南京大学
NANJING UNIVERSITY

NeurIPS 2026

Understanding the Out-of-Distribution Generalization of Chain-of-Thought Reasoning in LLMs

Nuo-Jing Liang[1], Xiao-Tong Yuan[1], and Pan Zhou[2]

[1] School of Intelligence Science and Technology, Nanjing University
[2] Singapore Management University

Chain-of-Thought Reasoning

In chain-of-thought (CoT) reasoning, a language model generates intermediate steps before giving its answer.

Early methods used worked examples or a prompt to reason step by step.

Later work extended CoT through task decomposition, training on reasoning traces, and search over candidate steps.

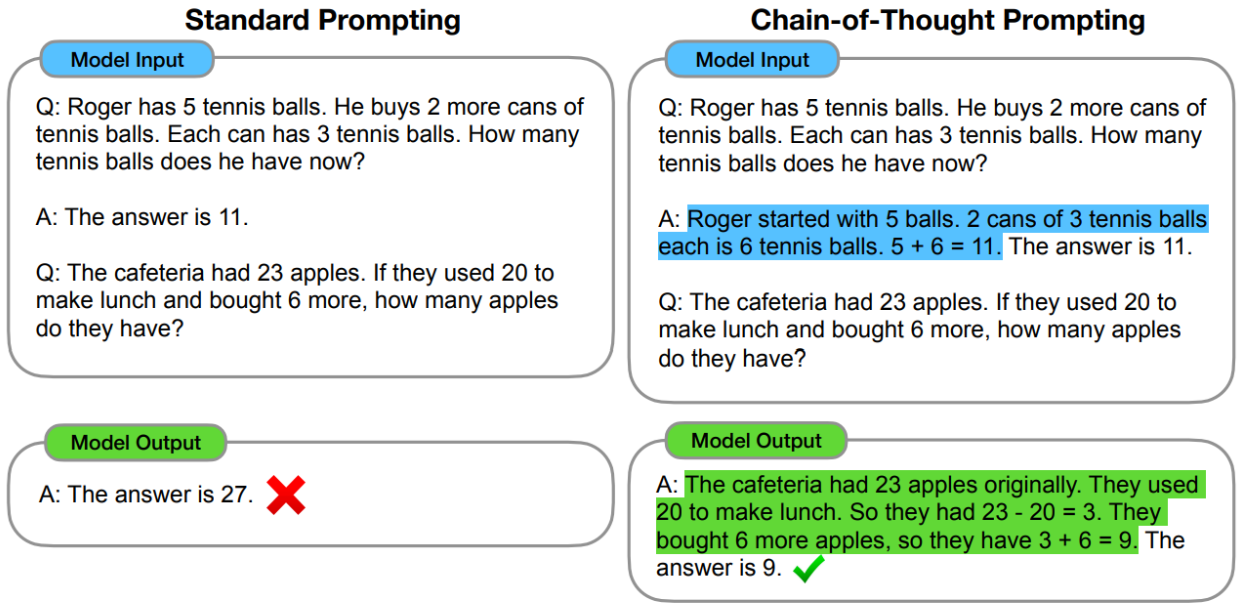
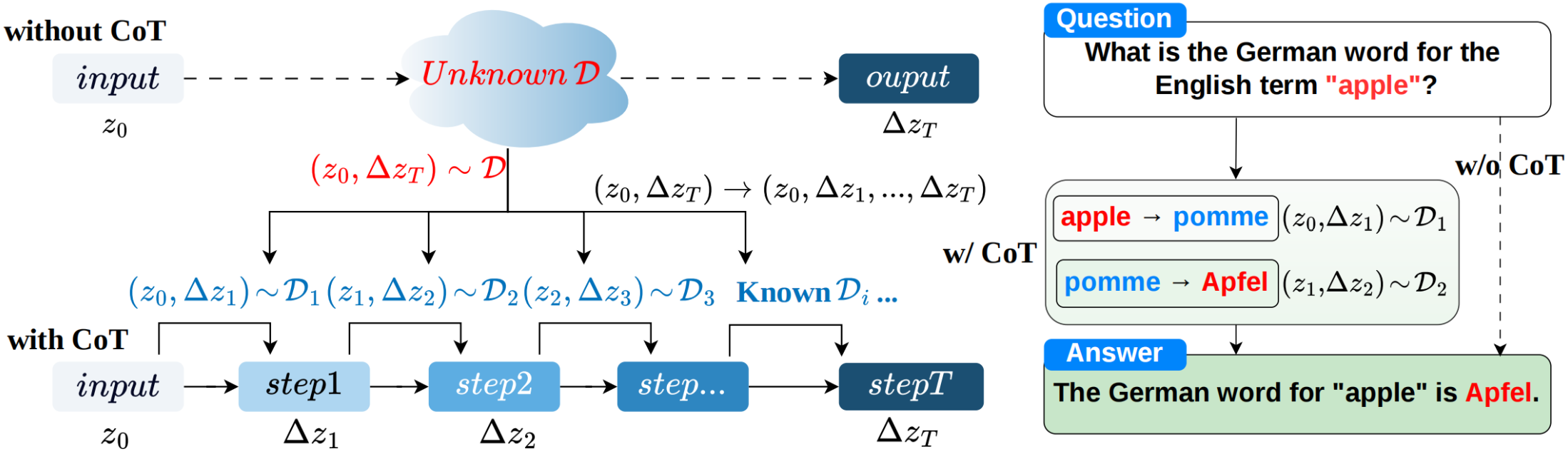


Figure 1: Chain-of-thought prompting enables large language models to tackle complex arithmetic, commonsense, and symbolic reasoning tasks. Chain-of-thought reasoning processes are highlighted.

Wei et al., NeurIPS 2022

Compositional OOD Generalization

Can a model trained on atomic steps solve unseen multi-step tasks?



A shared predictor updates the state at each step. The full input–answer mapping can be out of distribution even when every step follows a training distribution.

Main Theoretical Results

Generalization error decomposition

Theorem 1 bounds final-task risk by the sum of oracle step risks.

$$R_{\mathcal{D}}(F_\psi) \leq \frac{B_\ell k}{\log 2} \sum_{i=1}^{T} \int \mathfrak{L}(\psi(z_{i-1}), \Delta z_i) d\mathcal{D}_i(z_{i-1}, \Delta z_i),$$

Rademacher complexity with one shared predictor

At fixed per-step sample size n, pooling reduces the estimation rate from T/√n to √(T/n) under uniform capacity control.

Autoregressive next-token prediction

$$R_{\mathcal{D}}(F_{\psi_{\widehat{g},\widehat{h}}}) \leq \frac{B_\ell k}{\log 2} \sum_{i=1}^{T} R_i^k(\psi^\star) + \widetilde{\mathcal{O}}\left(k\sqrt{T/n}\right).$$

The estimation rate holds for fixed decoder dimensions, norm bounds, and clipping.
T is reasoning depth, n is the per-step sample size, and k is the number of output tokens per step.

Atom-Task Experimental Setup

We fine-tune Qwen3-4B with LoRA on chains using Reverse, Substitute, Duplicate, Deduplicate, and Rotate.

Short-to-long generalization

Train on single-step or short-chain data. Evaluate on held-out chains of 3, 5, 7, and 9 steps.

Unseen combinations

Each training split excludes one operation from multi-step chains. Single-step data still includes all five operations.

Example CoT output

```
Input Sequence: [αβγ]
Task Chain: [Duplicate → Reverse → Substitute]

Step 1: [αβγ] → [ααββγγ]
Step 2: [ααββγγ] → [γγββαα]
Step 3: [γγββαα] → [δδααββ]

Final Answer: [δδααββ]
```

CoT predicts intermediate steps and the final answer. No-CoT predicts the final answer only.

Atom-Task Results

Short-to-long accuracy (%) averaged over three seeds

Training setting	3-step test	5-step test	7-step test	9-step test
Single-step CoT	91.80	63.13	36.27	22.27
Three-step CoT	99.33	95.73	88.87	79.27
Mixed 3+5 CoT	100.00	97.87	94.00	**90.40**
Mixed 3+5 No-CoT	51.00	26.60	12.20	5.73

Unseen combinations. Across the five operation exclusions, 9-step CoT accuracy ranges from 66.73% to 87.13%.

State tracking. Removing the input and remaining-task records lowers mean 7-step accuracy from 85.37% to 67.56%.