standard decoding, making it practical for real-world deployments.

Finding 2. Window size and uncertainty threshold can control the mix of modes and thus the total token budget.

MixReasoning exposes two runtime knobs that deterministically shape the ratio between thinking and concise spans during decoding: (i) an uncertainty threshold τ_{\uparrow} (on entropy or confidence) that triggers thinking, and (ii) a window size $W = [B, F]$ that sets how long we persist in thinking

Table 1: MixReasoning and Long-to-short compression methods results on GSM8K, AIME24, and Math500 with QwQ-32B-Preview, Qwen3-14B and Qwen3-8B. Ori. denotes the original reasoning process without extra prompt, training or our method. We report the average accuracy and token usage.

Models	Methods	GSM8K		Math-500		AIME 2024	
		Pass@1	#Tokens	Pass@1	#Tokens	Pass@1	#Tokens
QwQ-32B-Preview	Ori.	0.9512	750.3	0.8937	2230	0.4333	6827
	Prompt	0.9365	378.2	0.8734	1703	0.4000	6102
	DEER	—	—	—	—	—	—
	NoWait	—	—	—	—	—	—
	CoT-Valve	0.9421	352.8	0.8633	1756	0.4000	5975
	ConciseHint	—	—	—	—	—	—
	MixReasoning	0.9613	400.5	0.8986	1646	0.4483	5277
Qwen3-14B	Ori.	0.9593	1745	0.9360	4516	0.6444	11478
	Prompt	0.9510	1248	0.9233	4071	0.6500	10693
	DEER	0.9530	957	0.9400	3074	0.6834	7894
	NoWait	0.9598	1076	0.9340	3332	0.6881	8786
	CoT-Valve	0.9573	1401	0.9133	3933	0.5998	10692
	ConciseHint	0.9601	1493	0.9248	3654	0.6533	10184
	MixReasoning	0.9621	1196	0.9410	3476	0.6789	9431
Qwen3-8B	Ori.	0.9583	2239	0.9320	5192	0.6333	12205
	Prompt	0.9382	1619	0.9205	4391	0.5897	11481
	DEER	0.9520	1071	0.9260	3032	0.6100	9017
	CoT-Valve	0.9482	1622	0.9275	4591	0.5967	11281
	NoWait	0.9538	1406	0.9240	3232	0.6181	10786
	ConciseHint	0.9553	1593	0.9198	3143	0.6417	11228
	MixReasoning	0.9562	1217	0.9313	3531	0.6433	10738

once triggered. Increasing sensitivity (larger τ_{\uparrow} or stricter confidence cutoff) or enlarging W raises the coverage $c(\tau_{\uparrow}, W)$ of thinking tokens; decreasing them yields more concise traces. With fixed $(\alpha_{\text{low}}, \alpha_{\text{high}})$ for the two modes, this controller provides per-request control of response length without retraining or model swapping.

Empirically (Fig. 3), sweeping (τ_{\uparrow}, W) moves points smoothly along the accuracy–efficiency curve: higher coverage improves accuracy at a predictable token cost, while lower coverage favors brevity with minimal loss. Compared to long-to-short compression, which applies a uniform shortening policy across problems and steps, our uncertainty-gated windows preserve decision-critical segments and compress routine spans, yielding strictly better trade-offs at matched budgets. In practice, a service can expose a single “budget slider” that tunes τ_{\uparrow} (optionally with a cap on W for latency predictability) to meet different usage scenarios and readability preferences. A qualitative illustration is given in Fig. 4: a small window produces a short, highly readable response with low thinking coverage, whereas a larger window yields a longer answer with more detailed reasoning, reflecting the controlled mode ratio.

4.3 QUALITATIVE EXAMPLE: THINKING WHERE IT MATTERS

We contrast Long CoT with two MixReasoning settings on a representative GSM8K instance: a *small uncertainty window* and a *large uncertainty window*. Long CoT meanders through step-by-step musings and repeated sanity checks, inflating sequence length and risking off-by-one or interpretation slips. In both MixReasoning variants, the model detects the high-uncertainty fork, opens a local window to reason in detail, resolves the ambiguity, and then anneals back to concise mode for routine arithmetic. Additional examples generated by Qwen3-8B and Qwen3-14B (Yang et al., 2025a) can be seen in Appendix A.3.

This example illustrates three takeaways aligned with our goal of “thinking where it matters”: (i) **Switching modes to think**: detailed reasoning is reserved for genuinely hard, decision-critical steps, while routine spans stay concise, avoiding the uniform elaboration of Long CoT; (ii) **Controllable budget**: the window size W directly tunes the mix of modes, small W makes non-thinking dominate (shorter, highly readable traces), whereas large W increases thinking coverage when more rationale

Question: Josh decides to try flipping a house. He buys a house for \$80,000 and then puts in \$50,000 in repairs. This increased the value of the house by 150%. How much profit did he make?

Long CoT (QwQ-32B-Preview)

#Token: 2157

Let's see what Josh did here. **<Omitted>** Hmm, so I need to understand what that means. A 150% increase on the original value? Let me think carefully **<Omitted>** Wait, no. A 150% increase means the value increases by 150% of the original value, so if the original value is \$80,000, the increase is \$120,000, making the new value \$200,000. **<Omitted>** But according to this, the repairs cost \$50,000, but the value increased by \$120,000, which seems high. **<Omitted>** Alternatively, perhaps the increase is based on the cost of repairs. **<Omitted>** But that doesn't make sense in this context. I think the first interpretation is correct. So, Josh made a profit of \$25,000.

****Final Answer**:** $\boxed{25000}$ **\ (Wrong answer)**

MixReasoning (small uncertain window)

#Token: 247

Let's see. Josh bought a house for \$80,000 and then spent \$50,000 on repairs. That means his total investment is \$80,000 + \$50,000 = \$130,000. The repairs increased the value of the house by 150%. So, the increase in value is 150% of the original price, which is $1.5 * \$80,000 = \$120,000$. Therefore, the new value of the house is the original price plus the increase, which is $\$80,000 + \$120,000 = \$200,000$. To find the profit, we subtract the total investment from the new value: $\$200,000 - \$130,000 = \$70,000$. So, Josh made a profit of \$70,000.

****Final Answer**** $\boxed{70000}$ **\ (Right answer)**

MixReasoning (large uncertain window)

#Token: 410

So Josh bought a house for \$80,000 and then spent \$50,000 on repairs. That means his total investment is $\$80,000 + \$50,000 = \$130,000$. The repairs increased the value of the house by 150%. So, the increase in value is 150% of the original price, which is 150% of \$80,000. First, I need to calculate 150% of \$80,000. 150% is the same as 1.5 in decimal form. So, $1.5 * \$80,000 = \$120,000$. That means the value of the house increased by \$120,000. So, the new value of the house is the original price plus the increase: $\$80,000 + \$120,000 = \$200,000$. Profit = new value - total investment = $\$200,000 - \$130,000 = \$70,000$. Wait a minute, does that make sense? New value is $\$80,000 + \$120,000 = \$200,000$. Profit is $\$200,000 - \$130,000 = \$70,000$. I think that's correct.

****Final Answer**** $\boxed{70000}$ **\ (Right answer)**

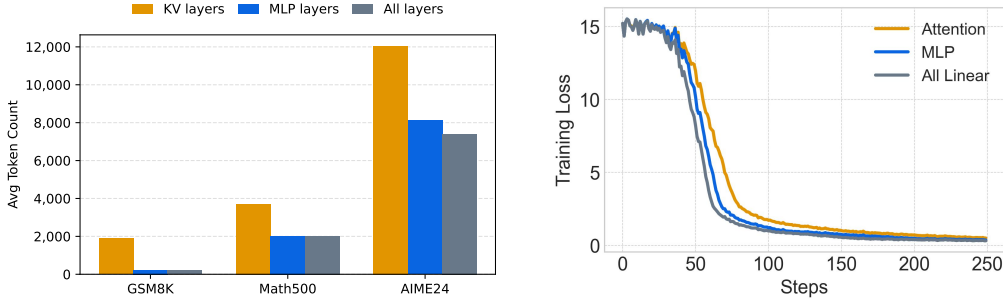
Figure 4: Qualitative comparison: Long CoT produces a verbose trace with coherence fillers and redundant self-checks. MixReasoning (small window) expands only at the high-uncertainty fork and then anneals back to concise mode, reaching the correct answer with a substantially shorter trace. MixReasoning (large window) allocates more detailed reasoning across adjacent steps, trading a larger budget for additional rationale while staying focused around the pivotal region. In MixReasoning responses, we highlight thinking mode tokens with red background and non-thinking mode tokens with blue background.

is desired thus controlling total tokens within a single served model; (iii) **Readability**: by eliminating filler and redundant self-checks by non-thinking mode, MixReasoning yields human-like explanations that emphasize the pivotal inference and suppress distracting verbosity.

Together with the quantitative results (Fig. 3), this case study shows that adjusting W lets us dial the level of elaboration to the task and user preference, achieving shorter, more readable traces than Long CoT at matched or better accuracy.

4.4 DETAILS OF CONCISE MODE LoRA FINETUNING.

To distill the non-thinking mode from LRMs, we train a lightweight LoRA adapter (Hu et al., 2022) on the GSM8K train split (7.47k problems; no test data are used). GSM8K's supervision naturally provides very short ground-truth solutions often answer-only or minimal one-two step rationales which makes it well-suited for learning a brevity preference without degrading correctness. Concretely, we freeze the backbone and fine-tune only the LoRA parameters using short-rationale/answer supervision, biasing the model toward brief, accurate traces. In practice, this simple recipe is highly effective: training converges quickly (Fig. 5 b), and the distilled adapter consistently reduces thinking length on GSM8K, MATH-500, and AIME (Fig. 5 a). The resulting concise adapter is then used as the runtime control in MixReasoning to modulate reasoning depth during inference.



(a) Token length under different LoRA targets.

(b) Training loss curves: all converge similarly.

Figure 5: Layerwise LoRA ablation for reasoning-chain compression. Fine-tuning only MLP layers achieves token-length compression comparable to fine-tuning all layers, despite similar training convergence across configurations. In contrast, attention K/V-only adapters provide little compression, suggesting that knowledge governing reasoning-path length resides primarily in MLPs. This motivates an MLP-only adapter design that enables full KV-cache reuse during mode switching.

Finding 3. Reasoning length and structure are governed by MLPs rather than attention K/V.

Layerwise contributions to reasoning-chain length. We isolate the effect of layer types by applying LoRA fine-tuning to specific components of LRMs: (i) MLP-only, (ii) attention-only (K/V projections), and (iii) all layers. As shown in Fig. 5b, all settings converge in teacher-forced training on GSM8K solutions (short CoT supervision). However, their downstream behavior differs markedly (Fig. 5a): **MLP-only** fine-tuning achieves nearly the same compression efficiency (token reduction at matched accuracy) as fine-tuning all layers, whereas **attention-only (K/V)** fine-tuning yields little or no compression. This indicates that the knowledge governing the length and structure of the reasoning path is concentrated in MLPs rather than in attention K/V, echoing prior findings that factual/associative content is stored predominantly in feed-forward pathways while attention chiefly routes or transforms information (Geva et al., 2020; Meng et al., 2022).

Beyond offering a clearer mechanism for reasoning-chain compression, this result suggests a practical design for low inference-overhead mode switching. In our MixReasoning implementation, we fine-tune all layers. Compared with multi-model speculation, MixReasoning is already much more efficient because switching between adapters only incurs occasional refill overhead, which is minor in practice. Nevertheless, if adapters are restricted to MLP-only while keeping attention K/V unchanged, the KV-cache can be fully reused across modes (MoE-style), so the end-to-end inference cost would be essentially indistinguishable from running a single model. This provides a promising direction to further reduce inference overhead. In this paper, our primary focus is to compress the reasoning length without sacrificing accuracy, achieving balance between concise and detailed reasoning.

5 CONCLUSION

we demonstrate that substantial portions of elaborate reasoning are redundant and inefficient and introduced **MixReasoning**, a training-light, model-agnostic framework that allocates reasoning detail within a chain of thought by adaptively switching between thinking and non-thinking modes at the step level. Using a LoRA-based concise mode, dynamic adapter strength, and an uncertainty-triggered sliding window, MixReasoning preserves pivotal derivations while compressing routine spans, enabling KV-cache reuse with minimal prefilling overhead. Across GSM8K, MATH-500, and AIME, it reduces thinking tokens while maintaining or improving accuracy, consistently outperforming long-to-short at matched budgets or even baseline reasoning models. The core principle is simple: *think where it matters*.

LLM DISCLAIMER

In this work, large language models (LLMs) were used solely for non-technical purposes, specifically to assist with literature review and to refine the readability of the manuscript. Their use was limited to phrasing and presentation. No technical contributions, including methodological design, model implementation, or experimental analysis, involved LLMs.

REFERENCES

- Pranjal Aggarwal and Sean Welleck. L1: Controlling how long a reasoning model thinks with reinforcement learning. *arXiv preprint arXiv:2503.04697*, 2025.
- Alibaba. <https://qwenlm.github.io/blog/qwen3/>, 2025.
- Anthropic. Claude 3.7 Sonnet. <https://www.anthropic.com/claude/sonnet>, 2025. Accessed: 2025-05-10.
- Zigeng Chen, Xinyin Ma, Gongfan Fang, Ruonan Yu, and Xinchao Wang. Verithinker: Learning to verify makes reasoning model efficient. *arXiv preprint arXiv:2505.17941*, 2025.
- Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.
- Gongfan Fang, Xinyin Ma, and Xinchao Wang. Thinkless: Llm learns when to think. *arXiv preprint arXiv:2505.13379*, 2025.
- Yichao Fu, Junda Chen, Siqi Zhu, Zheyu Fu, Zhongdongming Dai, Aurick Qiao, and Hao Zhang. Efficiently serving llm reasoning programs with certindex. *arXiv e-prints*, pp. arXiv–2412, 2024.
- Yichao Fu, Junda Chen, Yonghao Zhuang, Zheyu Fu, Ion Stoica, and Hao Zhang. Reasoning without self-doubt: More efficient chain-of-thought through certainty probing. In *ICLR Workshop on Foundation Models in the Wild*, 2025.
- Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. Transformer feed-forward layers are key-value memories. *arXiv preprint arXiv:2012.14913*, 2020.
- Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.
- Tingxu Han, Zhenting Wang, Chunrong Fang, Shiyu Zhao, Shiqing Ma, and Zhenyu Chen. Token-budget-aware llm reasoning. *arXiv preprint arXiv:2412.18547*, 2024.
- Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3, 2022.
- Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Suchin Gururangan, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. Editing models with task arithmetic. *arXiv preprint arXiv:2212.04089*, 2022.
- Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*, 2023.
- Baohao Liao, Yuhui Xu, Hanze Dong, Junnan Li, Christof Monz, Silvio Savarese, Doyen Sahoo, and Caiming Xiong. Reward-guided speculative decoding for efficient llm reasoning. *arXiv preprint arXiv:2501.19324*, 2025.
- 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 *The Twelfth International Conference on Learning Representations*, 2023.

- Wenjie Ma, Jingxuan He, Charlie Snell, Tyler Griggs, Sewon Min, and Matei Zaharia. Reasoning models can be effective without thinking. *arXiv preprint arXiv:2504.09858*, 2025a.
- Xinyin Ma, Guangnian Wan, Runpeng Yu, Gongfan Fang, and Xinchao Wang. Cot-valve: Length-compressible chain-of-thought tuning. *arXiv preprint arXiv:2502.09601*, 2025b.
- Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in gpt. *Advances in neural information processing systems*, 35:17359–17372, 2022.
- Rui Pan, Yinwei Dai, Zhihao Zhang, Gabriele Oliaro, Zhihao Jia, and Ravi Netravali. Specreason: Fast and accurate inference-time compute via speculative reasoning. *arXiv preprint arXiv:2504.07891*, 2025.
- Matthew Renze and Erhan Guven. The benefits of a concise chain of thought on problem-solving in large language models. In *2024 2nd International Conference on Foundation and Large Language Models (FLLM)*, 2024.
- Yi Sun, Han Wang, Jiaqiang Li, Jiacheng Liu, Xiangyu Li, Hao Wen, Yizhen Yuan, Huiwen Zheng, Yan Liang, Yuanchun Li, et al. An empirical study of llm reasoning ability under strict output length constraint. *arXiv preprint arXiv:2504.14350*, 2025.
- Siao Tang, Xinyin Ma, Gongfan Fang, and Xinchao Wang. Concisehint: Boosting efficient reasoning via continuous concise hints during generation. *arXiv preprint arXiv:2506.18810*, 2025.
- Qwen Team. Qwq: Reflect deeply on the boundaries of the unknown. *Hugging Face*, 2024.
- Hemish Veeraboina. Aime problem set 1983-2024, 2023. URL <https://www.kaggle.com/datasets/hemishveeraboina/aime-problem-set-1983-2024>.
- Chenlong Wang, Yuanning Feng, Dongping Chen, Zhaoyang Chu, Ranjay Krishna, and Tianyi Zhou. Wait, we don’t need to” wait”! removing thinking tokens improves reasoning efficiency. *arXiv preprint arXiv:2506.08343*, 2025a.
- Shenzhi Wang, Le Yu, Chang Gao, Chujie Zheng, Shixuan Liu, Rui Lu, Kai Dang, Xionghui Chen, Jianxin Yang, Zhenru Zhang, et al. Beyond the 80/20 rule: High-entropy minority tokens drive effective reinforcement learning for llm reasoning. *arXiv preprint arXiv:2506.01939*, 2025b.
- Yue Wang, Qiuzhi Liu, Jiahao Xu, Tian Liang, Xingyu Chen, Zhiwei He, Linfeng Song, Dian Yu, Juntao Li, Zhuosheng Zhang, et al. Thoughts are all over the place: On the underthinking of o1-like llms. *arXiv preprint arXiv:2501.18585*, 2025c.
- Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.
- Heming Xia, Yongqi Li, Jun Zhang, Cunxiao Du, and Wenjie Li. Swift: On-the-fly self-speculative decoding for llm inference acceleration. *arXiv preprint arXiv:2410.06916*, 2024.
- An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025a.
- Chenxu Yang, Qingyi Si, Yongjie Duan, Zheliang Zhu, Chenyu Zhu, Qiaowei Li, Zheng Lin, Li Cao, and Weiping Wang. Dynamic early exit in reasoning models. *arXiv preprint arXiv:2504.15895*, 2025b.
- Wang Yang, Xiang Yue, Vipin Chaudhary, and Xiaotian Han. Speculative thinking: Enhancing small-model reasoning with large model guidance at inference time. *arXiv preprint arXiv:2504.12329*, 2025c.
- Jiajie Zhang, Nianyi Lin, Lei Hou, Ling Feng, and Juanzi Li. Adaptthink: Reasoning models can learn when to think. *arXiv preprint arXiv:2505.13417*, 2025.

A APPENDIX

A.1 LIMITATIONS AND FUTURE WORKS

MixReasoning relies on local uncertainty signals (token-level entropy) to trigger expansion; while training-light, this controller does not learn end-to-end where within a CoT to be long vs. short and can be sensitive to calibration and non-local dependencies. Future work will replace or augment it with a learned policy (e.g., imitation learning or RL with length-accuracy rewards) and combine MixReasoning with complementary methods, problem-level hybrid routing, long-to-short compression, and speculative decoding, to jointly reduce redundancy in long chains while preserving accuracy and controllability.

A.2 IMPLEMENTATION DETAILS

Evaluation Metrics For all models, we use vLLM (Kwon et al., 2023) to generate responses on 4 A100 GPUs. We report results using the flexible-match metric. Specifically, we first extract the value enclosed within `\boxed{\}`. If no such boxed answer is found, we default to using the last digit in the response as the final answer. All results in Table 1 are the mean over 5 independent runs. Following prior work (Ma et al., 2025b), we set the maximum tokens for QwQ-32B-Preview (Team, 2024) to 4192 on the GSM8K dataset, and 8192 on the MATH-500 and AIME24 datasets at inference time. For the recent state-of-the-art open-source models Qwen3-8B and Qwen3-14B (Alibaba, 2025), we set the maximum tokens to 16384 for all three benchmarks.

Training Setting We use LoRA (Hu et al., 2022) fine-tuning to distill the non-thinking mode. The dataset is the GSM8K training split (7.47k problems; no test data are used). We use the GSM8K ground truth (very short solutions, often answer-only or with a minimal one-two-step rationale) as the target, without regenerating answers or rationales. All models are trained on 4 A100 80GB GPUs. We set the batch size to 64 and train for up to ten epochs. The learning rate is 1×10^{-5} , with a weight decay of 0.01. For LoRA, the rank is set to 2, and the adapter strength α is set to 8.

A.3 ADDITIONAL QUALITATIVE EXAMPLES

To verify that the behavior in Sec. 4.3 is not backbone-specific, we also compare on Qwen3-8B and Qwen3-14B (Figs. 6–7). In both backbones, MixReasoning triggers expansion only when next-token uncertainty crosses the upper threshold, opens a bounded window W around the high-entropy fork (red spans), regenerates the local segment in the thinking mode, and then anneals back to the concise (non-thinking) mode once uncertainty subsides. Qualitatively, the detailed spans align with decision-critical operations (e.g., choosing the arithmetic step, selecting a case/branch, or performing a delicate algebraic rearrangement), while routine low-entropy steps remain terse. This pattern traces that are shorter and easier to follow than Long CoT, yet faithfully preserve the pivotal reasoning that determines correctness.

Question: Emily has 4 kids named Amy, Jackson, Gorey, and James. Amy is 5 years older than Jackson and 2 years younger than Gorey. If James is 10 and is 1 year younger than Gorey, how old is Jackson?

Long CoT (Qwen3-8B)

#Token: 1297

<think>
 Okay, let's try to figure out how old Jackson is. <Omitted> Wait, let me check the original problem again: <Omitted>
 "Emily has 4 kids named Amy, Jackson, Corey, and James. <Omitted> Amy is 5 years older than Jackson and 2 years
 younger than Corey. <Omitted> If James is 10 and is 1 year younger than Corey, how old is Jackson?" <Omitted>
 <Omitted> Check if all conditions are satisfied: <Omitted> To determine Jackson's age, <Omitted> Amy is 2 years
 younger than Corey: $\text{Amy} = \text{Corey} - 2 = 11 - 2 = 9$ So, Amy is 9 years old.
 Final Answer: $\boxed{9}$ (Wrong answer)

MixReasoning

#Token: 79

Okay, let's try to figure out this problem.
 Corey is $10 + 1 = 11$ years old.
 Amy is $11 - 2 = 9$ years old.
 Jackson is $9 - 5 = 4$ years old.
 —
 Final Answer $\boxed{4}$ (Right answer)

Figure 6: Additional qualitative comparison using Qwen3-8B model. In MixReasoning responses, we highlight thinking mode generated tokens with red background and non-thinking mode generated tokens with blue background.

Jeremy saw 12 birds in their backyard and threw a stone at them, scaring away 1/3 of that number. A few minutes later, 20 more birds joined the fearless birds. How many birds are now in the backyard?

Long CoT (Qwen3-14B)

#Token: 867

Okay, let's see. So Jeremy saw 12 birds in the backyard. Then he threw a stone, which scared away 1/3 of that number. Hmm, so first I need to figure out how many birds were scared away. So, 1/3 of 12. Let me calculate that. 12 divided by 3 is 4. So, 4 birds were scared away. That means the number of birds left after the stone was thrown would be the original 12 minus 4. Let me write that down: $12 - 4 = 8$. So, there are 8 birds remaining in the backyard after Jeremy threw the stone. Then, a few minutes later, 20 more birds joined the fearless ones. Wait, the problem says "the fearless birds." So, does that mean the ones that weren't scared away? Yeah, probably. So, the 8 birds that were left are now joined by 20 more. So, adding those together: $8 + 20 = 28$. Let me do that math. $8 + 20 = 28$. So, the total number of birds now in the backyard should be 28. Let me just check my steps again to make sure I didn't make a mistake. Original number: 12. Scared away 1/3, which is 4. $12 - 4 = 8$. Then 20 more join, so $8 + 20 = 28$. Yeah, that seems right. I don't think I missed anything here. The problem didn't mention any other birds leaving or anything else happening. So, I think the answer is 28. </think> <Omitted> $\frac{1}{3} \times 12 = 4$ This means $12 - 4 = 8$ birds remained after the stone was thrown. A few minutes later, 20 more birds joined the remaining 8 birds. Adding these together: $8 + 20 = 28$
 Final Answer: $\boxed{28}$ (Right answer)

MixReasoning

#Token: 499

Okay, let's see. Jeremy saw 12 birds in the backyard. Then he threw a stone, and scared away 1/3 of that number. Hmm, so first I need to figure out how many birds were scared away. So, 1/3 of 12. Let me calculate that. 12 divided by 3 is 4. So, 4 birds were scared away. That means the number of birds left after the stone was thrown would be the original 12 minus 4. Let me write that down: $12 - 4 = 8$. So, there are 8 birds remaining in the backyard after Jeremy threw the stone. Then, a few minutes later, 20 more birds joined the fearless ones. Wait, the problem says "the fearless birds." So, does that mean the ones that weren't scared away? Yeah, probably. So, the 8 birds that were left are now joined by 20 more. So, adding those together: $8 + 20 = 28$. Let me do that math. $8 + 20 = 28$. So, the total number of birds now in the backyard should be 28. Let me just check my steps again to make sure I didn't make a mistake. Original number: 12. Scared away 1/3, which is 4. $12 - 4 = 8$. Then 20 more join, so $8 + 20 = 28$. Yeah, that seems right. I don't think I missed anything here. The problem didn't mention any other birds leaving or anything else happening. So, I think the answer is 28. </think> Jeremy initially saw 12 birds in the backyard. When he threw a stone, 1/3 of the birds were scared away. 1. **Birds scared away**: $\frac{1}{3} \times 12 = 4$ Remaining birds: $12 - 4 = 8$ 2. **Additional birds joining**: 20 more birds joined the remaining 8 birds. Total birds now: $8 + 20 = 28$
 Final Answer $\boxed{28}$ (Right answer)

Figure 7: Additional qualitative comparison using Qwen3-14B model. In MixReasoning responses, we highlight thinking mode generated tokens with red background and non-thinking mode generated tokens with blue background.