CLaM: Bridging Explicit and Implicit Chain-of-Thought via Controllable Latent Mediation

Anonymous ACL submission

Abstract

Chain-of-Thought (CoT) improves multi-step reasoning in LLMs, but it highlights a tension between efficiency and controllability: explicit CoT exposes a rationale channel that can be edited at inference time to steer outputs, whereas implicit/latent CoT internalizes reasoning and operates under an answer-only interface, making targeted intervention difficult. We characterize this gap with a counterfactual framework that distinguishes input-level perturbations from mediator-level interventions, and show empirically that explicit and implicit systems can appear similar under input counterfactuals yet diverge sharply when direct control over intermediate reasoning is required. Motivated by this boundary, we propose **CLaM** (**C**ontrollable **La**tent **M**ediation), which restores an intervention handle for implicit reasoning without emitting rationales: an extractor maps structured intermediate facts into a small set of latent mediator embedding that condition an answer-only student model. Across multiple backbones and editing settings, CLaM enables robust counterfactual interventions and reliable propagation from edited intermediates to final answers, improving controllability while preserving the efficiency of latent reasoning. Our data and code will be available at https://github.com/XXX.

1 Introduction

Large language models (LLMs) have recently shown strong performance on tasks requiring multi-step reasoning, with *Chain-of-Thought* (CoT) serving as a widely adopted catalyst (Yang et al., 2025a; Guo et al., 2025; Yu et al., 2025). As illustrated in Figure 1, the standard paradigm is *explicit* CoT: models are trained on (Q, CoT, A) triples (e.g., CoT-SFT) and generate a human-readable rationale before the final answer. Beyond improving accuracy, this explicit rationale channel provides a practical control handle at inference time—users can revise, swap, or constrain the rationale to steer the model's prediction under the same question input. However, explicit CoT also incurs longer generations and higher inference cost, and supervised rationale imitation can encourage superficial linguistic shortcuts. These limitations have motivated *implicit* (or *latent*) CoT (Shen et al., 2025; Tan et al., 2025; Xu et al., 2025), which internalizes reasoning into hidden states or a small number of continuous "thought" tokens while keeping the external interface *answer-only*. Recent works (Shen et al., 2025; Xu et al., 2025) show that compressing CoT into continuous latent tokens can retain competitive reasoning performance while improving efficiency, often by aligning representations between explicit and implicit modes.

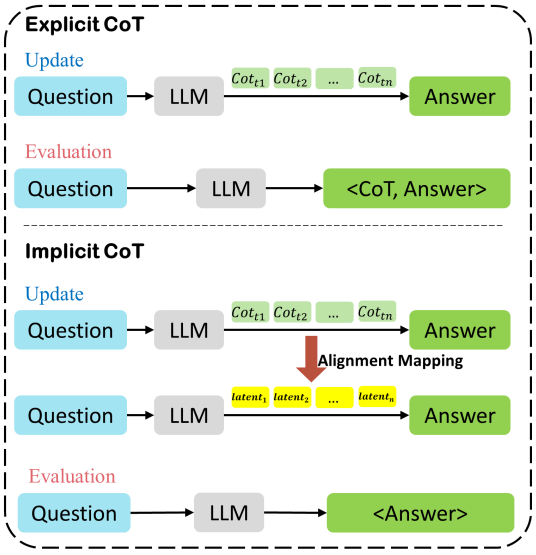

Figure 1: Illustration of explicit vs. implicit CoT. Explicit CoT models are trained on (Q, CoT, A) and output textual rationales at inference, while implicit CoT internalizes reasoning in latent representations and produces answers without exposing CoT.

However, this shift from language to latent space raises a core question that is under-explored: *what*

is gained and what is lost when reasoning is internalized? In particular, explicit CoT provides an intervention surface that can be manipulated externally, while implicit CoT may trade away controllability for compactness. This matters beyond interpretability: many downstream applications (e.g., knowledge editing (Yao et al., 2023; Zhang et al., 2024), counterfactual reasoning (Zhong et al., 2023), and multi-hop propagation (Cohen et al., 2024)) require changing a specific intermediate fact or substep while keeping the rest of the model stable. If the reasoning mechanism is internalized and not directly addressable, such targeted interventions become difficult.

Building on this empirical motivation, we formulate two hypotheses:

- $\mathcal{H}_1$ **(input-level observational equivalence)** posits that on the original input and input-level counterfactuals, explicit and implicit systems can be approximately equivalent when evaluated through an answer-only interface.
- $\mathcal{H}_2$ **(interventional non-equivalence)** posits that the two systems diverge under mediator-level intervention, because only explicit CoT provides a well-defined, externally implementable $\mathrm{do}(CoT)$ operation.

Two sets of experiments are conducted on GSM8K-Aug (Cobbe et al., 2021; Deng et al., 2023) to test our hypotheses by contrasting an explicit CoT-SFT system (**E**) with an implicit self-distilled system (**I**). (See Section § 2) (1) To evaluate $\mathcal{H}_1$, we construct input-level counterfactual pairs and compare how **E** and **I** respond under an *answer-only* interface; the results show highly similar average sensitivity to counterfactual edits, supporting a weak/conditional form of observational equivalence. (2) To evaluate $\mathcal{H}_2$, we perform mediator-level interventions on the explicit system by swapping or editing CoT while holding the question fixed; we find that these interventions reliably steer the output with high fidelity, exposing a clear *controllability boundary*.

Motivated by this boundary evidence, we ask: *can we recover controllability for implicit CoT without reverting to explicit rationales?* We propose **Controllable Latent Mediation**, a method that *bridges the gap* between explicit controllability and implicit efficiency. The key idea is to keep the student's interface fixed (still answering $Q \to A$), while introducing a *latent mediator* whose content can be modified externally. Concretely, an **Extractor** maps structured intermediate facts (e.g., Editable facts) into a small set of latent embedding; these embedding are injected into the student model at the to influence generation. During training, we align the mediator with teacher-side signals derived from explicit CoT, so that the student internalizes CoT-like reasoning while remaining steerable through latent-space interventions. At test time, editing is performed by changing the extractor's input facts, thereby producing a new mediator that can redirect the student's answer—without exposing any natural-language CoT.

In summary, our contributions are threefold:

- We clarify a **controllability boundary** between explicit and implicit CoT: they are approximately observationally equivalent under input-level counterfactuals, but explicit CoT has a clear advantage under mediator interventions.
- We propose **CLaM**, which restores an intervention handle for implicit reasoning via an extractor-controlled latent mediator while preserving an answer-only interface.
- Experiments show that CLaM enables robust interventions and supports reliable counterfactual propagation, advancing both reasoning performance and controllability.

2 Boundary Hypotheses and Empirical Analysis

This section studies a boundary between *explicit* chain-of-thought (CoT) and *implicit/latent* CoT. Intuitively, an explicit-CoT model exposes a human-readable reasoning trace, while an implicit-CoT model may still perform multi-step reasoning internally but produces only the final answer. A natural question follows: *when do the two paradigms behave similarly, and when do they diverge in a principled way?*

Our empirical analysis is to separate two levels of comparison: (i) **input-level counterfactuals**—we perturb the input question and observe the final answer; (ii) **mediator-level interventions**—we intervene on an intermediate reasoning variable and test whether the output can be predictably controlled.

2.1 Setup: Two Systems

We compare two systems trained in different ways:

- **E (Explicit-CoT; represented by CoT-SFT):** learns from supervised rationales to generate explicit textual rationale and answer for a given question q, i.e., $q \rightarrow (\text{CoT}, a)$.
- **I (Implicit/Latent-CoT; represented by self-distillation (Shen et al., 2025)):** trained to internalize reasoning in latent representations and evaluated as **answer-only**; it does *not* expose an external textual mediator that can be read/written as part of the input-output interface.

Interface constraint For $\mathcal{H}_1$, we evaluate both **E** and **I** in the same answer-only interface, i.e., both are prompted to output only the final answer. For $\mathcal{H}_2$, we consider interventions on a mediator variable. Crucially, **we do not "inject CoT" into I**, because once textual CoT is provided as an executable input, the setting becomes an *explicit* rationale-input interface by definition. Therefore, $\mathcal{H}_2$ is evaluated via mediator interventions that are *well-defined for E* (which exposes a mediator), while we treat the absence of such an externally manipulable mediator in I as the core structural difference.

We construct counterfactual instances from GSM8K-Aug (Deng et al., 2023), an arithmetic dataset on which both **E** and **I** perform well. The construction process and data details are provided in Appendix B. For **input-level counterfactuals ($\mathcal{H}_1$)**, we build paired questions (q, q^{cf}) with gold answers (a, a^{cf}). For **mediator-level interventions ($\mathcal{H}_2$)**, we alter the mediator while keeping the question fixed, (e.g. (q, cot^{cf})) to make the mediator intentionally inconsistent with the input. Examples of q/q^{cf}, a/a^{cf}, and cot/cot^{cf} are shown in Table 6.

2.2 $\mathcal{H}_1$: Input-Level Observational Equivalence

Motivation. Many implicit CoT approaches (Yu et al., 2024; Shen et al., 2025) report strong performance on tasks like arithmetic word problems. This suggests that, when we only perturb the *input*, implicit reasoning might resemble explicit-CoT behavior at the level of observed answers.

$\mathcal{H}_1$ Hypothesis statement. *When evaluated under the same answer-only interface, explicit-CoT (**E**) and implicit-CoT (**I**) exhibit similar behavioral responses under base inputs and input-level counterfactuals.* A practical implication is that if we perturb $q \rightarrow q^{cf}$, both systems should show comparable changes in accuracy and comparable sensitivity patterns.

2.2.1 Experiments for $\mathcal{H}_1$: Input-Level Counterfactual Evaluation

Let $C^{\mathbf{S}}(q_i) \in \{0, 1\}$ denote correctness of system $\mathbf{S} \in \{\mathbf{E}, \mathbf{I}\}$ on input q_i. The three metrics are defined as follows.

Accuracy on base and counterfactual (Acc).

$$\begin{aligned} Acc_q^{\mathbf{S}} &= \frac{1}{N}\sum_{i=1}^{N} C^{\mathbf{S}}(q_i), \\ Acc_{q^{cf}}^{\mathbf{S}} &= \frac{1}{N}\sum_{i=1}^{N} C^{\mathbf{S}}(q_i^{cf}). \end{aligned} \tag{1}$$

Counterfactual joint success (CS).

$$CS^{\mathbf{S}} = \frac{1}{N}\sum_{i=1}^{N} \mathbb{1}\left(C^{\mathbf{S}}(q_i) = 1 \wedge C^{\mathbf{S}}(q_i^{cf}) = 1\right) \tag{2}$$

Average counterfactual shift and Difference-in-Differences (DID). Define per-item shift

$$\Delta_i^{\mathbf{S}} = C^{\mathbf{S}}(q_i^{cf}) - C^{\mathbf{S}}(q_i) \in \{-1, 0, 1\}, \tag{3}$$

so that,

$$\mathbb{E}[\Delta^{\mathbf{S}}] = Acc_{q^{cf}}^{\mathbf{S}} - Acc_q^{\mathbf{S}}. \tag{4}$$

We compare counterfactual sensitivity via difference-in-differences[1]:

$$DID = \mathbb{E}[\Delta^E] - \mathbb{E}[\Delta^I]. \tag{5}$$

Results and Analysis. We validate $\mathcal{H}_1$ on two backbones, LLaMA-3.2-1B-Instruct (Dubey et al., 2024) and GPT-2 (Radford et al., 2019), with results reported in Tables 1 and 2. Empirically, we observe a consistent pattern across both models: due to different training signals, **E** and **I** can differ in baseline accuracy, so the **strong** form of observational equivalence (identical overall behavior) is not supported. However, when viewed through the **weak/conditional** lens of $\mathcal{H}_1$, their *counterfactual sensitivity*—how predictions change from q to q_{cf}—is highly similar: both systems exhibit comparable shifts under input-level counterfactual edits, and the resulting DID remains close to zero.

[1] $DID \approx 0$ indicates **E** and **I** have similar *average* sensitivity to input-level counterfactual perturbations, which supports $\mathcal{H}_1$ in its weak/conditional form.

This suggests that counterfactual perturbations do not introduce an additional systematic divergence beyond overall capability differences, providing evidence for $\mathcal{H}_1$ **in a weak/conditional sense**.

Overall, these results support $\mathcal{H}_1$ **in a weak/conditional sense**: **E** *and* **I** *exhibit nearly indistinguishable average response to input-level counterfactual perturbations (DID* ≈ 0*), even though E retains a modest baseline advantage.*

Method	Input	Acc	CS	DID
E	q	58.00%	51.25%	-0.23%
E	q^{cf}	56.18%		
I	q	53.60%	46.55%	
I	q^{cf}	52.01%		

Table 1: Evaluation results of $\mathcal{H}_1$ pattern on LlaMa3.2-1B-Instruct model.

Method	Input	Acc	CS	DID
E	q	45.87%	39.73%	-0.15%
E	q^{cf}	44.50%		
I	q	41.55%	34.87%	
I	q^{cf}	40.33%		

Table 2: Evaluation results of $\mathcal{H}_1$ pattern on GPT-2 model.

2.3 $\mathcal{H}_2$: Mediator-Level Interventional Non-Equivalence

Motivation. Even if two systems look similar under input perturbations, they may differ in what can be *controlled*. Explicit CoT exposes a mediator M (the CoT) that can be directly edited or swapped. This creates a well-defined notion of a mediator intervention $do(M)$ without accessing internal activations. In contrast, an answer-only implicit system does not expose such an externally manipulable mediator under the same interface constraints.

$\mathcal{H}_2$ **Hypothesis statement.** **E** and **I** diverge under mediator-level counterfactuals: **E** admits well-defined, externally realizable mediator interventions (via explicit CoT), whereas I does not under the same fixed answer-only interface and without internal-state access. We operationalize mediator interventions only for E, where a textual rationale can be provided as a mediator. For each pair (q, q^{cf}) with rationales $(\text{cot}, \text{cot}^{cf})$, we evaluate four conditions:

- **Consistent mediator:** (q, cot) and $(q^{cf}, \text{cot}^{cf})$.
- **Swapped mediator (mediator counterfactual):** (q, cot^{cf}) and (q^{cf}, cot).

2.3.1 Experiments for $\mathcal{H}_2$: Mediator-Level Interventions (Explicit CoT)

Let $a_i^{(M)}$ denote the answer implied by executing the provided mediator M (i.e., the rationale's endpoint). The related two metrics are defined as follows.

Follow-the-mediator rate (FMR).

$$FMR = \frac{1}{N}\sum_{i=1}^{N} \mathbb{1}\left(y_i = a_i^{(M)}\right). \quad (6)$$

High FMR indicates that providing a mediator reliably controls the output.

Mediator causal effect (MCE).

$$MCE = \Pr\ (y = a \mid q, M = \text{cot}) - \Pr\ (y = a \mid q, M = \text{cot}^{cf}). \quad (7)$$

MCE quantifies how the model's output changes when we replace the mediator while keeping the input fixed. A large MCE indicates that mediator replacement induces a strong and consistently reproducible intervention effect.

Results and Analysis. Across backbones, mediator-level interventions on **E** exhibit a clear and consistent pattern (Tables 3 and 4): the model follows the provided rationale with high FMR score, and swapping in a counterfactual mediator reliably induces a large behavioral shift (high MCE), moving predictions away from the original target and toward mediator-consistent outputs even when the swapped mediator is intentionally inconsistent with the question. Together, these results indicate that explicit CoT functions as an externally writable and effective control variable for **E**, strongly supporting $\mathcal{H}_2$: *explicit CoT provides a controllable mediator enabling well-defined interventions*. Under the same external constraints (answer-only interface, no internal-state access), **I** exposes no comparable mediator to intervene on, implying the two paradigms are **interventionally non-equivalent** in terms of intervention realizability.

Method	Input	FMR	MCE
E	q, cot	94.31%	93.71%
E	q, cot^{cf}	96.97%	
E	q^{cf}, cot	97.12%	96.44%
E	q^{cf}, cot^{cf}	94.09%	

Table 3: Evaluation results of $\mathcal{H}_2$ pattern on LlaMa3.2-1B-Instruct model.

Method	Input	FMR	MCE
E	q, cot	93.33%	92.72%
E	q, cot^{cf}	96.59%	
E	q^{cf}, cot	96.36%	95.75%
E	q^{cf}, cot^{cf}	93.33%	

Table 4: Evaluation results of $\mathcal{H}_2$ pattern on gpt2 model.

2.4 Summary: What the Boundary Means

Our empirical evidence suggests a boundary:

- **At the input level ($\mathcal{H}_1$):** **E** and **I** can look similar in how their answer behavior responds to small, controlled input counterfactuals (DID ≈ 0), supporting weak input-level observational equivalence.
- **At the mediator level ($\mathcal{H}_2$):** explicit CoT enables externally realizable mediator interventions (high FMR/MCE), whereas implicit answer-only systems lack an equivalent intervention object under the same interface constraints.

3 Bridging the Gap: Controllable Latent Mediation

The boundary evidence above reveals a clear asymmetry between explicit and implicit reasoning systems. This tension motivates a principled design that preserves the efficiency and usability of implicit reasoning, while recovering the controllability characteristic of explicit-CoT pipelines.

To this end, we propose **CLaM**, a bridge between explicit and implicit reasoning that introduces a controllable latent mediator without exposing textual rationales at inference time. Figure 2 provides an overview of **CLaM**, highlighting its three key stages: (i) a frozen teacher model that reasons with explicit CoT, (ii) a student model that answers directly but is conditioned on a small number of latent mediator tokens, and (iii) an extractor that maps structured intermediate facts or steps into these mediator tokens.

3.1 Problem Formulation

We study controllable reasoning in settings where models are deployed with an *answer-only* interface, yet must support targeted interventions on intermediate knowledge. In particular, we focus on multi-hop question answering and editing scenarios in which the edited knowledge is often local (e.g., a single fact or step) but its effect must propagate through a multi-hop reasoning process.

3.1.1 Controlled Mediation Without Explicit Rationales

Let Q denote a multi-hop question and A its final answer. Let $Z = \{(c_i, a_i)\}_{i=1}^{H}$ be an ordered set of intermediate steps, where each step consists of a (*subject*, *relation*, *object*) single-hop triple fact A teacher model consumes Q together with an explicit chain-of-thought C constructed from Z, and produces A.

In contrast, a student model must predict A given only Q and a compact mediator signal derived from Z, without accessing or emitting explicit rationales. Our objective is **controllable mediation**: modifying Z—for example, editing a single intermediate fact—should induce a predictable change in the student's output distribution, ideally yielding a counterfactual answer A' when such edited facts are available.

3.1.2 Latent Mediator and Intervention Operators

We introduce a latent mediator represented as a continuous latent embedding $M \in \mathbb{R}^{K \times d}$, where K is small and d matches the hidden dimension of the student. An extractor E_ϕ maps intermediate steps into the mediator:

$$M = E_\phi(Z). \tag{8}$$

The student model $S_{\theta,\Delta}$—with backbone parameters θ and optional lightweight adapters Δ—then predicts:

$$P_{S_{\theta,\Delta}}(A \mid Q, M). \tag{9}$$

We formalize interventions as operators applied to the intermediate step set Z, including: (i) *Edit*, where a step (s_i, r_i, o_i) is modified to (s_i, r_i, o_i'); and (ii) *Compose*, where multiple steps are combined to induce multi-hop propagation. An intervention is realized by recomputing the mediator $M' = E_\phi(Z')$ and querying the student with (Q, M').

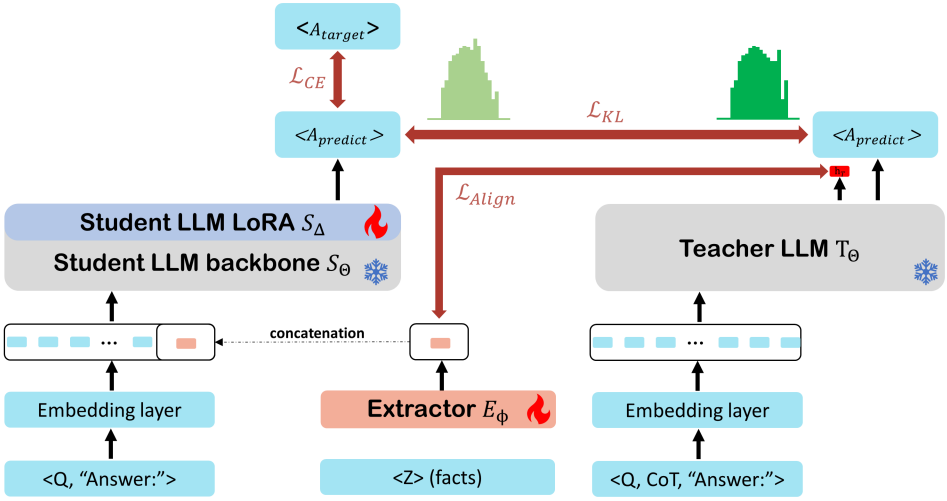

Figure 2: An overview of the proposed **CLaM** framework: 1) **Teacher-guided mediation.** CLaM starts from a frozen teacher model that solves the input question using explicit CoT, providing a structured reasoning signal that serves as supervision for controllable mediation. 2) **Answer-only student with latent mediators.** A student model is trained to answer directly under an **answer-only** interface, but its generation is conditioned on a small number of learned latent mediator tokens, allowing reasoning-relevant information to influence outputs without exposing textual rationales at inference time. 3) **Extractor-based mediator synthesis.** To make the mediator *editable* and *recomputable*, an extractor maps structured intermediate facts/steps into these mediator tokens, so interventions can be implemented by modifying the intermediate facts and re-deriving the mediator used by the student.

3.2 Model Components and Architecture

Our method consists of three components. For more architectural details, please refer to Appendix C.

- **Teacher T_Θ.** A (frozen) autoregressive LM conditioned on explicit CoT:

$$P_{T_\Theta}(A \mid Q, C). \tag{10}$$

- **Extractor E_ϕ.** A smaller LM model that encodes Z into K intermediary latent embedding $M \in \mathbb{R}^{K \times d}$.

- **Student $S_{\theta,\Delta}$.** A backbone same as teacher model with optional LoRA (Hu et al., 2021) adapters Δ, predicting A conditioned on Q and M.

3.2.1 Training Data and Supervision Signals

Each training instance provides (Q, Z, A), and optionally a counterfactual edited version (Q, Z', A'). The teacher's explicit CoT C is derived from Z. The student never observes C directly; it only consumes Q and latent embedding M.

We leverage three supervision signals:

1. **Answer supervision** for the student $(Q, M \rightarrow A)$.
2. **Distributional distillation** from the teacher (teacher answer-token distribution).
3. **Representation alignment** between the mediator and a teacher internal state.

3.2.2 Objective Functions and Optimization

Let y denote the answer token sequence for A. Let $\mathcal{A}$ index the positions of answer tokens (e.g., tokens after "Answer:").

(i) Student answer supervision (Cross-entropy loss). We apply causal LM cross-entropy only on answer tokens:

$$\begin{aligned} \mathcal{L}_{\text{CE}} &= -\sum_{t \in \mathcal{A}} \log p_S(y_t \mid y_{<t}, Q, M) \\ M &= E_\phi(Z). \end{aligned} \tag{11}$$

(ii) Teacher-to-student distillation (KL divergence loss (Ouyang et al., 2022)). We match the student's answer-token distribution to the teacher's using temperature τ:

$$\mathcal{L}_{\text{KL}} = \sum_{t \in \mathcal{A}} \text{KL}\Big(\text{s}(z_T^{(t)}/\tau) \parallel \text{s}(z_S^{(t)}/\tau)\Big) \cdot \tau^2 \tag{12}$$

where $\text{s}(\cdot)$ is softmax fuction and $z_T^{(t)}$ and $z_S^{(t)}$ are teacher/student logits at answer-aligned positions. In our setting the teacher is frozen; KL remains effective as a fixed target distribution.

(iii) Mediator–teacher state alignment. We extract a teacher hidden representation h_T at a boundary tied to answer onset (e.g., the hidden state immediately before the first answer token). For $K > 1$, we mean-pool the mediator tokens $\bar{M} = \frac{1}{K}\sum_{k=1}^{K} M_k$:

$$\mathcal{L}_{\text{Align}} = \|\bar{M} - h_T\|_2^2. \quad (13)$$

This encourages the extractor to produce a compact mediator that approximates a teacher state predictive of the answer under explicit CoT.

Overall objective. The final training objective is:

$$\mathcal{L} = \lambda_{\text{CE}}\mathcal{L}_{\text{CE}} + \lambda_{\text{KL}}\mathcal{L}_{\text{KL}} + \lambda_{\text{Align}}\mathcal{L}_{\text{Align}} \quad (14)$$

We update the extractor parameters E_ϕ and student adapter parameters Δ while keeping the student backbone θ frozen. This preserves the backbone's general capability and substantially reduces the computational cost of training.

3.3 Results Analysis and Ablations

In this section, we present the key results of our experiments. The experimental setup is provided in Appendix A.2.

3.3.1 Main Results and Analysis

We evaluate the effectiveness of our method using GPT2-large as the extractor and GPT-J as the student backbone, with results summarized in Table 5. Across both benchmarks, CLaM consistently outperforms prior editing baselines, demonstrating substantially stronger editing accuracy and generalization. In particular, while existing methods exhibit limited robustness under counterfactual and multi-hop editing settings, CLaM achieves reliable propagation from edited intermediate facts to final answers. This performance gap suggests that controlling reasoning through an extractor-driven latent mediator enables more effective and stable interventions than directly modifying model parameters.

3.3.2 Ablation of Latent Number

To assess the effect of latent capacity, we use GPT2-large as the extractor and GPT-J (Wang and Komatsuzaki, 2021) as the backbone, and evaluate four settings with different numbers of latent tokens ($K = 1, 2, 4, 8$) on MQuAKE-CF-v2, as shown in figure 3. The results show that increasing K yields consistent performance improvements, suggesting that a larger mediator provides more bandwidth to encode and transmit reasoning-relevant intermediate information needed for multi-hop editing. Meanwhile, the parameter overhead grows only moderately, making this a relatively efficient way to strengthen controllability. Gains are largest when moving from small to moderate K and then begin to saturate, indicating diminishing returns beyond a certain mediator capacity.

	MQuAKE-CF-v2	MQuAKE-T
	E_ϕ: *GPT2-large*, S: *GPT-J*	
MEMIT	12.3%	4.8%
MEND	11.5%	38.2%
MeLLo	20.3%	85.9%
CLaM	68.3%	89.2%

Table 5: Main results on MQuAKE-CF-v2 and MQuAKE-T using GPT2-large as the extractor and GPT-J as the student backbone.

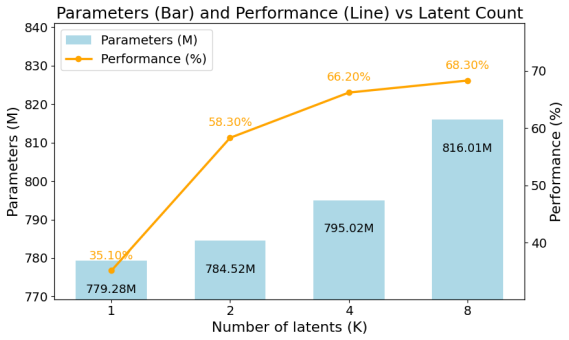

Figure 3: Effect of latent mediator capacity on multi-hop editing performance on MQuAKE-CF-v2, using GPT2-large as the extractor and GPT-J as the backbone, evaluated across four settings with different numbers of latent tokens.

3.3.3 Ablation at the Extractor Scale

We ablate the extractor backbone by using GPT2, GPT2-Medium, and GPT2-Large while keeping the rest of the setup fixed (Figure 4). Performance improves consistently as the extractor scales, indicating that a stronger extractor produces higher-quality latent mediators that better encode editable intermediate information and support reliable propagation to final answers. This suggests extractor capacity is a key driver of CLaM's controllability and effectiveness.

4 Related Work

Efficient and Concise Reasoning While explicit CoT reasoning significantly enhances model per-

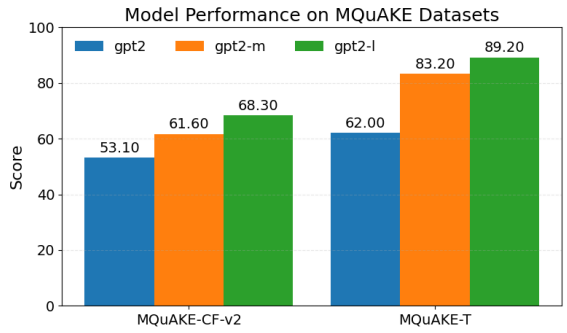

Figure 4: Ablation study on extractor capacity.

formance on complex tasks, its verbose generation process imposes substantial inference costs and latency. Addressing this efficiency bottleneck, recent research has focused on "Concise Thinking." For instance, Renze and Guven (2024) proposed Concise CoT, which utilizes prompt engineering to guide models toward generating shorter explanations without compromising accuracy. Similarly, Nayab et al. (2024) introduced Constraint CoT, effectively compressing reasoning paths by imposing length constraints. Other approaches like TokenComplexity (Lee et al., 2025) and NoThinking (Ma et al., 2025) further explore the minimal token requirements for valid reasoning. Furthermore, recent works have explored "Adaptive Thinking" strategies, such as CoThink (Fan et al., 2025) and ThinkSwitcher (Liang et al., 2025), which dynamically switch between fast, direct answering and slow, detailed reasoning based on problem difficulty. Although these methods reduce the number of explicit tokens, they largely remain dependent on textual intermediate steps and do not fully achieve the internalization of the reasoning process.

Implicit and Latent Reasoning To pursue extreme efficiency, another line of research aims to distill explicit reasoning steps into the model's internal hidden states, known as Implicit CoT. Yu et al. (2024) formalized the process of distilling "System 2" (slow thinking) capabilities into "System 1" (fast thinking), enabling models to skip intermediate steps and directly output the correct answer. Shen et al. (2025) proposed CODI, a self-distillation framework that allows a model to act as its own "teacher" by aligning critical hidden states across explicit and implicit reasoning paths. While these approaches successfully "black-box" and accelerate the reasoning process, they often do so at the expense of interpretability and controllability.

Controllability and Mechanism Intervention While users can steer explicit reasoning by editing CoT text, restoring such controllability in implicit models presents a novel challenge. Recent studies have begun to investigate understanding and controlling the internal reasoning processes of large models. Eisenstadt et al. (2025) discovered that large models are aware of their relative position in a thought process and demonstrated control over decoding using a "thinking progress vector." Sheng et al. (2025) proposed that models can pre-plan reasoning intensity, allowing for the prediction and control of reasoning token counts via linear models. (Panickssery et al.) demonstrated the feasibility of manually controlling the reasoning process through Activation Steering. Other interventions include Flash Think (Jiang et al., 2025) and DEER (Yang et al., 2025b), which manipulate decoding to accelerate inference. However, most of these works focus on controlling the "length" or "progress" of reasoning rather than the specific details of the logical content. Our proposed CLaM framework bridges this gap by introducing a Controllable Latent Mediator, which restores the ability to perform specific, fact-level interventions on reasoning logic while maintaining the efficiency of implicit inference.

5 Conclusion

We studied the boundary between explicit and implicit Chain-of-Thought (CoT) systems. While explicit CoT exposes a natural-language reasoning channel that can be edited at inference time to steer outputs, implicit CoT internalizes the reasoning trace into parameters, improving efficiency but making post-hoc control difficult. This tension is especially consequential in editing and counterfactual settings, where one often needs to intervene on a specific intermediate fact while keeping the rest of the model's behavior stable.

To bridge this gap, we proposed **Controllable Latent Mediation (CLaM)**, which keeps an answer-only student interface while introducing a controllable latent mediator generated by an *Extractor* from structured facts or intermediate steps. This design enables steering and editing through latent interventions without requiring the model to output explicit rationales, offering a practical path toward reasoning models that are both efficient and controllable.

Limitations

Our study has several limitations. First, our training objective may introduce dependence on the teacher and inherit its biases: because the student is optimized via distillation and representation alignment, its behavior can mirror systematic errors from the teacher. If the teacher's CoT or intermediate steps are unreliable, CLaM may learn to *faithfully reproduce mistakes under intervention*, rather than capturing a more causally grounded reasoning process. Second, interpretability remains limited. Although CLaM restores *intervenability*, the mediator is a continuous latent representation and is not directly human-readable; compared to explicit CoT, it is harder to diagnose why a particular intervention succeeds or fails, potentially shifting debugging effort to the construction of Z and the behavior of the extractor.

References

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

Roi Cohen, Eden Biran, Ori Yoran, Amir Globerson, and Mor Geva. 2024. Evaluating the ripple effects of knowledge editing in language models. *Transactions of the Association for Computational Linguistics*, 12:283–298.

Yuntian Deng, Kiran Prasad, Roland Fernandez, Paul Smolensky, Vishrav Chaudhary, and Stuart Shieber. 2023. Implicit chain of thought reasoning via knowledge distillation. *arXiv preprint arXiv:2311.01460*.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, and 1 others. 2024. The llama 3 herd of models. *CoRR*.

Roy Eisenstadt, Itamar Zimerman, and Lior Wolf. 2025. Overclocking llm reasoning: Monitoring and controlling thinking path lengths in llms. *arXiv preprint arXiv:2506.07240*.

Siqi Fan, Bowen Qin, Peng Han, Shuo Shang, Yequan Wang, and Aixin Sun. 2025. The price of a second thought: On the evaluation of reasoning efficiency in large language models.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, and 1 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

Guochao Jiang, Guofeng Quan, Zepeng Ding, Ziqin Luo, Dixuan Wang, and Zheng Hu. 2025. Flashthink: An early exit method for efficient reasoning. *arXiv preprint arXiv:2505.13949*.

Ayeong Lee, Ethan Che, and Tianyi Peng. 2025. How well do llms compress their own chain-of-thought? a token complexity approach. *arXiv preprint arXiv:2503.01141*.

Guosheng Liang, Longguang Zhong, Ziyi Yang, and Xiaojun Quan. 2025. Thinkswitcher: When to think hard, when to think fast. *arXiv preprint arXiv:2505.14183*.

Wenjie Ma, Jingxuan He, Charlie Snell, Tyler Griggs, Sewon Min, and Matei Zaharia. 2025. Reasoning models can be effective without thinking. *arXiv preprint arXiv:2504.09858*.

Eric Mitchell, Charles Lin, Antoine Bosselut, Chelsea Finn, and Christopher D Manning. 2021. Fast model editing at scale. *arXiv preprint arXiv:2110.11309*.

Sania Nayab, Giulio Rossolini, Marco Simoni, Andrea Saracino, Giorgio Buttazzo, Nicolamaria Manes, and Fabrizio Giacomelli. 2024. Concise thoughts: Impact of output length on llm reasoning and cost. *arXiv preprint arXiv:2407.19825*.

OpenAI. 2025. Introducing gpt-5.2. Https://openai.com/index/introducing-gpt-5-2.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, and 1 others. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.

Nina Panickssery, Nick Gabrieli, Julian Schulz, Meg Tong, Evan Hubinger, and Alexander Matt Turner. Steering llama 2 via contrastive activation addition, 2024. *URL https://arxiv. org/abs/2312.06681*, 3.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, and 1 others. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Matthew Renze and Erhan Guven. 2024. 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)*, pages 476–483.

Zhenyi Shen, Hanqi Yan, Linhai Zhang, Zhanghao Hu, Yali Du, and Yulan He. 2025. CODI: Compressing chain-of-thought into continuous space via self-distillation. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 677–693, Suzhou, China. Association for Computational Linguistics.

Leheng Sheng, An Zhang, Zijian Wu, Weixiang Zhao, Changshuo Shen, Yi Zhang, Xiang Wang, and Tat-Seng Chua. 2025. On reasoning strength planning in large reasoning models. *arXiv preprint arXiv:2506.08390*.

Chenmien Tan, Ge Zhang, and Jie Fu. 2023. Massive editing for large language models via meta learning. *arXiv preprint arXiv:2311.04661*.

Wenhui Tan, Jiaze Li, Jianzhong Ju, Zhenbo Luo, Jian Luan, and Ruihua Song. 2025. Think silently, think fast: Dynamic latent compression of llm reasoning chains. *arXiv preprint arXiv:2505.16552*.

Ben Wang and Aran Komatsuzaki. 2021. Gpt-j-6b: A 6 billion parameter autoregressive language model.

Yige Xu, Xu Guo, Zhiwei Zeng, and Chunyan Miao. 2025. Softcot: Soft chain-of-thought for efficient reasoning with llms. *arXiv preprint arXiv:2502.12134*.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025a. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Chenxu Yang, Qingyi Si, Yongjie Duan, Zheliang Zhu, Chenyu Zhu, Qiaowei Li, Minghui Chen, Zheng Lin, and Weiping Wang. 2025b. Dynamic early exit in reasoning models. *arXiv preprint arXiv:2504.15895*.

Yunzhi Yao, Peng Wang, Bozhong Tian, Siyuan Cheng, Zhoubo Li, Shumin Deng, Huajun Chen, and Ningyu Zhang. 2023. Editing large language models: Problems, methods, and opportunities. *arXiv preprint arXiv:2305.13172*.

Ping Yu, Jing Xu, Jason Weston, and Ilia Kulikov. 2024. Distilling system 2 into system 1. *arXiv preprint arXiv:2407.06023*.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, and 1 others. 2025. Dapo: An open-source llm reinforcement learning system at scale. *arXiv preprint arXiv:2503.14476*.

Ningyu Zhang, Yunzhi Yao, Bozhong Tian, Peng Wang, Shumin Deng, Mengru Wang, Zekun Xi, Shengyu Mao, Jintian Zhang, Yuansheng Ni, and 1 others. 2024. A comprehensive study of knowledge editing for large language models. *arXiv preprint arXiv:2401.01286*.

Zexuan Zhong, Zhengxuan Wu, Christopher D Manning, Christopher Potts, and Danqi Chen. 2023. Mquake: Assessing knowledge editing in language models via multi-hop questions. *arXiv preprint arXiv:2305.14795*.

A Experiments Details

In this section, we describe some of the setups used in the experiment in detail.

A.1 Implementation Details of $\mathcal{H}_1$ and $\mathcal{H}_2$

For the experiments on $\mathcal{H}_1$ and $\mathcal{H}_2$, we trained the implicit-CoT system (following the CODI framework) by strictly adhering to the setup of Shen et al. (2025) et al. We finetuned all models with LoRA (Hu et al., 2021) using rank $r = 128$ and scaling $\alpha = 32$, and trained in bfloat16 precision. For GPT-2, we used a learning rate of 3×10^{-3} and trained for 40 epochs. For LLaMA-3.2-1B, we used a learning rate of 8×10^{-4} and trained for 10 epochs. We applied a cosine learning-rate schedule with a 3% warmup ratio, and optimized with `AdamW`. Using the same hyperparameter settings, we also trained an explicit-CoT baseline (CoT-SFT) with LoRA.

A.2 Implementation Details of CLaM

Dataset We evaluate on the MQuAKE benchmark (Zhong et al., 2023), including its counterfactual split **MQuAKE-CF** and the time-synchronized split **MQuAKE-T**.

Experimental Model We use GPT2-large (Radford et al., 2019) as the Extractor, and adopt GPT-J (Wang and Komatsuzaki, 2021) as the backbone for both the teacher and the answer-only student.

Methods We compare CLaM against representative knowledge editing baselines, including MEMIT (Tan et al., 2023), MEND (Mitchell et al., 2021), and MeLLo (Zhong et al., 2023).

Experimental Setting We train the extractor parameters ϕ and the student adapters Δ while keeping the student backbone θ frozen. We train for 1 epoch with batch size 1 and gradient accumulation 8 (effective batch size 8), using AdamW with an initial learning rate of 1×10^{-4}, warmup ratio 0.03, no weight decay, and a cosine learning-rate schedule with warmup. The overall objective combines $\mathcal{L}_{CE}, \mathcal{L}_{KL}, \mathcal{L}_{Align}$, with loss weights λ_{CE}=1.0, λ_{KL}=1.0, and λ_{align}=1.0; the distillation temperature is set to τ=2.0. The default latent

mediator length is K=8 tokens. For parameter-efficient tuning, we apply LoRA to the student with rank r=16, α=32, dropout=0.05, and use mixed bfloat16 precision.

B GSM8K-Aug Counterfactual Data Construction

We construct a counterfactual version of GSM8K-Aug (Deng et al., 2023) by minimally perturbing numerical conditions while preserving the original problem structure. Each original instance provides a question q, a gold answer a, and an explicit chain-of-thought CoT from the dataset. These components are kept unchanged.

For each instance, we use a large language model GPT-5.2 (OpenAI, 2025) as a controlled data generator to produce a counterfactual variant $(q^{cf}, a^{cf}, cot^{cf})$ under strict constraints. Specifically, qcf is obtained by modifying exactly one numerical quantity or fraction in q, while keeping all other textual content identical. The counterfactual answer a^{cf} is then recomputed accordingly.

To ensure structural alignment at the mediator level, the generated cot^{cf} is required to consist solely of symbolic calculation chunks in the form «·», with no natural language explanations or additional tokens. This restriction enforces a direct correspondence between numerical changes in the question and their induced computational effects in the reasoning trace.

The generation process is guided by few-shot examples and validated using a strict JSON schema to guarantee well-formed outputs and consistency across fields. Instances that violate any constraint (e.g., multiple calculation, malformed calculation blocks) are recreated. As a result, each retained counterfactual pair differs from the original instance only in numerical condition, yielding aligned tuples (q, a, cot) and $(q^{cf}, a^{cf}, cot^{cf})$ suitable for input-level and mediator-level counterfactual analysis. An example is shown in table 6.

C Structural Details

In this section, we describe some of the model structural details.

C.1 Teacher and Student

In CLaM, the teacher and student share the same backbone architecture. The teacher model is fully frozen and operates with explicit CoT to provide reliable supervision signals. The student model adopts the same backbone but is trained under an answer-only interface; its backbone parameters are kept frozen, and only lightweight LoRA adapters are introduced and optimized. This design preserves the general reasoning capability of the backbone while enabling parameter-efficient adaptation driven by latent mediation.

C.2 Extractor

The extractor E_ϕ is implemented as a lightweight language model augmented with a projection head to produce a fixed-length latent mediator. Concretely, we instantiate the extractor using a GPT-2 encoder, which maps the input sequence of structured intermediate steps Z into contextual token representations. Let $H \in \mathbb{R}^{L \times d_{\text{enc}}}$ denote the last-layer hidden states of the extractor for an input sequence of length L, where d_{enc} is the hidden size of the GPT-2 encoder.

To obtain a sequence-level representation, we apply a simple pooling strategy by selecting the hidden state corresponding to the last non-padding token:

$$h_{\text{pool}} = H_{i^*}, \quad i^* = \max\{i \mid \text{mask}_i = 1\}. \quad (15)$$

This pooled representation $h_{\text{pool}} \in \mathbb{R}^{d_{\text{enc}}}$ summarizes the intermediate facts or steps provided to the extractor.

To produce a latent mediator of length K, we apply a linear projection that expands the pooled representation into K embeddings:

$$\tilde{M} = W h_{\text{pool}} + b, \quad \tilde{M} \in \mathbb{R}^{K \cdot d}, \quad (16)$$

where d matches the hidden size of the student model. The projected vector is then reshaped into a sequence of mediator tokens

$$M = \text{reshape}(\tilde{M}) \in \mathbb{R}^{K \times d}. \quad (17)$$

The resulting mediator M serves as a compact latent representation of the intermediate reasoning content and is injected into the student model to condition answer generation. By constraining K to be small, the extractor is encouraged to encode reasoning-relevant information rather than acting as a high-capacity passthrough.

Original Instance	**Counterfactual Instance**
Question (q) Charlie wants to sell beeswax candles. For every pound of beeswax, he can make 10 tapered candles. One pound of beeswax and the wicks cost $10.00 in supplies. If he sells each candle for $2.00 each, what is his net profit if he makes and sells 20 candles?	**Question** (q_{cf}) Charlie wants to sell beeswax candles. For every pound of beeswax, he can make 10 tapered candles. One pound of beeswax and the wicks cost $10.00 in supplies. If he sells each candle for $3.00 each, what is his net profit if he makes and sells 20 candles?
CoT (cot) «20/10=2» «10*2=20.00» «2*20=40.00» «40-20=20.00»	**Counterfactual CoT** (cot_{cf}) «20/10=2» «10*2=20.00» «3*20=60.00» «60-20=40.00»
Answer (a) 20	**Answer** (a_{cf}) 40

Table 6: An example of counterfactual construction on GSM8K-Aug. The counterfactual instance is obtained by modifying a single numerical condition in the question (selling price), which induces consistent changes in both the answer and the symbolic reasoning trace.