C²DLM: Causal Concept-Guided Diffusion Large Language Models

Kairong Han[1], Nuanqiao Shan[1], Ziyu Zhao[1], Zijing Hu[1], Xinpeng Dong[1], Junjian Ye[2], Lujia Pan[2], Fei Wu[1,3], Kun Kuang[1]†
[1]College of Computer Science and Technology, Zhejiang University,
[2]Noah's Ark Lab, Huawei Technologies,
[3]Shanghai AI Laboratory,

Abstract

Autoregressive (AR) language models and Diffusion Language Models (DLMs) constitute the two principal paradigms of large language models. However, both paradigms suffer from insufficient reasoning capabilities. Human reasoning inherently relies on causal knowledge and thought, which are reflected in natural language. But in the AR paradigm, language is modeled as next token prediction (a strictly left-to-right, token-by-token order), whereas natural language itself exhibits more flexible causal structures. In the DLM paradigm, the attention mechanism is fully connected, which entirely disregards causal order. To fill this gap, we propose the **C**ausal **C**oncept-Guided **D**iffusion **L**anguage **M**odel (C²DLM). Starting from DLM's fully connected attention, C²DLM first obtains a concept-level causal graph from the teacher model, and then explicitly guides attention to learn causal relationships between concepts. By focusing on causal relationships and avoiding interference from difficult subgoals involving causal inversion, C²DLM achieves a 12% improvement and a 3.2× training speedup on the COT-OrderPerturb task, along with an average gain of 1.31% across six downstream reasoning tasks. Code and data are available here.

1 Introduction

In recent years, the development of large language models (LLMs) (Zhao et al., 2023; Liu et al., 2024; Team et al., 2023) has led to two dominant paradigms: autoregressive (AR) LLMs and diffusion language models (DLMs) (Li et al., 2025a; Nie et al., 2025). In the AR paradigm, a causal mask (Vaswani et al., 2017) constrains the model with a lower-triangular matrix to predict the next token based on preceding tokens. In contrast, the DLM paradigm employs fully connected attention to guide the model in globally modeling data from coarse to fine (Ye et al., 2024).

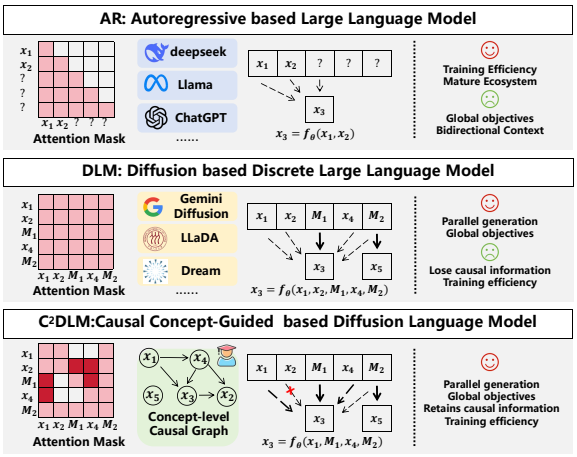

Figure 1: Difference between AR, DLM, and C²DLM. AR models struggle to capture global information, and linguistic flexibility is not bound to a strict left-to-right, token-by-token causal order. DLMs discard causal priors entirely. The C²DLM explicitly guides the model to learn causal relations between concepts, capturing the underlying causal priors of natural language generation.

However, both AR and DLMs suffer from insufficient reasoning capabilities, such as frequent hallucinations (Huang et al., 2025) and unreliable reasoning chains (Lanham et al., 2023; Yehudai et al., 2025), imposing fundamental limitations on tasks that require reasoning and posing significant challenges in real-world deployment (Pan et al., 2025; Acharya et al., 2025; Wu et al., 2024a). Specifically, AR models exhibit limitations in complex reasoning, long-term planning, and maintaining global coherence (Ye et al., 2024; Bubeck et al., 2023; Kambhampati et al., 2024; Zečević et al., 2023). DLMs, as strong competitors to AR models, reduce training efficiency and hinder the effective scaling of reasoning depth, thereby limiting their potential for complex tasks.

Human reasoning inherently relies on causal knowledge and thought. From a natural lan-

† Corresponding author.

Findings of the Association for Computational Linguistics: ACL 2026, pages 821–838
July 2-7, 2026 ©2026 Association for Computational Linguistics

guage perspective, it is inherently flexible rather than strictly left-to-right and token-by-token causal structures. However, AR generation enforces unidirectional information flow, resulting in local greediness and a limited understanding of global objectives and long-term structure. In contrast, DLMs discard the causal order between tokens, often producing final answers before the intermediate reasoning steps that COT methods would generate (Wang et al., 2025a). Their training further involves numerous difficult sub-tasks (e.g., predicting masked cause variables from outcomes under random masking) (Kim et al., 2025), which reduces training efficiency and constrains the scalable development of reasoning depth.

To address the above fundamental problems, we hypothesize that *these limitations stem from a misalignment between the attention mechanism's modeling priors of natural language and the causal priors underlying natural language*. Therefore, we aim to guide the model to capture the underlying causal priors of the natural language generation process, rather than superficial correlations. Inspired by this, we propose the **C**ausal **C**oncept-Guided **D**iffusion **L**anguage **M**odel (C^{2}DLM) paradigm, as shown in Figure 1.

The C^{2}DLM extends DLMs by two key steps: (1) concept-level causal meta-knowledge extraction, and (2) causal alignment via the *V-aware Reattention* mechanism. In the first step, to obtain concept-level causal graphs at low cost, an automated workflow leverages the in-context learning (ICL) (Dong et al., 2022) capabilities of teacher LLMs to extract concept-level meta-knowledge. In the second step, we propose the V-aware Reattention mechanism to align the attention map weighted by the L2-norm of the value matrix with the underlying causal priors extracted in step one for the natural language generation process.

To compare AR, DLM, and C^{2}DLM systematically, we design the COT-OrderPerturb dataset to quantify the impact of priors. AR models are sensitive to concept order, whereas DLMs are more robust but limited by efficiency and performance bottlenecks. Building on DLMs, C^{2}DLM achieves a 12% higher performance and 3.2× faster training. On downstream tasks with explicit causal priors, C^{2}DLM yields average improvements of 7.43% on STG (Han et al., 2025b) and 10.84% on Sudoku (training set size 200). Across six reasoning-related datasets, it delivers an average gain of 1.31%, while causal prior extraction with GLM-4.5 costs only $0.46 per million tokens. Our contributions can be summarized as follows:

- We propose C^{2}DLM, a new paradigm distinct from AR and DLM. It enhances reasoning ability by guiding attention through causal knowledge between concepts to achieve causal alignment.
- The C^{2}DLM achieves a 12% improvement and a 3.2× acceleration of training efficiency in the COT-OrderPerturb tasks, 7.43% on the STG dataset, and 1.31% across six reasoning-related downstream datasets on average.
- We reveal the risk of misalignment between attention mechanisms and the causal priors underlying natural language, which shows the potential of combining causality into language models.

2 Preliminaries and Related Work

2.1 Diffusion Large Language Model

Recently, researchers have adapted the diffusion paradigm (Yang et al., 2023; Cao et al., 2024; Tong et al., 2025) to discrete text data, proposing DLMs (Nie et al., 2025; Ye et al., 2025, 2024; Austin et al., 2021), which achieve competitive performance compared to AR models. DLMs employ a bidirectional attention mechanism and leverage the Negative Evidence Lower Bound to provide an upper bound on the negative log-likelihood of the training data, thereby modeling the distribution of language. LLaDA (Nie et al., 2025) first demonstrated the effectiveness of DLMs at the 8B scale, using the following supervised fine-tuning (SFT) loss $\mathcal{L}_{\mathcal{DLM}}$:

$$-\mathbb{E}_{t,p_0,r_0,r_t}\left[\sum_{i=1}^{L'}\mathbf{1}[r_i^t = M]\cdot\log p_\theta(r_i^0 \mid p_0, r_t)\right],$$

where r_t denotes the noised sequence appended to the prompt p_0. Recent work has mainly focused on improving DLM by reinforcement learning (RL) (Kaelbling et al., 1996; Hu et al., 2026b), such as d1 (Zhao et al., 2025a), wd1 (Tang et al., 2025), BranchGRPO (Li et al., 2025b), and others (Zhao et al., 2025b; Zhu et al., 2025). Another line of work focuses on speeding up DLM inference time, such as Fast-dllm (Wu et al., 2025), SlowFast (Wei et al., 2025), and others (Wang et al., 2025b; Hu et al., 2025).

However, C^{2}DLM focuses on causal alignment of the attention mechanism in the SFT stage.

2.2 Combining Causality and Attention Mechanisms

Transformer (Vaswani et al., 2017) proposes the multi-head attention mechanism, which models dependencies between tokens:

$$\mathbf{A}_i^{attn} = \text{softmax}\left(\frac{\mathbf{Q}_i \cdot \mathbf{K}_i^\top}{\sqrt{d_k}}\right) \cdot \mathbf{V}_i,$$

where for each head i in multi-head attention, $\mathbf{Q}_i, \mathbf{K}_i \in \mathbb{R}^{n\times d_k}$,$\mathbf{V}_i \in \mathbb{R}^{n\times d_v}$.

Although attention and causal (Pearl, 2009; Han et al., 2025a) graphs are correlated in the covariance structure (Rohekar et al., 2023; Han et al., 2024), the attention sink (Sun et al., 2024; Xiao et al., 2023; Gu et al., 2024) phenomenon reveals outliers in the attention distribution, reducing interpretability (Kobayashi et al., 2020). To correct and denoise attention, some studies leverage causal backdoor mechanisms for debiasing in text (Wu et al., 2024b) and vision (Yang et al., 2021). Recent work proposed the Re-attention mechanism (Han et al., 2025b) to inject causal knowledge into a student model.

C^{2}DLM provides a new paradigm for investigating attention mechanisms in DLMs, interpreting the importance of guiding the model to align with data generation priors.

2.3 The Limitations of Attention in AR and DLM

The limitations of AR and DLM models stem from the structural priors imposed by their attention mechanisms. The AR model, with its lower-triangular attention matrix and modeling objective $P(x) = \prod_i p_\theta(x_i|x_{<i})$, is unable to handle situations in natural language where the outcome precedes the cause. On the other hand, DLMs can be regarded as an any-order AR model (Arriola et al., 2025). DLMs adopt a fully connected structure that can model arbitrary dependencies: $P(x_i) = p_\theta(x_i \mid x_{\neq i})$. The absence of causal constraints causes key causal signals in each step of the COT process to be diluted by redundant information from both past and future contexts. This makes it difficult for the model to perform stable and effective reasoning over COT.

3 Method

As shown in Figure 2, C^{2}DLM consists of two main steps: (1) concept-level causal meta-knowledge extraction, and (2) causal alignment via the V-aware Re-attention mechanism.

3.1 Concept-level Causal Meta-knowledge Extraction

Humans, when confronted with downstream tasks, analyze relationships among conceptual entities, perform reasoning, and integrate contextual information to verbalize their thought process in natural language. Inspired by this, we extract and construct concept-level reasoning graphs for tasks described in natural language. Each concept encapsulates the core information necessary for reasoning and reflects human causal logic, thereby encoding the true and flexible priors underlying the data. Therefore, the generating function of language should be consistent with human prior understanding of causal concepts.

To reduce the cost of generating such priors, we design an automated workflow. The teacher model first extracts a set of concepts $\mathcal{C} = \{c_1, c_2, \ldots, c_n\}$ from the reasoning steps, and denotes the remaining text as context $\mathcal{T}$. Each concept $c \in \mathcal{C}$ represents a semantically complete entity or sentence. The teacher then constructs a reasoning graph over $\mathcal{C}$, capturing causal dependencies between concepts.

Unlike prior work (Han et al., 2025b), the graph is not restricted to pairwise causal forms such as c_A causes c_B. For a given concept c_A, information from c_B that is unnecessary for generation is pruned, preventing reverse dependencies. For example, as shown in Figure 2(a), the teacher decomposes the problem into four steps and identifies causal meta-knowledge within each step. Conditions like "2, 3, 5, 7, 11, 13, 17, 19" and "each prime factor of 20! must be assigned entirely to a or b" together imply 256 valid (a, b) pairs, though this fact is not required when generating those conditions. For certain tasks, we further introduce a rule-based, semi-autoregressive supervisory signal that decomposes the reasoning chain into coarser-grained steps $s \in \mathcal{S}$ based on inter-rule inference. This mechanism prunes irrelevant information from earlier steps, improving the efficiency of subsequent reasoning. Prompt details are provided in Appendix A.

Based on the above constraints, we define the prior supervision mask as:

$$M_{j,i} = \begin{cases} 1, & c_i, c_j \in \mathcal{C},\ c_i \to c_j, \\ 0, & c_i \in \mathcal{T} \text{ or } c_j \in \mathcal{T}, \\ -1, & c_i, c_j \in \mathcal{C},\ c_j \to c_i \text{ or } s_i > s_j. \end{cases}$$

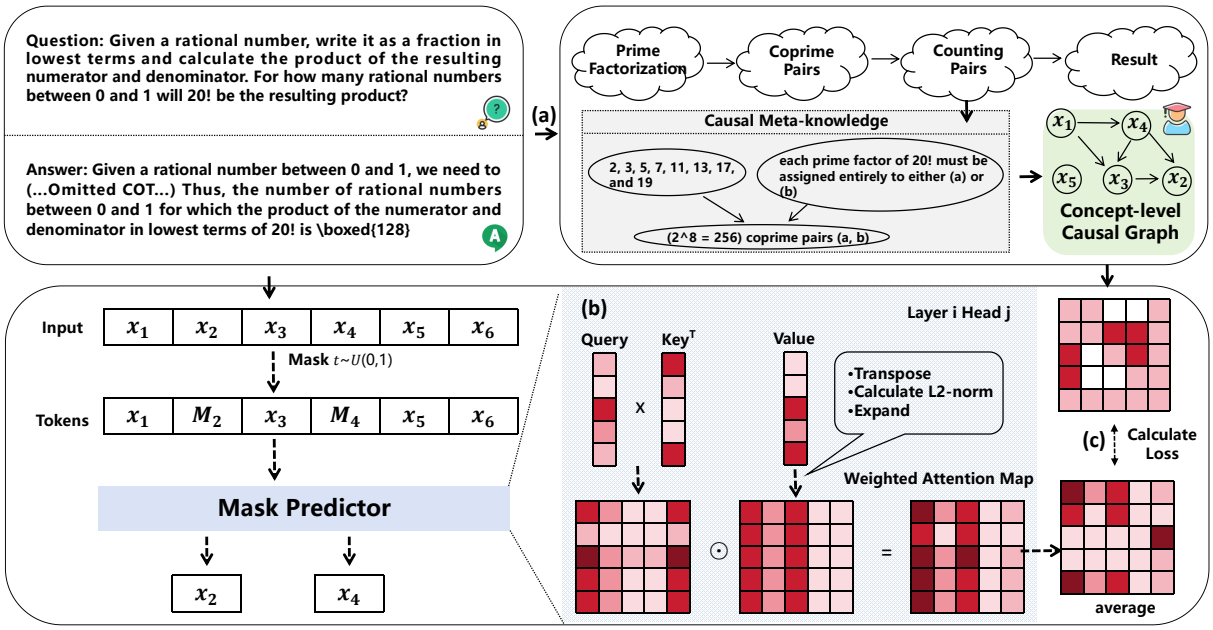

Figure 2: (a) Leveraging the contextual learning capability of a strong model, the causal teacher model uses prompts to automatically extract concept-level information from COTs and generates causal meta-knowledge links between concepts as supervisory signals. (b) During training, for the internal attention map obtained from COTs, the V-aware Re-attention mechanism weights the attention maps by the norms of the corresponding value matrix. (c) The tokenizer maps the textual supervisory signals from step (a) to the weighted attention maps, and a loss-based intervention is applied to guide the C^{2}DLM's decision-making process.

where c and s are concepts and steps indexed by token i and j respectively.

3.2 Causal Alignment via the V-aware Re-attention Mechanism

To align the model's decision dependencies with the underlying causal mechanisms of natural language and eliminate instability caused by outliers in the attention map, we propose the *V-aware Re-attention* mechanism. With respect to the supervisory mask introduced in the previous section, we define the index sets as

$$I_k = \{j \mid M_{i,j} = k\}, \quad k \in \{-1, 0, 1\}.$$

Because the attention map can be distorted by the attention sink phenomenon, its raw values may fail to accurately reflect token-level interactions. To address this, we incorporate the L2-norm of the value matrix as weighting information. Let $A^{(h)} \in \mathbb{R}^{T_q \times T_k}$ denote the attention map of the h-th head, and $V^{(h)} \in \mathbb{R}^{T_q \times d_h}$ the corresponding value matrix. We use the L2 norm of the value matrix as weighted information (Kobayashi et al., 2020):

$$\|V_i^{(h)}\|_2 = \sqrt{\sum_{d=1}^{d_h} \left(V_{i,d}^{(h)}\right)^2}, \quad i = 1, \ldots, T_q.$$

The weighted attention map is then

$$\widetilde{A}_{i,j}^{(h)} = A_{i,j}^{(h)} \cdot \|V_i^{(h)}\|_2, \quad \forall i \in [1, T_q],\ j \in [1, T_k],$$

and averaging across n_h heads yields

$$\widetilde{A}_{i,j} = \frac{1}{n_h} \sum_{h=1}^{n_h} \widetilde{A}_{i,j}^{(h)}.$$

Based on $\widetilde{A}$, we compute the average attention values for encouraged and neutral sets as

$$\bar{A}_1 = \frac{1}{|I_1|} \sum_{j \in I_1} \widetilde{A}_{i,j}, \qquad \bar{A}_0 = \frac{1}{|I_0|} \sum_{j \in I_0} \widetilde{A}_{i,j}.$$

The ratio loss for the i-th row is

$$\mathcal{L}_{ratio}(i) = \begin{cases} -\dfrac{\bar{A}_1}{\bar{A}_1 + \bar{A}_0}, & \dfrac{\bar{A}_1}{\bar{A}_0} < \alpha, \\ 0, & \text{otherwise}, \end{cases}$$

where $\alpha > 0$ enforces a minimum ratio between encouraged and neutral attentions. In addition, for masked entries with $M_{i,j} = -1$, we penalize the squared weighted attention values:

$$\mathcal{L}_{neg}(i) = \lambda \sum_{j \in I_{-1}} \widetilde{A}_{i,j}^2,$$

with $\lambda > 0$ controlling the penalty strength. The total loss for the i-th row is then

$$\mathcal{L}_{row}(i) = \mathcal{L}_{neg}(i) + \mathcal{L}_{ratio}(i).$$

For the $\mathcal{J}_r$ valid rows that have supervisory signals, we apply weighting and combine them with the DLM downstream SFT loss (as described in Section 2.1) to obtain the final training loss:

$$\mathcal{L}_{total} = \mathcal{L}_{DLM} + \frac{\gamma}{|\mathcal{J}_r|} \sum_{i \in \mathcal{J}_r} \mathcal{L}_{row}(i),$$

where $\gamma > 0$ is a balancing coefficient controlling the relative strength of the proposed constraint loss.

Because we directly intervene on the weights of the attention matrix, we introduce a smoothing mechanism to stabilize training. Inspired by learning rate scheduling, we define a γ-parameter scheduler $\mathcal{S}_\gamma$ that modulates γ over training steps. Specifically, for the initial set of steps, γ increases linearly from $\gamma_{\min}$ to $\gamma_{\max}$, and for the subsequent steps, it decreases linearly back to $\gamma_{\min}$. Formally, this can be expressed as:

$$\gamma_t = \begin{cases} \gamma_{\min} + \frac{t}{T_1}(\gamma_{\max} - \gamma_{\min}), & t \in [0, T_1] \\ \gamma_{\max} - \frac{t - T_1}{T_2 - T_1}(\gamma_{\max} - \gamma_{\min}), & t \in [T_1, T_2], \end{cases}$$

where t denotes the current training step, T_1 is the number of warm-up steps, and $T_2 - T_1$ is the number of cool-down steps.

4 Experimental Results

4.1 Experimental Setup

Datasets. We first constructed a synthetic dataset, COT-OrderPerturb, to examine how the ordering of concepts within COT influences AR and DLM. We then employed the Sudoku* and STG (Han et al., 2025b) datasets to assess how models benefit when downstream tasks exhibit explicit causal structures. Finally, we evaluated broader reasoning-related downstream tasks, including MATH500 (Lightman et al., 2023), GSM8K (Cobbe et al., 2021), GPQA (Rein et al., 2024), ARC_C (Clark et al., 2018), SAT (Zhong et al., 2023), and MMLU_STEM (Hendrycks et al., 2021b,a).

Baselines and hyperparameters. We adopt LLaDA-8B-Instruct† as the primary experimental model and apply LoRA (Hu et al., 2022) for fine-tuning, with SFT serving as the main DLM baseline. During training and evaluation, all hyperparameters are consistent except for the loss introduced by C^{2}DLM. In addition, we include the following commonly used AR models for comparison: Llama-3.1-8B, Llama-3.2-1B (Dubey et al., 2024), Qwen-2.5-1.5B, and Qwen3-8B (Yang et al., 2025). For the γ-parameter scheduler, we set $T_1 = 0.1 \times T_2$. For the λ-parameter, we set 100 for the COT-OrderPerturb and 10 for all other tasks. The learning rate is uniformly fixed at 2×10^{-5}, and the LoRA rank is set to 128. For STG and Sudoku, we set α as 5, and for other tasks, α is 3. Unless otherwise specified, the block length during the test defaults to 32. More detailed hyperparameter configurations are provided in Appendix B.

4.2 Quantifying the Impact of Priors

Previous studies (Hu et al., 2026a) have shown that order has a certain impact on the reasoning performance of diffusion-based models. To verify the impact of AR and DLM attention limitations, we propose the COT-OrderPerturb synthetic dataset, thereby quantifying the impact of misalignment between the attention mechanism's modeling priors of natural language and the causal priors underlying natural language. We first generate COT simulation data based on a given prior causal graph, as shown in Figure 3, including both COTs that follow the standard causal order and COTs with permuted concept sequences, where outcomes may precede their causes. To systematically explore different types of perturbations, we apply the following shuffling strategies: DFS, local reverse (LR), output first (OF), reverse (RE), and three random shuffles R_1, R_2, and R_3 (details in the Appendix C), along with a control condition named No COT in which answers are generated directly without COT reasoning. Results are summarized in Table 1.

Structural Bias in AR Models. We observe that AR models exhibit declining performance consistency when data is perturbed such that outcomes precede their causes. However, linguistic flexibility is not bound to a strict left-to-right, token-by-token causal order, e.g., "The ground is slippery today because it rained" or "Lung cancer is caused by smoking". The outcome may precede the cause. The misalignment between AR priors and underlying causal priors of natural language introduces structural risks that cannot be resolved by simply scaling the training data.

*https://github.com/Black-Phoenix/4x4-Sudoku-Dataset
†https://github.com/ML-GSAI/LLaDA/

Model	Normal COT	Shuffle COT								No COT
		DFS	LR	OF	RE	R_1	R_2	R_3	avg ± std	
AR										
Llama-3.2-1B	22.40%	20.60%	25.80%	31.40%	22.60%	25.00%	24.00%	24.20%	24.80%±3.37%	15.60%
Qwen3-8B	**60.60%**	2.41%	44.20%	0.20%	0.20%	23.00%	32.40%	33.20%	19.37%±18.32%	36.40%
Llama-3.1-8B	47.60%	18.20%	**44.00%**	14.00%	21.40%	4.00%	32.60%	29.80%	23.43%±13.20%	44.60%
DLM										
LLaDA-8B-Instruct (SFT)	38.60%	**38.20%**	36.60%	**42.40%**	**41.80%**	**33.80%**	**35.00%**	**40.60%**	**38.34%±3.38%**	**57.60%**

Table 1: Performance of AR and DLM under different settings on the COT-OrderPerturb dataset.

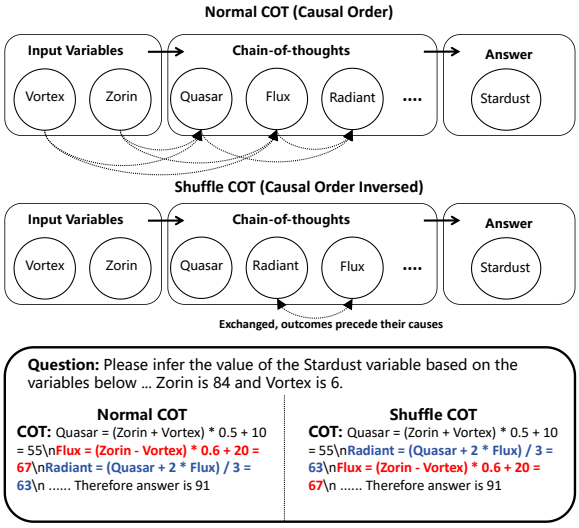

Figure 3: Normal COT follows the causal topological order of the data-generating process to construct reasoning steps, whereas the Shuffle setting simulates cases where COT exhibits causal misordering.

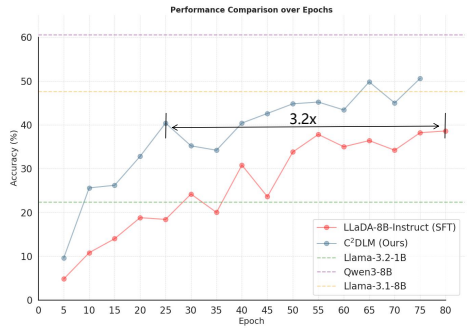

Figure 4: Accuracy curve as training progresses in the COT-OrderPerturb task.

Model	Normal COT
LLaDA-8B-Instruct (SFT)	38.60%
C^2DLM (ours)	**50.60%**
Δ	+12.00%

Table 2: Performance comparison under the Normal COT setting. Notation Δ denotes the performance gain relative to direct SFT on LLaDA-8B-Instruct.

Robustness from Order-Independence in DLMs. The DLM, trained with fully connected attention and order-independent masking, demonstrates greater robustness. In the shuffled COT setting, DLM achieves both a better mean and standard deviation of accuracy.

Performance Bottlenecks of DLMs. Interestingly, DLMs perform notably worse than AR models under the Normal COT setting. While AR models consistently benefit from COT supervision, DLMs achieve substantially higher performance in the No-COT condition than when COT is included. This discrepancy arises because, by discarding causal order, DLM training effectively becomes a form of multi-objective learning across all reasoning steps. With limited data, this hinders the acquisition of deeper reasoning capabilities. Moreover, the longer COT sequences further exacerbate inefficiency: DLMs typically require nearly 80 epochs to converge, whereas AR models converge within only 4 epochs.

Performance Gain Using C^2DLM. C^2DLM explicitly incorporates step-wise conceptual causal relationships, guiding the model to learn the data generation process via a V-aware Re-attention mechanism, and suppresses attention on element interactions that violate the causal structure. As shown in Table 2, aligning with causal priors leads to a significant performance improvement of 12%, surpassing that of Llama-3.1-8B. Moreover, as illustrated in Figure 4, the training efficiency of C^2DLM is 3.2 times that of DLM.

4.3 Downstream Task Experiments with Explicit Causal Structures

Some downstream tasks contain explicit causal structures as priors in their data generation processes. We selected Sudoku and STG as representative benchmarks to evaluate C^2DLM.

4.3.1 Sudoku dataset

For the Sudoku task, we focus on a 4×4 grid, which is consistent with the setting in (Zhao et al., 2025a). In Sudoku, each number is determined by

Setting	Sudoku		
	n=200	n=500	n=5000
AR			
Llama-3.2-1B	3.00%	22.40%	80.60%
Qwen3-8B	67.40%	76.40%	83.00%
Llama-3.1-8B	8.60%	29.20%	80.80%
DLM			
LLaDA-8B-Instruct (SFT)	77.05%	90.23%	92.14%
C^{2}DLM (ours)	**87.89%**	**91.21%**	**92.97%**
Δ	+10.84%	+0.98%	+0.83%

Table 3: Performance on Sudoku task. Here, n denotes the size of the training data. Notation Δ denotes the performance gain relative to direct SFT on LLaDA-8B-Instruct.

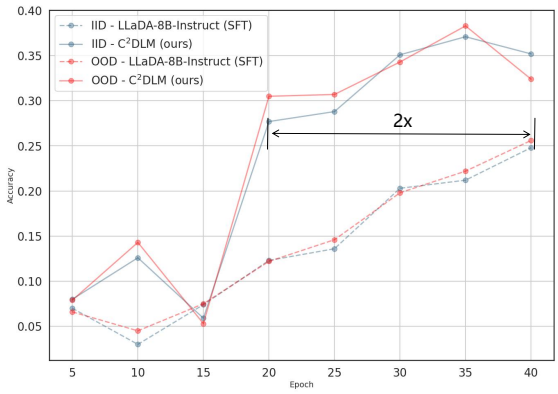

Figure 5: Performance change curve during different epochs of training on the STG_H dataset.

the values in its row, column, and corresponding subgrid. Experimental results in Table 3 indicate that AR approaches are constrained by the unidirectional flow of information, causing a misalignment between attention priors and the data generation process, which leads to low performance. By contrast, C^{2}DLM effectively leverages the causal priors, which allows the model to better fit the data and avoid learning spurious and unrelated correlations. This advantage is particularly obvious in small-data scenarios ($n = 200$), where LLaDA-8B-Instruct lacks clear supervisory guidance and performs worse than C^{2}DLM.

4.3.2 STG dataset

Similarly, the STG dataset is also generated from explicit causal graphs and provides both IID and OOD (Han et al., 2025b; Tong et al., 2023) testing scenarios, which facilitate a more systematic evaluation of robustness in the presence of spurious correlations. As shown in Table 4, C^{2}DLM significantly outperforms direct SFT across different STG subsets, with an average improvement of 7.43% across IID and OOD settings. Compared to AR, introducing the causal prior via C^{2}DLM markedly narrows their gap and even surpasses the best AR baselines in the OOD setting of STG_S, STG_M, and STG_L.

We further examined the training efficiency of C^{2}DLM in STG_H. As shown in Figure 5, the training efficiency of C^{2}DLM is 2 times that of DLM.

Figure 6: Attention visualization and weight distribution bar charts, where the x-axis represents different attributes in the STG task. Purple indicates causal factors, green denotes spurious correlations, and yellow represents unrelated factors.

To better interpret the impact of C^{2}DLM on the attention mechanism, we conducted attention visualizations weighted by the value matrix. As shown in Figure 6, C^{2}DLM effectively learns causal relationships, whereas direct fine-tuning of LLaDA fails to distinguish spurious correlations and irrelevant factors from causal factors. By mechanistically guiding the model to fit the data generation process, C^{2}DLM yields more robust predictions and enhances model reliability. Further analysis is provided in Appendix E.

4.4 Evaluation on Broader Math and Reasoning Downstream Tasks

For broader downstream tasks without explicit causal structure, we leverage the automated workflow introduced in Section 3.1. Specifically, we adopt the latest GLM-4.5 (Zeng et al., 2025), which provides a balance between scalability and cost-

Setting	STG_S		STG_M		STG_L		STG_H	
	IID	OOD	IID	OOD	IID	OOD	IID	OOD
AR								
Llama-3.2-1B	83.50%	67.25%	**95.75%**	61.50%	**97.25%**	87.00%	37.40%	27.90%
Qwen-2.5-1.5B†	81.50%	78.50%	93.25%	82.00%	95.75%	82.00%	51.50%	**53.40%**
Llama-3.1-8B†	**90.50%**	86.25%	93.25%	64.50%	96.00%	88.25%	**57.80%**	49.60%
DLM								
LLaDA-8B-Instruct (SFT)	86.50%	81.25%	85.75%	83.00%	88.25%	83.25%	24.80%	25.60%
C^{2}DLM (ours)	88.00%	**88.50%**	93.00%	**95.00%**	93.50%	**92.25%**	35.20%	32.40%
Δ	+1.50%	+7.25%	+7.25%	+12.00%	+5.25%	+9.00%	+10.40%	+6.80%

Table 4: Performance on STG task. Notation "†" means results from (Han et al., 2025b). Notation Δ denotes the performance gain relative to direct SFT on LLaDA-8B-Instruct.

Setting	GSM8K	MATH500	GPQA	MMLU_STEM	ARC_C	SAT	Avg
LLaDA-8B-Instruct	80.36%	36.60%	28.79%	58.74%	85.75%	71.36%	60.27%
LLaDA-8B-Instruct (SFT)	80.74%	36.20%	29.24%	59.21%	85.67%	75.00%	61.01%
C^{2}DLM (ours)	**81.96%**	**37.20%**	**29.46%**	**60.42%**	**86.26%**	**78.64%**	**62.32%**
Δ	+1.22%	+1.00%	+0.22%	+1.21%	+0.59%	+3.64%	+1.31%

Table 5: Performance on diverse math and reasoning downstream datasets. Notation Δ denotes the performance gain relative to direct SFT on LLaDA-8B-Instruct.

effectiveness while addressing the challenges of extracting causal relationships from long COT sequences. Aligning with previous work (Zhao et al., 2025a; Tang et al., 2025), we start from the s1k dataset and construct a training dataset containing 686 instances annotated by the GLM-4.5.

We conducted a manual random sampling of 50 instances to evaluate the causal graphs generated by the teacher model. Two instances (4% of the sampled data) failed to produce causal graphs due to decoding errors. Among the successfully decoded instances, the accuracy was 93.42% ± 1.41%. Detailed experimental procedures are provided in Appendix F.

Both SFT and C^{2}DLM are trained on this dataset, with the only difference being that C^{2}DLM incorporates the causal prior loss. The resulting models are then evaluated on six downstream tasks. As shown in Table 5, C^{2}DLM achieves consistent improvements across six test datasets, with an average gain of 1.31%. Notably, these gains are obtained using only 686 causally annotated examples. As performance improvements from scaling next-token prediction alone are approaching a bottleneck, our pipeline highlights the potential of leveraging two-dimensional supervision signals based on token interactions as a promising direction for future scaling, with the cost as low as $0.46 per million tokens (see Appendix D for details).

4.5 Ablation Study

Setting	GSM8K	MATH500	SAT	Avg
$\alpha = 2$	81.43%	**37.60%**	77.73%	65.58%
$\alpha = 3$	**81.96%**	37.20%	**78.64%**	**65.93%**
w/o $\mathcal{S}_\gamma$	81.65%	33.20%	74.09%	62.98%
w/o V-aware	81.35%	34.00%	68.64%	61.33%
$\alpha = 4$	81.65%	34.00%	76.82%	64.16%
$\alpha = 5$	81.50%	36.80%	78.18%	65.49%

Table 6: Ablation study under different α and γ scheduler. Gray line ($\alpha = 3$) is the default setting. The notation w/o $\mathcal{S}_\gamma$ means that the γ scheduler is not used.

To analyze the effects of different components and hyperparameters, we conducted ablation studies on the parameters α and the γ scheduler. α represents the degree of emphasis on causal relationships, and an appropriate level of emphasis contributes to improved model performance. Additionally, we evaluated an ablation of the V-aware strategy, in which the model performance was assessed without the weighting provided by the value matrix (denoted as w/o V-aware). As shown in Table 6, within a certain range, model performance first improves and then declines as α increases. We further examined the impact of the γ scheduler. Without the γ scheduler, the performance of C^{2}DLM declines on all datasets. When the V-aware strategy is not employed, and causal knowledge is directly injected, performance decreases. This is due to the direct manipulation of attention scores.

Ignoring the influence of the value matrix introduces substantial instability during training, which in turn leads to performance degradation. These results demonstrate the effectiveness of the V-aware strategy.

5 Conclusion

To address the limitations of language models in reasoning, we propose C^{2}DLM, a new paradigm distinct from AR and DLM. C^{2}DLM leverages automated pipelines to extract causal meta-knowledge and employs the V-aware Reattention mechanism to align attention. We propose the COT-OrderPerturb task to quantify the influence of language modeling priors, and we validate the effectiveness of C^{2}DLM on Sudoku, STG, and six broad downstream tasks. C^{2}DLM improves both the model's reasoning ability and training efficiency. Furthermore, we reveal the risk of misalignment between attention mechanisms and the causal priors underlying natural language, which shows the potential of combining causality into language models.

Limitations

Our experiments focus on the LLaDA-8B-Instruct model, but due to constraints in training resources and the base model, C^{2}DLM's performance still lags behind the SOTA AR models. Furthermore, larger-scale DLMs remain underexplored, so the effectiveness of C^{2}DLM on such models is still unknown. Limited by computational resources, we are unable to inject causal knowledge at scale during pretraining; the pretraining stage remains largely unexamined, and our current work focuses solely on the SFT phase. Causal knowledge in the real world is highly complex, and thus extracting causal structures and benefiting from them in more intricate causal graphs or ultra-long COT remains a significant challenge.

Ethical Considerations

A potential risk of this work lies in the possibility that the proposed method could be misused to inject illegal or unethical information into models. Therefore, we strongly urge users to ensure that all training datasets comply with relevant laws and ethical guidelines when applying this approach.

Acknowledgments

This work was supported in part by the National Key Research and Development Program of China (2024YFE0203700), "Pioneer" and "Leading Goose" R&D Program of Zhejiang (2025C02037), and National Natural Science Foundation of China (62376243). All opinions in this paper are those of the authors and do not necessarily reflect the views of the funding agencies.

References

Deepak Bhaskar Acharya, Karthigeyan Kuppan, and B. Divya. 2025. Agentic ai: Autonomous intelligence for complex goals—a comprehensive survey. *IEEE Access*, 13:18912–18936.

Marianne Arriola, Aaron Gokaslan, Justin T Chiu, Zhihan Yang, Zhixuan Qi, Jiaqi Han, Subham Sekhar Sahoo, and Volodymyr Kuleshov. 2025. Block diffusion: Interpolating between autoregressive and diffusion language models. *arXiv preprint arXiv:2503.09573*.

Jacob Austin, Daniel D Johnson, Jonathan Ho, Daniel Tarlow, and Rianne Van Den Berg. 2021. Structured denoising diffusion models in discrete state-spaces. *Advances in neural information processing systems*, 34:17981–17993.

Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, and 1 others. 2023. Sparks of artificial general intelligence: Early experiments with gpt-4. *arXiv preprint arXiv:2303.12712*.

Hanqun Cao, Cheng Tan, Zhangyang Gao, Yilun Xu, Guangyong Chen, Pheng-Ann Heng, and Stan Z Li. 2024. A survey on generative diffusion models. *IEEE transactions on knowledge and data engineering*, 36(7):2814–2830.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv:1803.05457v1*.

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*.

Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Jingyuan Ma, Rui Li, Heming Xia, Jingjing Xu, Zhiyong Wu, Tianyu Liu, and 1 others. 2022. A survey on in-context learning. *arXiv preprint arXiv:2301.00234*.

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. *arXiv e-prints*, pages arXiv–2407.

Xiangming Gu, Tianyu Pang, Chao Du, Qian Liu, Fengzhuo Zhang, Cunxiao Du, Ye Wang, and Min Lin. 2024. When attention sink emerges in language models: An empirical view. *arXiv preprint arXiv:2410.10781*.

Kairong Han, Weidong Huang, Taiyang Zhou, Peng Zhen, and Kun Kuang. 2025a. Augmenting limited and biased rcts through pseudo-sample matching-based observational data fusion method. In *Proceedings of the 34th ACM International Conference on Information and Knowledge Management*, pages 5715–5722.

Kairong Han, Kun Kuang, Ziyu Zhao, Junjian Ye, and Fei Wu. 2024. Causal agent based on large language model. *arXiv preprint arXiv:2408.06849*.

Kairong Han, Wenshuo Zhao, Ziyu Zhao, Ye Jun Jian, Lujia Pan, and Kun Kuang. 2025b. CAT: Causal attention tuning for injecting fine-grained causal knowledge into large language models. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 9904–9921, Suzhou, China. Association for Computational Linguistics.

Dan Hendrycks, Collin Burns, Steven Basart, Andrew Critch, Jerry Li, Dawn Song, and Jacob Steinhardt. 2021a. Aligning ai with shared human values. *Proceedings of the International Conference on Learning Representations (ICLR)*.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021b. Measuring massive multitask language understanding. *Proceedings of the International Conference on Learning Representations (ICLR)*.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, and 1 others. 2022. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3.

Zhanqiu Hu, Jian Meng, Yash Akhauri, Mohamed S Abdelfattah, Jae-sun Seo, Zhiru Zhang, and Udit Gupta. 2025. Accelerating diffusion language model inference via efficient kv caching and guided diffusion. *arXiv preprint arXiv:2505.21467*.

Zijing Hu, Yunze Tong, Fengda Zhang, Junkun Yuan, Jun Xiao, and Kun Kuang. 2026a. Asynchronous denoising diffusion models for aligning text-to-image generation. In *The Fourteenth International Conference on Learning Representations*.

Zijing Hu, Junkun Yuan, Kairong Han, Yunze Tong, Shengyu Zhang, Fei Wu, and Kun Kuang. 2026b. Reinforcement learning in generative multimodal ai: A survey.

Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, and 1 others. 2025. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *ACM Transactions on Information Systems*, 43(2):1–55.

Leslie Pack Kaelbling, Michael L Littman, and Andrew W Moore. 1996. Reinforcement learning: A survey. *Journal of artificial intelligence research*, 4:237–285.

Subbarao Kambhampati, Karthik Valmeekam, Lin Guan, Mudit Verma, Kaya Stechly, Siddhant Bhambri, Lucas Saldyt, and Anil Murthy. 2024. Llms can't plan, but can help planning in llm-modulo frameworks. *arXiv preprint arXiv:2402.01817*.

Jaeyeon Kim, Kulin Shah, Vasilis Kontonis, Sham Kakade, and Sitan Chen. 2025. Train for the worst, plan for the best: Understanding token ordering in masked diffusions. *arXiv preprint arXiv:2502.06768*.

Goro Kobayashi, Tatsuki Kuribayashi, Sho Yokoi, and Kentaro Inui. 2020. Attention is not only a weight: Analyzing transformers with vector norms. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7057–7075, Online. Association for Computational Linguistics.

Tamera Lanham, Anna Chen, Ansh Radhakrishnan, Benoit Steiner, Carson Denison, Danny Hernandez, Dustin Li, Esin Durmus, Evan Hubinger, Jackson Kernion, and 1 others. 2023. Measuring faithfulness in chain-of-thought reasoning. *arXiv preprint arXiv:2307.13702*.

Tianyi Li, Mingda Chen, Bowei Guo, and Zhiqiang Shen. 2025a. A survey on diffusion language models. *arXiv preprint arXiv:2508.10875*.

Yuming Li, Yikai Wang, Yuying Zhu, Zhongyu Zhao, Ming Lu, Qi She, and Shanghang Zhang. 2025b. Branchgrpo: Stable and efficient grpo with structured branching in diffusion models. *arXiv preprint arXiv:2509.06040*.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2023. Let's verify step by step. In *The Twelfth International Conference on Learning Representations*.

Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, and 1 others. 2024. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*.

Shen Nie, Fengqi Zhu, Zebin You, Xiaolu Zhang, Jingyang Ou, Jun Hu, Jun Zhou, Yankai Lin, Ji-Rong Wen, and Chongxuan Li. 2025. Large language diffusion models. *arXiv preprint arXiv:2502.09992*.

Melissa Z Pan, Mert Cemri, Lakshya A Agrawal, Shuyi Yang, Bhavya Chopra, Rishabh Tiwari, Kurt Keutzer, Aditya Parameswaran, Kannan Ramchandran, Dan Klein, and 1 others. 2025. Why do multiagent systems fail? In *ICLR 2025 Workshop on Building Trust in Language Models and Applications*.

Judea Pearl. 2009. *Causality*. Cambridge university press.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R Bowman. 2024. Gpqa: A graduate-level google-proof q&a benchmark. In *First Conference on Language Modeling*.

Raanan Y Rohekar, Yaniv Gurwicz, and Shami Nisimov. 2023. Causal interpretation of self-attention in pre-trained transformers. *Advances in Neural Information Processing Systems*, 36:31450–31465.

Mingjie Sun, Xinlei Chen, J Zico Kolter, and Zhuang Liu. 2024. Massive activations in large language models. *arXiv preprint arXiv:2402.17762*.

Xiaohang Tang, Rares Dolga, Sangwoong Yoon, and Ilija Bogunovic. 2025. wd1: Weighted policy optimization for reasoning in diffusion language models. *arXiv preprint arXiv:2507.08838*.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, Katie Millican, and 1 others. 2023. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*.

Yunze Tong, Junkun Yuan, Min Zhang, Didi Zhu, Keli Zhang, Fei Wu, and Kun Kuang. 2023. Quantitatively measuring and contrastively exploring heterogeneity for domain generalization. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*.

Yunze Tong, Fengda Zhang, Didi Zhu, Jun Xiao, and Kun Kuang. 2025. Decoding correlation-induced misalignment in the stable diffusion workflow for text-to-image generation. In *Proceedings of the IEEE/CVF international conference on computer vision*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Wen Wang, Bozhen Fang, Chenchen Jing, Yongliang Shen, Yangyi Shen, Qiuyu Wang, Hao Ouyang, Hao Chen, and Chunhua Shen. 2025a. Time is a feature: Exploiting temporal dynamics in diffusion language models. *arXiv preprint arXiv:2508.09138*.

Xu Wang, Chenkai Xu, Yijie Jin, Jiachun Jin, Hao Zhang, and Zhijie Deng. 2025b. Diffusion llms can do faster-than-ar inference via discrete diffusion forcing. *arXiv preprint arXiv:2508.09192*.

Qingyan Wei, Yaojie Zhang, Zhiyuan Liu, Dongrui Liu, and Linfeng Zhang. 2025. Accelerating diffusion large language models with slowfast: The three golden principles. *arXiv preprint arXiv:2506.10848*.

Anpeng Wu, Kun Kuang, Minqin Zhu, Yingrong Wang, Yujia Zheng, Kairong Han, Baohong Li, Guangyi Chen, Fei Wu, and Kun Zhang. 2024a. Causality for large language models. *Preprint*, arXiv:2410.15319.

Chengyue Wu, Hao Zhang, Shuchen Xue, Zhijian Liu, Shizhe Diao, Ligeng Zhu, Ping Luo, Song Han, and Enze Xie. 2025. Fast-dllm: Training-free acceleration of diffusion llm by enabling kv cache and parallel decoding. *arXiv preprint arXiv:2505.22618*.

Yiquan Wu, Yifei Liu, Ziyu Zhao, Weiming Lu, Yating Zhang, Changlong Sun, Fei Wu, and Kun Kuang. 2024b. De-biased attention supervision for text classification with causality. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(17):19279–19287.

Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. 2023. Efficient streaming language models with attention sinks. *arXiv preprint arXiv:2309.17453*.

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

Ling Yang, Zhilong Zhang, Yang Song, Shenda Hong, Runsheng Xu, Yue Zhao, Wentao Zhang, Bin Cui, and Ming-Hsuan Yang. 2023. Diffusion models: A comprehensive survey of methods and applications. *ACM computing surveys*, 56(4):1–39.

Xu Yang, Hanwang Zhang, Guojun Qi, and Jianfei Cai. 2021. Causal attention for vision-language tasks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9847–9857.

Jiacheng Ye, Jiahui Gao, Shansan Gong, Lin Zheng, Xin Jiang, Zhenguo Li, and Lingpeng Kong. 2024. Beyond autoregression: Discrete diffusion for complex reasoning and planning. *arXiv preprint arXiv:2410.14157*.

Jiacheng Ye, Zhihui Xie, Lin Zheng, Jiahui Gao, Zirui Wu, Xin Jiang, Zhenguo Li, and Lingpeng Kong. 2025. Dream 7b: Diffusion large language models. *arXiv preprint arXiv:2508.15487*.

Asaf Yehudai, Lilach Eden, Alan Li, Guy Uziel, Yilun Zhao, Roy Bar-Haim, Arman Cohan, and Michal Shmueli-Scheuer. 2025. Survey on evaluation of llm-based agents. *arXiv preprint arXiv:2503.16416*.

Matej Zečević, Moritz Willig, Devendra Singh Dhami, and Kristian Kersting. 2023. Causal parrots: Large language models may talk causality but are not causal. *arXiv preprint arXiv:2308.13067*.

Aohan Zeng, Xin Lv, Qinkai Zheng, Zhenyu Hou, Bin Chen, Chengxing Xie, Cunxiang Wang, Da Yin, Hao Zeng, Jiajie Zhang, and 1 others. 2025. Glm-4.5: Agentic, reasoning, and coding (arc) foundation models. *arXiv preprint arXiv:2508.06471*.

Siyan Zhao, Devaansh Gupta, Qinqing Zheng, and Aditya Grover. 2025a. d1: Scaling reasoning in diffusion large language models via reinforcement learning. *arXiv preprint arXiv:2504.12216*.

Siyan Zhao, Mengchen Liu, Jing Huang, Miao Liu, Chenyu Wang, Bo Liu, Yuandong Tian, Guan Pang, Sean Bell, Aditya Grover, and 1 others. 2025b. Inpainting-guided policy optimization for diffusion large language models. *arXiv preprint arXiv:2509.10396*.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, and 1 others. 2023. A survey of large language models. *arXiv preprint arXiv:2303.18223*, 1(2).

Wanjun Zhong, Ruixiang Cui, Yiduo Guo, Yaobo Liang, Shuai Lu, Yanlin Wang, Amin Saied, Weizhu Chen, and Nan Duan. 2023. Agieval: A human-centric benchmark for evaluating foundation models. *Preprint*, arXiv:2304.06364.

Fengqi Zhu, Rongzhen Wang, Shen Nie, Xiaolu Zhang, Chunwei Wu, Jun Hu, Jun Zhou, Jianfei Chen, Yankai Lin, Ji-Rong Wen, and 1 others. 2025. Llada 1.5: Variance-reduced preference optimization for large language diffusion models. *arXiv preprint arXiv:2505.19223*.

A Details of Prompt

This appendix provides the prompt templates used for concept-level causal meta-knowledge extraction with a Teacher LLM. By decomposing questions and extracting concept-level causal relations, we derive reasoning graphs, which serve as a surrogate for the underlying causal priors of natural language.

Teacher LLM's System Prompt

##Command: You are a causality expert. Your task is to identify causal relationships in a given problem and its solution, in order to generate a two-dimensional supervisory signal between tokens. Specifically, you need to extract concepts in the chain of thought. These concepts should:
1.Reflect the essential reasoning process.
2.Carry richer information content.
3.Have a stronger impact on predicting the final answer.
4.Be unique within the text, to avoid ambiguity during matching.
5.The extracted content must come strictly from the original text provided.
The extraction format should be a dictionary: { "Effect1": [Cause11, Cause12, ...], "Effect2": [Cause21, Cause22, ...], ... }.The dictionary key concept represents the effect, while the value list represents the causes. Therefore, in each segment of text, you need to carefully determine which concept is the cause and which is the effect — that is, without the cause, the effect cannot be naturally obtained. It is best if the concepts can be linked end-to-end through causal relationships, forming a complete reasoning map.

Teacher LLM's In-context Demo

Question: Given a rational number, write it as a fraction in lowest terms and calculate the product of the resulting numerator and denominator. For how many rational numbers between 0 and 1 will 20! be the resulting product?
Answer: Given a rational number between 0 and 1, we need to write it as a fraction in lowest terms $\frac{a}{b}$ and calculate the product of the resulting numerator and denominator. We need to determine how many such rational numbers exist where the product $a \times b = 20!$.

1. **Prime Factorization of 20!:**
- The prime factors of 20! are 2, 3, 5, 7, 11, 13, 17, and 19. These are 8 distinct primes.

2. **Coprime Pairs:**
- For $a \times b = 20!$ and $\gcd(a, b) = 1$, each prime factor of 20! must be assigned entirely to either a or b. This results in $2^8 = 256$ coprime pairs (a, b).

3. **Counting Pairs** $(a < b)$**:**
- Since a and b are coprime and $a \times b = 20!$, each pair (a, b) and (b, a) are distinct. To find pairs where $a < b$, we divide the total number of pairs by 2, excluding the case $a = b$ (which is impossible since 20! is not a square).

4. **Result:**
- The number of coprime pairs (a, b) with

$a < b$ is $\frac{256}{2} = 128$.

Thus, the number of rational numbers between 0 and 1 for which the product of the numerator and denominator in lowest terms is 20! is $\boxed{128}$.

Demo Output 1:
{
"2, 3, 5, 7, 11, 13, 17, and 19": ["The prime factors of 20!"],
"This results in $2^8 = 256$ coprime pairs (a, b)": ["2, 3, 5, 7, 11, 13, 17, and 19", "each prime factor of 20! must be assigned entirely to either (a) or (b)"],
"each pair (a, b) and (b, a) are distinct": ["Since (a) and (b) are coprime and $a \times b = 20!$"],
"The number of coprime pairs (a, b) with $a < b$ is $\frac{256}{2} = 128$" : ["This results in $2^8 = 256$ coprime pairs (a, b)", "we divide the total number of pairs by 2, excluding the case $(a = b)$"]
}

Demo Input 2:
Question: Square $AIME$ has sides of length 10 units. Isosceles triangle GEM has base EM, and the area common to triangle GEM and square $AIME$ is 80 square units. Find the length of the altitude to EM in $\triangle GEM$.
Answer: Square $(AIME)$ has sides of length 10 units. Isosceles triangle (GEM) has base (EM), and the area common to triangle (GEM) and square $(AIME)$ is 80 square units. We need to find the length of the altitude to (EM) in $\triangle GEM$.

1. **Vertices of the Square:**
- Place the square on a coordinate system with vertices $(A(0, 0))$, $(I(0, 10))$, $(M(10, 10))$, and $(E(10, 0))$.

2. **Isosceles Triangle (GEM):**
- Base (EM) is vertical from $(E(10, 0))$ to $(M(10, 10))$.
- The apex G of the triangle is to the left of (EM) (outside the square for larger altitudes).

3. **Coordinates of G:**
- Let the altitude from G to (EM) be h. The coordinates of G are $(10 - h, 5)$ because the triangle is isosceles with $(GE = GM)$.

4. **Equations of Lines:**
- Line (GE) has the equation $y = -\frac{5}{h}x + \frac{50}{h}$.
- Line (GM) has the equation $y = \frac{5}{h}x + 10 - \frac{50}{h}$.

5. **Intersection with the Square:**
- The lines (GE) and (GM) intersect the left edge of the square (x=0) at points $(0, \frac{50}{h})$ and $(0, 10 - \frac{50}{h})$.

6. **Area Calculation:**
- The overlap area is:

$$\text{Area} = \int_0^{10} \left(\left(\tfrac{5}{h}x + 10 - \tfrac{50}{h}\right) - \left(-\tfrac{5}{h}x + \tfrac{50}{h}\right) \right) dx$$

- Simplifying: $\text{Area} = \int_0^{10} \left(\frac{10}{h}x + 10 - \frac{100}{h}\right) dx = 100 - \frac{500}{h}$.
- Setting Area $= 80$: $100 - \frac{500}{h} = 80 \implies h = 25$.

Thus, the length of the altitude to (EM) in $\triangle GEM$ is $\boxed{25}$.

Demo Output 2:
{
"The coordinates of (G) are $(10 - h, 5)$": ["the triangle is isosceles with $(GE = GM)$", "Let the altitude from (G) to (EM) be (h)", "Place the square on a coordinate system with vertices $(A(0, 0))$, $(I(0, 10))$, $(M(10, 10))$, and $(E(10, 0))$"],
"The lines (GE) and (GM) intersect the left edge of the square (x=0) at points $(0, \frac{50}{h})$ and $(0, 10-\frac{50}{h})$": ["Line (GE) has the equation $y = -\frac{5}{h}x + \frac{50}{h}$", "Line (GM) has the equation $y = \frac{5}{h}x + 10 - \frac{50}{h}$"],
"the length of the altitude to (EM) in $\triangle GEM$ is $\boxed{25}$." : ["Setting the area equal to 80: $100 - \frac{500}{h} = 80 \implies h = 25$"]
}

B Details of Hyperparameters

Since different tasks vary in sequence length and convergence speed, we assign task-specific training epochs and generation lengths at evaluation. The detailed configurations are as follows:

For the COT-OrderPerturb task, we train for 10 epochs with a generation length of 512. To fully examine the impact of the DLM generation paradigm without any AR strategy, we additionally set the block length to 512.

For the Sudoku task, we train for 10 epochs with a generation length of 256 at evaluation.

For the STG task, the STG_E subset (including STG_S, STG_M, and STG_L) is a binary classification problem, and we train for 10 epochs. The more challenging STG_H subset is trained for 40 epochs. Since these tasks do not involve chain-of-thought reasoning, the generation length is fixed at 8.

For the six downstream tasks, we follow the setup of (Zhao et al., 2025a; Tang et al., 2025) and train on the s1k[‡], with the training context length set to 1600. At evaluation, the generation length is set to 512 for GSM8K, MATH500, and SAT, while all other choice tasks use a generation length of 32.

For the autoregressive model baselines, we uniformly adopt a LoRA learning rate of 2×10^{-4}. Sudoku is trained for 8 epochs, while all other tasks are trained for 4 epochs.

All training is conducted on 2 × NVIDIA A100 40GB GPUs, with the random seed fixed at 42 across all experiments. For testing, GSM8K, MATH500, and SAT are run on 4 × A100 40GB GPUs, while all other tasks are evaluated on 2 × A100 40GB GPUs.

Detailed implementation examples can be found in our code repository.

C Generation Details of COT-OrderPerturb Dataset

The goal of this process is to generate chain-of-thought reasoning trajectories with controlled order perturbations while preserving the underlying causal structure.

C.1 Graph Construction and Chain-of-Thought Generation

We define a directed acyclic graph (DAG) template in which nodes represent abstract variables (e.g., Quasar, Flux, Radiant, Nova), and edges encode functional dependencies. Each non-source variable is associated with a deterministic transformation rule, typically a linear or nonlinear combination of its parent variables. Two source nodes (Zorin and Vortex) are sampled uniformly from the integer range [0,100], while all other variables are computed sequentially via topological ordering. The target variable Stardust is uniquely determined by this process, ensuring consistency across samples.

Given the DAG and computed values, we construct step-wise reasoning traces. Each reasoning step includes: the functional rule applied (e.g., Quasar = (Zorin + Vortex) * 0.5 + 10), the input variables, and the evaluated output.

When arranged in strict topological order, these steps form a reasoning trajectory that faithfully reflects the data-generating process. Finally, the data generation process is shown in Figure 7. One example is as follows:

COT-OrderPerturb Example

Question: Please infer the value of the `Stardust` variable based on the variables below. The input variables are `Zorin` (value: 80) and `Vortex` (value: 79).

COT:
Quasar = (Zorin + Vortex) * 0.5 + 10 = 90
Flux = (Zorin - Vortex) * 0.6 + 20 = 21
Radiant = (Quasar + 2 * Flux) / 3 = 44
Nova = (Quasar - Flux + Zorin) / 3 + 5 = 55
Gravity = (Radiant * Quasar) / 120 + 8 = 41
Pulse = Radiant * 0.4 + Flux * 0.9 = 36
Helix = (Gravity + Pulse + Radiant) / 3 = 40
Echo = (Pulse - Flux) * 0.8 = 12
Comet = (Pulse + Gravity) * 0.6 + 2 = 48
Aether = (Echo + Gravity) * 0.5 = 26
Nebula = (Helix + Comet) / 2 + 3 = 47
Celestia = (Nebula + Aether + Echo) * 1.1 + 6 = 100
Stardust = int(Celestia * 0.7) = 70
Therefore, the final answer is 70.

Answer: 70

C.2 Order Perturbations

To examine robustness to reasoning irregularities, we apply controlled perturbations to the canonical

[‡]https://huggingface.co/datasets/simplescaling/s1K-1.1

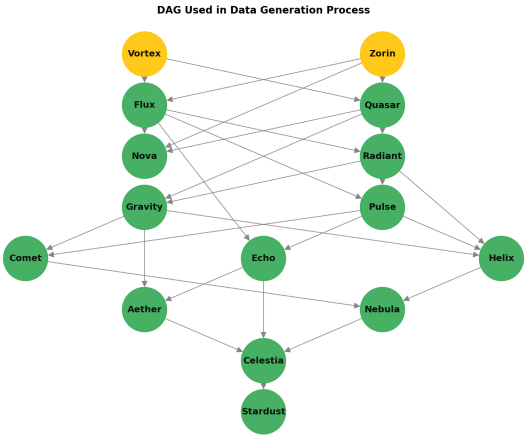

Figure 7: DAG used in data generation process. Yellow nodes are input variables. The Stardust variable is the final answer.

reasoning sequence. The perturbation modes are:

- **reverse (RE)**: complete reversal of all steps.
- **local reverse (LR)**: pairwise reversal within local consecutive steps.
- **output first (OF)**: moving the final output computation to the beginning.
- **DFS**: depth-first–style ordering based on node depth.
- **random** $\times 3$ **(R1, R2, R3)**: three fixed random permutations of reasoning steps.
- **No COT**: omission of reasoning; only the final answer is given.

These perturbations preserve the correctness of the final answer (Stardust) while only shuffling the intermediate reasoning trajectory.

In total, 2000 unique base samples are generated, yielding multiple perturbed training subsets. We generate another 500 different samples as a unique test dataset. The pseudo-code of the data generation process is shown in Algorithm 1.

D Cost Analysis of C^{2}DLM

We also analyze the cost of using GLM-4.5 for causal annotation. Specifically, we randomly select 100 examples as a representative subset to estimate annotation costs and calculate both input and output token counts. For the input, the average input length is 865.2 tokens, with an additional prompt overhead of 1981 tokens. For the output, the average length is 295.3 tokens.

Algorithm 1 COT-OrderPerturb Data Generation

Require: Number of base samples N, DAG template $\mathcal{G}$, Perturbation modes $\mathcal{M}$
Ensure: Test set $\mathcal{T}$, Training sets $\{\mathcal{D}_m\}_{m\in\mathcal{M}}$

Initialize $\mathcal{T} \leftarrow \emptyset$, $\mathcal{D}_m \leftarrow \emptyset$ for all $m \in \mathcal{M}$
Initialize seen signatures $\mathcal{S} \leftarrow \emptyset$
Define perturbation modes: `RE`, `LR`, `OF`, `DFS`, `R`$_{1\ldots3}$, `No COT`
while $|\mathcal{T}| < N$ **do**
  Sample source nodes $(Zorin, Vortex) \sim \text{Uniform}(0, 100)$
  Evaluate all other nodes in $\mathcal{G}$ via topological order
  Compute final target variable $Stardust$
  Create signature $\sigma = (Zorin, Vortex, Stardust)$
  if $\sigma \in \mathcal{S}$ **then**
    continue
  end if
  Add σ to $\mathcal{S}$
  Construct canonical chain-of-thought steps π
  Store canonical sample $(Q, \pi, Stardust)$ into test set $\mathcal{T}$
  for all $m \in \mathcal{M}$ **do**
    Apply perturbation m to steps $\pi \rightarrow \pi_m$
    Store perturbed sample $(Q, \pi_m, Stardust)$ into $\mathcal{D}_m$
  end for
end while
return $\mathcal{T}, \{\mathcal{D}_m\}_{m\in\mathcal{M}}$

Formally, the average cost for one token is computed as:

$$\text{Cost} = \frac{1}{865.2}(T_{\text{in}} \cdot P_{\text{in}} + T_{\text{out}} \cdot P_{\text{out}}),$$

where T_{in} and T_{out} denote the total input and output token counts. Thus, $T_{in} = 2846.2$, $T_{out} = 295.3$. The official pricing of GLM-4.5 is P_{in}=0.8 RMB/M tokens, P_{out}=2.0 RMB/M tokens.

Substituting the empirical statistics:

$$\text{Cost} = \frac{1}{865.2} \times (2846.2 \times 0.8 + 295.3 \times 2.0) = 3.31.$$

Converting to USD (1 RMB $\approx$ 0.14 USD)[§], the total annotation cost is about \$0.46 per million tokens.

[§]The pricing unit of the GLM API is in CNY, approximately 1 CNY per million tokens, which is equivalent to about 0.14 USD based on the exchange rate as of October 2, 2025.

E Attention Visualization of C^{2}DLM

This section provides a supplementary analysis of the attention visualizations presented in the main text. Specifically, the attention map illustrates how tokens interact within the model. However, examining the attention map alone is insufficient. For instance, in the phenomenon of attention sink, a disproportionate amount of attention is assigned to semantically insignificant tokens such as punctuation or prepositions. As compensation, the value matrix of these tokens exhibits substantially lower norms compared to normal tokens. Consequently, using value-weighted attention offers a more accurate visualization target. Our visualization results are shown in Figure 8 and 9. Based on the visualization, the following conclusions can be drawn:

Regardless of whether value weighting is applied, direct fine-tuning of LLaDA fails to effectively differentiate among the three types of factors, leading to a substantial decline in OOD performance. This occurs because the model primarily captures token correlations; when such correlations are disrupted in OOD settings, performance deteriorates significantly.

In contrast, C^{2}DLM effectively learns the underlying causal mechanisms of the model. As illustrated, the causal factors consistently exhibit greater importance than other factors, irrespective of whether value weighting is applied. Notably, although the Exercise attribute receives lower attention scores than Hormones in the raw attention map, its importance surpasses Hormones and ranks third once value weighting is considered. This highlights that weighted attention better captures the influence of value matrix norms across tokens, thereby providing a more faithful basis for analyzing token interactions.

F Human Evaluation of Teacher Model Generated Causal Graphs

Ideally, human experts should be employed for annotation. In this work, however, we use a teacher model to reduce costs and facilitate potential scaling. Accordingly, we conducted a human evaluation to verify the accuracy of the generated causal graphs. The evaluation procedure is detailed as follows:

We randomly sampled 50 annotated instances from the dataset, representing 7.3% of the total data. Two human experts with undergraduate degrees in science and engineering independently conducted the evaluation. During the evaluation, the experts were allowed to use any online resources to search and cross-check concepts to ensure the accuracy of their assessments.

For the causal graphs, given the absence of a pre-established ground truth, we focused on the causal logical consistency of each edge. Specifically, for each effect, we checked whether the list of causes extracted by the LLM could logically account for it. If the causal relationship was correct, it was scored as 1; if the cause list was partially correct or incomplete, it was scored as 0.5; if incorrect, it was scored as 0.

The evaluation metric is computed as follows. Let $E_i = \{(c_{i,j}, e_i)\}$ denote the set of causal pairs, where $c_{i,j}$ is a cause and e_i is the corresponding effect for instance i, and let score$(c_{i,j}, e_i)$ be the score assigned to each pair as described above. Then the accuracy for instance i is:

$$Acc_i = \frac{1}{|E_i|} \sum_{(c_{i,j}, e_i) \in E_i} \text{score}(c_{i,j}, e_i)$$

The overall causal accuracy across all N evaluated instances is:

$$\text{Overall Accuracy} = \frac{1}{N} \sum_{i=1}^{N} Acc_i$$

The experimental results are shown in Table 7, and can be summarized as follows:

1. Two instances failed to generate causal graphs due to decoding errors, accounting for 4% of the sampled data.

2. For instances with correctly decoded causal graphs, the accuracy is 93.42% $\pm$ 1.41%.

G Use of AI Assistants

We used generative AI, ChatGPT, to check for syntactic and grammatical errors in the manuscript. We carefully verified the correctness of the revised content.

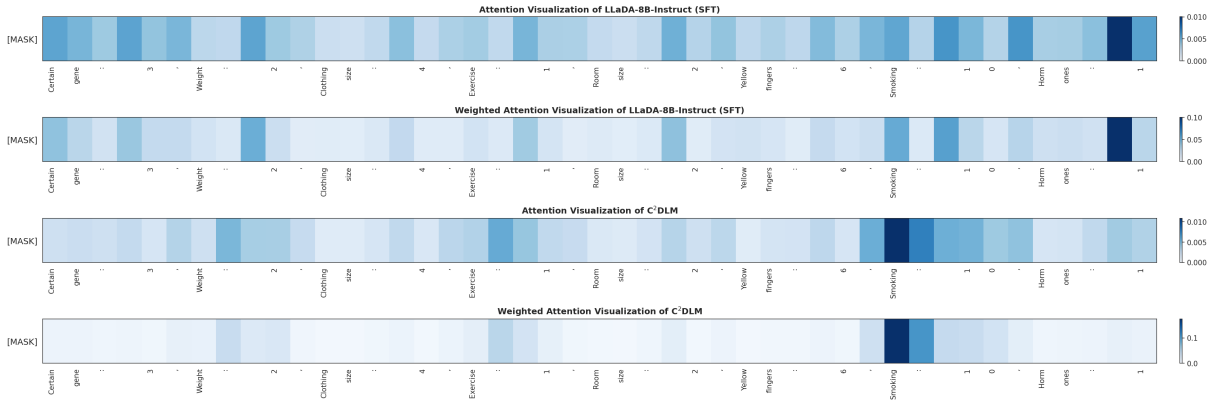

Figure 8: Mask is the position which will be decoded as high or low (directly determines the final answer). Visualization of the attention matrices, presented from top to bottom: direct visualization of LLaDA's attention map; LLaDA's attention map weighted by the Value norm; direct visualization of C^2DLM's attention map; and C^2DLM's attention map weighted by the Value norm.

Evaluator	Total Pairs	#Score=1	#Score=0.5	#Score=0	Average Accuracy
Human 1	311	280	28	3	94.84%
Human 2	311	271	29	11	92.02%

Table 7: Evaluation results for human assessment of causal graphs.

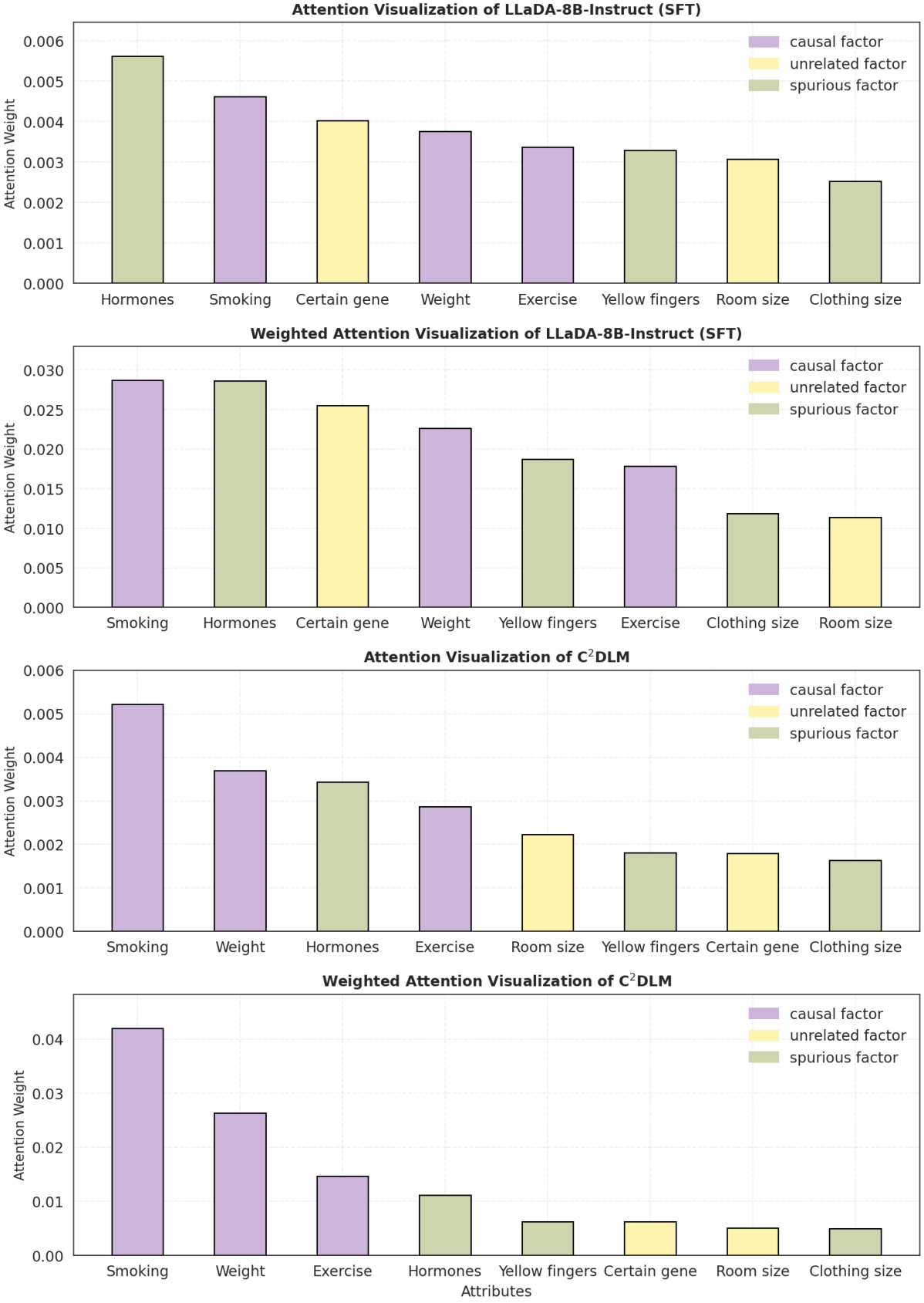

Figure 9: Visualization of the weight distribution bar charts, presented from top to bottom: direct visualization of LLaDA's attention weights bar chart; LLaDA's attention weights weighted by the Value norm; direct visualization of C^2DLM's attention weights bar chart; and C^2DLM's attention weights weighted by the Value norm.