

Verbalized Probabilistic Graphical Modeling

Hengguan Huang ^{*1} Xing Shen ^{*2} Songtao Wang ³ Lingfa Meng ¹ Dianbo Liu ³ Hao Wang ⁴ Samir Bhatt ¹

Abstract

Human cognition excels at transcending sensory input and forming latent representations that structure our understanding of the world. Although Large Language Models (LLMs) can produce chain-of-thought reasoning, they lack a principled framework to capture latent structures and model uncertainty, especially in compositional reasoning tasks. We propose Verbalized Probabilistic Graphical Modeling (vPGM), a Bayesian prompting framework that guides LLMs to simulate key principles of Probabilistic Graphical Models (PGMs) in natural language. Unlike many traditional probabilistic methods requiring substantial domain expertise or specialized training, vPGM bypasses expert-driven model design, making it well-suited for scenarios with limited assumptions or scarce data. We evaluated our model on several compositional reasoning tasks, both close-ended and open-ended. Our results indicate that the model effectively enhances confidence calibration and text generation quality.

1. Introduction

In addressing complex reasoning problems, such as solving challenging science questions, the human brain is thought to have the capability to go beyond mere sensory input, potentially forming insights into latent patterns of the world. This ability suggests that humans might have a sophisticated skill to interpret the underlying structures and uncertainties (Tenenbaum et al., 2011), although the exact mechanisms remain the subject of ongoing research and debate. As of now, such depth of understanding demonstrated by humans has not been fully achieved in artificial intelligence (AI) systems (Lake et al., 2017; Bender & Koller, 2020; Zheng et al., 2021; Sumers et al., 2023).

While large language models (LLMs) have demonstrated impressive capabilities in processing and generating human language (Devlin et al., 2018; Brown et al., 2020; Achiam

et al., 2023), their performance is often constrained by the scope of their training data. These models, built primarily on vast corpora of text, excel at generating responses that are syntactically coherent and contextually relevant. A notable advancement in LLMs is their ability to perform chain-of-thought (CoT) reasoning (Wei et al., 2022), which involves generating intermediate reasoning steps to arrive at a final answer. However, when faced with tasks that require an understanding of implicit knowledge, or the ability to integrate and reason with undisclosed information from multiple sources — skills that humans typically employ in complex reasoning — LLMs often struggle. This challenge arises not only from their reliance on explicit data patterns within their training data but also because LLMs lack a principled framework to capture latent structures and model uncertainty, especially in compositional reasoning tasks.

Aiming to address this from the LLM’s inference stage, we propose Verbalized Probabilistic Graphical Modeling (vPGM), a Bayesian prompting framework that guides LLMs to simulate key principles of Probabilistic Graphical Models (PGMs) in natural language. Unlike traditional Bayesian inference frameworks (Griffiths et al., 2008; Bielza & Larrañaga, 2014; Wang & Yeung, 2020; Abdullah et al., 2022), which typically require substantial domain expertise or specialized training, vPGM bypasses expert-driven model design, making it well-suited for scenarios with limited assumptions or scarce data. Specifically, Bayesian structure learning methods (Kitson et al., 2023) facilitate the discovery of Bayesian networks, they often require expert domain knowledge for manual validation of statistical dependencies or rely on computationally expensive scoring functions to assess the graphical model’s goodness of fit to the data. Our approach leverages the knowledge and reasoning capabilities of LLMs, employing Bayesian prompting to guide LLMs in simulating Bayesian reasoning principles, thus significantly reducing the reliance on data training and expert input.

Concretely, our method consists of three core stages: (1) Graphical Structure Discovery, in which the LLM is prompted to identify latent variables and their probabilistic dependencies; (2) Prompting- Based Inference, where LLMs are guided to infer verbalized posterior distributions of each latent variable given new input data; and (3) Predictions under Uncertainty, where confidence in the final

^{*}Equal contribution. ¹University of Copenhagen, Denmark.

²McGill University, Canada. ³National University of Singapore, Singapore. ⁴Rutgers University, USA. Correspondence to: Hengguan Huang hengguan.huang@sund.ku.dk.

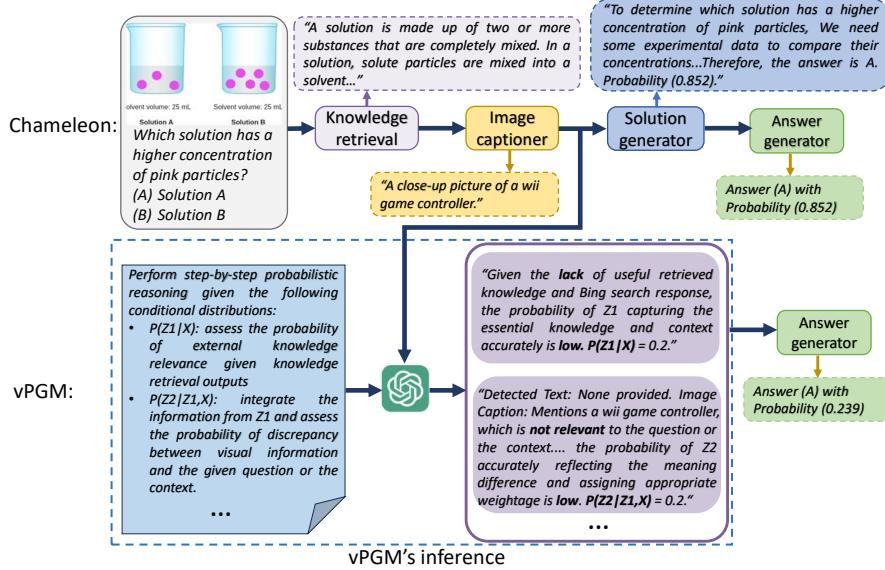


Figure 1: Example of inference using the vPGM with GPT-3.5. The Chameleon model erroneously assigns high confidence to the answer despite its LLM agents capturing irrelevant information. Conversely, our vPGM accurately identifies this discrepancy and assigns low confidence. Here, we show a simplified prompt for vPGM. See Appendix for a more detailed description in Table 8.

predictions is achieved by computing the expected value of the conditional predictive distribution over the inferred latent variables. Furthermore, to fully leverage the multiple response samples generated by LLMs within the vPGM framework and enhance uncertainty quantification, we extend vPGM with *numerical* Bayesian inference techniques that infer posterior distributions over predictions and augment confidence calibration through a differentiable calibration loss function.

We evaluate our method on several compositional reasoning tasks, designed in both close-ended and open-ended answering formats. The experiments demonstrate improvements in confidence calibration and the quality of generated responses, highlighting the efficacy of vPGM in enhancing probabilistic reasoning capabilities of LLMs.

2. Related Work

Prompting methods in Large Language Models (LLMs) represent a significant research domain, where the focus is on tailoring model responses for specific tasks. In this landscape, two prominent strategies have emerged: in-context learning (Brown et al., 2020), where models are provided with relevant task-specific examples, and instruction prompting (Wang et al., 2022b; Ouyang et al., 2022), which embed explicit task instructions within prompts.

A key development in this field is the Chain-of-Thought (CoT) prompting (Wei et al., 2022). This paradigm enhances complex reasoning in LLMs by incorporating a se-

ries of rationale steps within the prompting process. Building upon this, the zero-shot CoT approach (Kojima et al., 2022) extends CoT to handle tasks without exemplars or rationale steps. Further advancements include the automation of rationale chain generation (auto-CoT) (Zhang et al., 2022; Shum et al., 2023; Yao et al., 2024), and the self-consistency method (Wang et al., 2022a) for maintaining coherence across rationale steps, and chain-of-continuous-thought (Hao et al., 2024), which introduces a latent representation space for reasoning. Additionally, (Xiong et al., 2023) built upon the consistency-based method and conducted an empirical study on confidence elicitation for LLMs. In contrast, our proposed vPGM tackles the confidence elicitation problem from the perspective of Bayesian inference, which follows the principles of a more theoretically grounded Bayesian inference framework, PGM.

Closely related to our approach is ThinkSum (Ozturkler et al., 2022), which provides a two-step prompting method for probabilistic inference by operating over sets of objects or facts. Its “Think” stage retrieves associations, followed by a “Sum” stage that performs probabilistic inference. However, ThinkSum relies on carefully hand-crafted prompts for each question type, limiting its applicability in complex compositional reasoning scenarios where tasks vary widely in structure. It also faces difficulties when external tools are required, such as in our setting. In contrast, by integrating Bayesian principles from PGMs into our prompting strategy, we accommodate a broader range of reasoning tasks and more effectively capture the uncertainties and latent structures inherent in complex compositional reasoning

Table 1: This is an example prompt for PGM discovery in the context of compositional reasoning tasks.

Prompt for PGM Discovery in Solving Multiple-Choice Science Questions
Develop a Bayesian inference framework, denoted as $P(\mathbf{Y} \mathbf{Z})$, to achieve the task: { Task Description }. This entails identifying and defining a set of latent variables, $\mathbf{Z} = \{Z_1, Z_2, \dots, Z_n\}$, with $n \leq 4^1$. These variables should capture the decision-making process of a science expert evaluating an answer option.
Task Description: systematically determine the most probable answer \mathbf{Y} among a set of options for a given science question \mathbf{X} along with background context, such as image caption, OCR and relevant knowledge.
Context: {Context}
Input-Output Data Pairs: {Input-Output Data Pairs}
Prior Knowledge and Constraints: {Prior Knowledge and Constraints}

Table 2: An example response to the PGM discovery prompt generated by GPT-4, providing a list of latent variables along with their descriptions for PGM construction.

Discovered Latent Variables for PGM in Solving Multiple-Choice Science Questions
Discovered latent variables:
1. Z_1 Relevance Assessment: This variable quantifies the relevance of input data, including image captions, OCR results, and textual content, to the posed question. It encapsulates the conditional probabilities of the data being pertinent to understanding or answering the question effectively.
2. Z_2 Knowledge Quality Evaluation: Z_2 measures the reliability and adequacy of external knowledge sources. This variable assesses how well external data supports the interpretation of the question and the associated data, facilitating a Bayesian update of belief based on external evidence.
3. Z_3 Question Clarity: This variable evaluates the clarity and comprehensibility of the question. Z_3 captures the likelihood that the question can be clearly understood and processed to yield a definite outcome, influencing the interpretability and ease of response generation.
4. Z_4 Logical Reasoning: Z_4 is concerned with the logical analysis of each answer option. It involves a probabilistic assessment of the correctness of each option based on synthesized insights from the relevant data and external knowledge. This variable underpins the decision-making process by evaluating how logically coherent and supported each answer choice is given the available information.

scenarios, where ThinkSum fails to solve.

3. Background: Probabilistic Graphical Models in Bayesian Inference

Probabilistic Graphical Models (PGMs) are powerful tools for representing uncertainty and dependencies among variables (Koller & Friedman, 2009; Murphy, 2012). We focus on *Bayesian Networks* (BNs), a directed class of PGMs whose nodes correspond to random variables and whose edges encode conditional dependencies in a directed acyclic graph (DAG). Concretely, a BN over n latent variables $\mathbf{Z} = \{Z_1, \dots, Z_n\}$ factors their joint distribution as

$$P(\mathbf{Z}) = \prod_{i=1}^n P(Z_i \mid \text{Pa}(Z_i)), \quad (1)$$

where $\text{Pa}(Z_i)$ denotes the parent nodes of Z_i . Each term $P(Z_i \mid \text{Pa}(Z_i))$ is called a *conditional probability distribution* (CPD), and it specifies how a variable depends on its parents in the DAG.

Within the Bayesian paradigm, model parameters (i.e., of each CPD) are initially assigned with priors; as new data arrive, Bayesian inference refines these priors into posteriors, thereby capturing revised beliefs. However, designing a DAG and estimating its parameters can be challenging, especially when data are scarce or when domain expertise is limited. In this work, we overcome these constraints by leveraging Large Language Models (LLMs) to verbalize, discover, and perform inference in a simulated or verbalized Bayesian network without conventional data-intensive training or expert-defined structures, thus broadening the applicability of PGMs.

4. Our Method: Verbalized Probabilistic Graphical Modeling (vPGM)

Verbalized Probabilistic Graphical Modeling (vPGM) is a *Bayesian prompting* approach that leverages Large Language Models (LLMs) to simulate key principles of Probabilistic Graphical Models (PGMs) in natural language. Un-

¹Although we set $n \leq 4$ in this example, the LLM may generate the maximum number of variables. To reduce redundancies, we can add additional constraints to encourage a more compact representation.

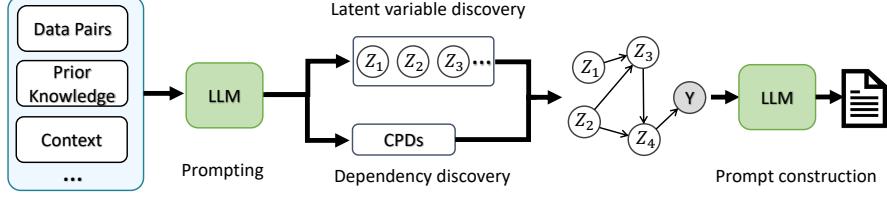


Figure 2: Overview of the vPGM’s learning framework. CPDs represent conditional probability distributions. The LLM in the figure refers to GPT-4, which is employed in the prompt construction step to adapt the resultant PGM into an inference prompt for GPT-3.5. We omit the observed variable \mathbf{X} for clarity.

Table 3: This shows an simplify example of obtained variable dependencies from the GPT-4. Each edge in the directed graph is presented as a condition distribution between distinct variables.

Exemplar Identified Dependencies of Latent Variables for PGM Construction

Identified dependencies of variables ($a \rightarrow b$ means b depends on a):

1. $\mathbf{X} \rightarrow Z_1, \mathbf{X} \rightarrow Z_2, \mathbf{X} \rightarrow Z_3, \mathbf{X} \rightarrow Z_4$
2. $Z_1 \rightarrow Z_3$
3. $Z_2 \rightarrow Z_3, Z_2 \rightarrow Z_4$
4. $Z_3 \rightarrow Z_4$
5. $Z_4 \rightarrow \mathbf{Y}$

like many existing probabilistic methods that demand extensive domain knowledge and specialized training, vPGM bypasses the need for expert-based model design, making it suitable for handling complex reasoning tasks where domain assumptions are limited or data are scarce.

4.1. Overview of vPGM

From an application standpoint, vPGM can be embedded into a range of complex reasoning systems, such as compositional reasoning tasks (see Figure 1). Our approach factorizes the overall reasoning process into three core steps: **(1) Graphical Structure Discovery**, in which the LLM is prompted to identify latent variables and their probabilistic dependencies (see Figure 2); **(2) Prompting-Based Inference**, where LLMs are guided to infer verbalized posterior distributions of each latent variable given new input data; and **(3) Predictions under Uncertainty**, where confidence in the final predictions is achieved by computing the expected value of the conditional predictive distribution over the inferred latent variables.

4.2. Graphical Structure Discovery

Our method begins by formulating a specialized prompt (see Table 1) to uncover latent variables for compositional reasoning. The prompt comprises several key elements: (1) General Task Description, a concise statement of the reasoning objective; (2) Input-Output Data Pairs, which illustrate representative data samples; (3) Contextual Information, providing any essential background or domain insights; and (4) Prior Knowledge and Constraints, specifying constraints

such as the maximum number of latent variables and predefined dependencies among them.

After identifying a set of latent variables $\mathbf{Z} = \{Z_1, Z_2, \dots, Z_n\}$ (see Table 2), we further prompt LLMs to determine how each latent variable depends on the others. An example of these dependencies is shown in Table 3, where each relationship $a \rightarrow b$ indicates that b is conditionally dependent on a . Like traditional PGMs, our verbalized PGM (vPGM) encodes these dependencies as conditional probability distributions $P(Z_i | \text{Pa}(Z_i))$. However, instead of relying on explicit distributional forms, vPGM uses natural language descriptions (see Table 9 in the Appendix) to specify each conditional relationship, reducing the need for extensive domain expertise or parameter estimation.

4.3. Prompting-based Bayesian Inference

Traditionally, Bayesian inference focuses on inferring posterior distributions over model parameters given a probabilistic model and new observations. In the context of LLMs, however, it is reformulated as generating prompts that simulate posterior inference under the vPGM framework, leveraging its discovered structure and new observations. This approach is reliant on, and leverages the advanced reasoning capabilities of LLMs (e.g., GPT-4) to produce instructions enabling a more cost-effective LLM (e.g., GPT-3.5) to simulate Bayesian inference. An example prompt is: *“Generate the prompt that guides GPT-3.5 through step-by-step probabilistic reasoning based on the provided task description, discovered PGM, and testing data...”*

4.4. Prediction under Uncertainty

Compositional reasoning tasks often involve significant uncertainty. For instance, an LLM agent (e.g., an image captioner) may produce noisy outputs, introducing aleatoric uncertainty. Under the vPGM framework, this variability is captured by the verbalized posterior distributions of latent variables. After constructing the verbalized posterior $P(\mathbf{Z} \mid \mathbf{X})$ via prompting-based Bayesian inference, we quantify confidence in the final predictions by taking the expected value of $P(\mathbf{Y} \mid \mathbf{Z})$ over \mathbf{Z} :

$$\mathbb{E}_{P(\mathbf{Z} \mid \mathbf{X})}[P(\mathbf{Y} \mid \mathbf{Z})] \approx \sum_{\mathbf{Z}} P(\mathbf{Y} \mid \mathbf{Z}) P(\mathbf{Z} \mid \mathbf{X}), \quad (2)$$

where \mathbf{X} denotes observed inputs, and \mathbf{Z} is sampled by querying LLM using vPGM’s Bayesian inference prompt. In practice, both $P(\mathbf{Z} \mid \mathbf{X})$ and $P(\mathbf{Y} \mid \mathbf{Z})$ are simulated within a single prompt (see Table 9 in the Appendix). Consequently, the expected posterior probabilities can be approximated by averaging the numerical values of $P(\mathbf{Y} \mid \mathbf{Z})$ generated by the LLM during these inference steps.

5. BayesVPGM: Bayesian-enhanced vPGM

When repeatedly querying a Large Language Model (LLM) under the vPGM framework, we obtain multiple samples of responses, i.e., categorical predictions and their numerical probabilities. A natural question is how to leverage these data to better capture the underlying uncertainty in the LLM’s predictions. To do this, we propose to infer such a posterior distribution, denoted $q(\mathbf{y} \mid \tilde{\mathbf{x}})$, where $\tilde{\mathbf{x}}$ denotes categorical predictions.

5.1. Posterior Inference under a Dirichlet Prior

We specify the form of the posterior $q(\mathbf{y} \mid \tilde{\mathbf{x}}) = \text{Cat}(\boldsymbol{\pi})$, where $\boldsymbol{\pi} = (\pi_1, \dots, \pi_K)$ lies in the probability simplex over K categories. To incorporate prior beliefs, we place a Dirichlet prior on $\boldsymbol{\pi}$:

$$\boldsymbol{\pi} \sim \text{Dirichlet}(\alpha_1, \dots, \alpha_K),$$

with $\alpha_k = \lambda p(y=k \mid \mathbf{Z})$ for some hyperparameter $\lambda > 0$, reflecting the vPGM’s initial belief in category k .

Next, suppose we query the LLM under the vPGM framework for n times, obtaining labels $\{y_1, \dots, y_n\}$. For each category k , let n_k be the number of labels that fall into that category. Assuming these labels are drawn i.i.d. from $\text{Cat}(\boldsymbol{\pi})$, the likelihood is

$$P(\{y_i\} \mid \boldsymbol{\pi}) = \prod_{k=1}^K \pi_k^{n_k}.$$

By Bayes’ rule, the posterior distribution is then

$$q(\mathbf{y} \mid \tilde{\mathbf{x}}) \propto \left(\prod_{k=1}^K \pi_k^{n_k} \right) \times \left(\prod_{k=1}^K \pi_k^{\alpha_k - 1} \right) = \prod_{k=1}^K \pi_k^{n_k + \alpha_k - 1},$$

i.e. a $\text{Dirichlet}(n_1 + \alpha_1, \dots, n_K + \alpha_K)$. The posterior mean of π_k becomes

$$\pi_k^{(\text{mean})} = \frac{n_k + \alpha_k}{\sum_{j=1}^K (n_j + \alpha_j)}.$$

Consequently, we adopt

$$q(\mathbf{y} \mid \tilde{\mathbf{x}}) = \text{Cat}(\boldsymbol{\pi}^{(\text{mean})})$$

as our final predictive distribution, which balances empirical label frequencies with the original vPGM’s numerical probabilities.

5.2. Optimizing λ via a Differentiable Calibration Loss

One key limitation of this posterior distribution is its reliance on a manually tuned λ , which governs how strongly the vPGM’s numerical probabilities influence the final outcome. To automate this process and improve calibration, we introduce a differentiable calibration loss that learns λ through gradient-based optimization.

Specifically, we minimize the following loss function with respect to λ :

$$\mathcal{L}(\boldsymbol{\pi}(\lambda)) = \mathcal{L}_c(\boldsymbol{\pi}(\lambda)) + \beta \mathcal{L}_v(\boldsymbol{\pi}(\lambda)), \quad (3)$$

where $\boldsymbol{\pi}(\lambda) = (\pi_1^{(\text{mean})}, \dots, \pi_K^{(\text{mean})})$ is the posterior-mean vector, \mathcal{L}_c is a standard classification loss (e.g., cross-entropy), and \mathcal{L}_v is a differentiable class-wise alignment term; β is a hyperparameter balancing the two losses. Let j index the categories, and let $\bar{\pi}_j = \frac{1}{n} \sum_{i=1}^n \pi_j^{(i)}$ be the average predicted probability of class j over a mini-batch of size n . Likewise, let $\bar{y}_j = \frac{1}{n} \sum_{i=1}^n y_j^{(i)}$ be the empirical fraction of class j , where $y_j^{(i)} \in \{0, 1\}$ indicates whether sample i belongs to class j . Inspired by class-wise expected calibration error (Kull et al., 2019), which aligns predictions to empirical frequencies on a per-category basis but whose binning procedure impedes differentiability, we define:

$$\mathcal{L}_v(\boldsymbol{\pi}) = \frac{1}{K} \sum_{j=1}^K |\bar{\pi}_j - \bar{y}_j|, \quad (4)$$

using a bin-free version of class-wise expected calibration error.

To minimize $\mathcal{L}(\boldsymbol{\pi})$ with respect to λ , we employ a quasi-Newton method (e.g. L-BFGS) (Broyden, 1967). This second-order gradient-based solver converges more rapidly than simple gradient descent.

6. Experiments

We evaluate the efficacy of the proposed vPGM and BayesVPGM in modeling uncertainty across three compositional reasoning tasks. The first, a closed-ended task named ScienceQA (Lu et al., 2022), and the second, an open-ended task named ChatCoach (Huang et al., 2024), both require reasoning with undisclosed information from multiple sources. We then introduce a negative control experiment derived from A-OKVQA (Schwenk et al., 2022) to investigate whether latent variables can enhance confidence calibration by detecting mismatches in the presence of misinformation. See Appendix for the more detailed experimental configurations.

6.1. Science Question Answering

The Science Question Answering (ScienceQA) benchmark, introduced by (Lu et al., 2022), serves as a comprehensive benchmark for multi-modal question answering across a diverse range of scientific disciplines, including physics, mathematics, biology, and the humanities. It features 4,241 question-answer pairs that cover various topics and contexts. This task demands the integration of information from multiple sources or LLM agents (e.g., Bing search results, image captions), a process that can introduce errors and increase the complexity of reasoning. Given these challenges, ScienceQA serves as an ideal testbed for evaluating how effectively vPGM identifies latent structures and model uncertainties. In this experiment, we use a vPGM with 2 latent variables for inference (see Table 10 in the Appendix for the inference prompt, and Table 8 for an example query). See Appendix for the more detailed data setups.

Baseline Methods We compare vPGM/BayesVPGM with the following baseline methods:

- **Chain-of-Thought** This is one of the non-tool-augmented LLMs: Chain-of-Thought (CoT) prompting (Wei et al., 2022) equipped with verbalized confidence estimation by prompting it to provide a numerical confidence for the selected answer.
- **Chameleon** This is based on a tool-augmented LLM: Chameleon (Lu et al., 2023), and we equip it with verbalized confidence estimation.
- **Chameleon+** It extends Chameleon with a state-of-art uncertainty quantification framework based on the combination of verbalized confidence estimation and self-consistency measurement (Wang et al., 2022a), as recommended in (Xiong et al., 2023).

Evaluation Metrics In line with previous evaluation settings in (Naeini et al., 2015; Guo et al., 2017; Xiong et al., 2023) on confidence calibration, we adopt the expected calibration error (**ECE**) to evaluate model confidence, represented as numeric probabilistic predictions. The ECE

quantifies the divergence between the predicted probabilities and the observed accuracy across each confidence levels (bins). Throughout our experiments, we fix the number of confidence bins as 10 with uniform confidence contribution across bins. In addition, we evaluate the capability of a given method in solving problems correctly by measuring the accuracy (**Acc.**).

Table 4: We report the accuracy and ECE for each method tested on ScienceQA. M represents number of sampled candidate responses, the verbalized confidence of these M responses is then averaged. The best-performing and the second-best-performing method for each metric is highlighted in **bold** and underlined, respectively.

Method	M	Acc. (%) \uparrow	ECE ($\times 10^2$) \downarrow
CoT	1	83.34	19.83
Chameleon	1	83.93	10.63
Chameleon+	3	81.97	10.74
vPGM (Ours)	3	84.39	<u>2.17</u>
BayesVPGM (Ours)	3	84.39	1.75

Results Table 4 details the performance of different methods on the ScienceQA dataset. It shows that CoT results in the highest (worst) ECE ($\times 10^2$) of 19.83, indicating serious overconfidence issues in handling complex reasoning tasks. In contrast, Chameleon substantially outperforms CoT in terms of ECE, suggesting that integrating external tools such as Bing search and advanced image captioners can improve confidence estimation. In comparison, our vPGM outperforms these methods in both accuracy and ECE, likely due to its superior ability to capture latent structural information that other baseline methods overlook. Figure 3 shows the reliability diagram for vPGM and BayesVPGM, demonstrating its near-perfect alignment with the ideal calibration curve across all bins, highlighting its precision in confidence calibration.

Qualitative Study on the Inferred Latent Variables Figure 1 shows a case study of vPGM’s inference capabilities to qualitatively assess the model’s ability to utilize latent structural information for improving confidence estimation. Here vPGM employs its latent variables to critically assess the relevance of retrieved information. For example, when faced with irrelevant data from external tools such as Bing search or inaccurate captions from image captioners, the baseline, Chameleon, erroneously maintains high confidence in its predictions. In contrast, vPGM carefully adjusts its confidence, assigning lower probabilities when essential contextual knowledge is missing or incorrect, a process that is particularly effective through the inference of latent variables Z_1 and Z_2 . These observations highlight the significance of inferring latent structures to improve the reliability of compositional reasoning systems. Moreover, due

Table 5: Results of various methods on the detection and correction of medical terminology errors.

Method	Detection			Correction		
	BLEU-2	Rouge-L	BERTScore	BLEU-2	Rouge-L	BERTScore
Instruction Prompting	27.4	3.3	67.6	1.4	2.1	61.6
Vanilla CoT	17.7	2.7	64.1	0.1	2.3	58.1
Zero-shot CoT	27.6	1.9	69.0	3.0	0.9	58.8
GCoT	34.2	3.7	72.4	1.6	2.0	65.4
vPGM (Ours)	37.2	2.3	76.3	1.7	2.0	68.3
Human	76.6	6.0	90.5	33.5	3.6	84.1

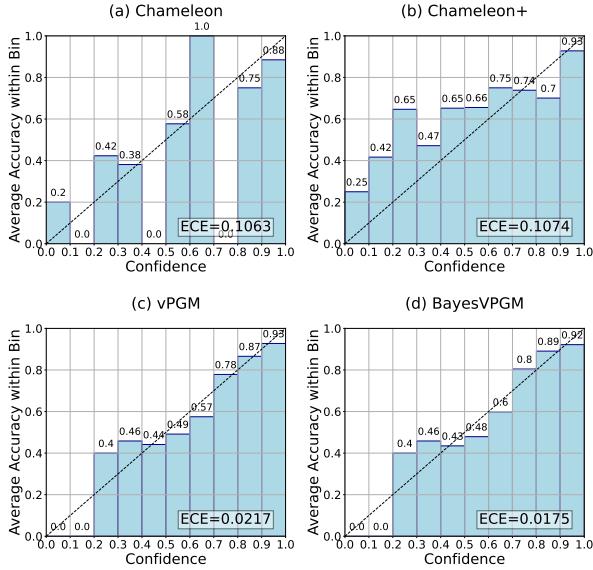


Figure 3: Reliability diagrams of (a) Chameleon, (b) Chameleon+, (c) vPGM, and (d) BayesVPGM on ScienceQA. vPGM and BayesVPGM achieve a much lower ECE comparing to Chameleon + Self-Random + Avg-Conf and approaches to the ideal confidence calibration curve (the diagonal dashed line).

to the natural language representation of the latent variables, vPGM also enhances system interpretability, explaining how predictions and associated confidences are derived.

6.2. Communicative Medical Coaching

The Communicative Medical Coaching benchmark, ChatCoach, introduced in (Huang et al., 2024), establishes a complex multi-agent dialogue scenario involving doctors, patients, and a medical coach across 3,500 conversation turns. The medical coach is tasked with detecting inaccuracies in medical terminology used by doctors (**detection task**) and suggesting appropriate corrections (**correction task**). These tasks require integrating external medical knowledge, inherently introducing uncertainty into response formulation. This benchmark was chosen to test vPGM’s ability

to generalize across complex open-ended reasoning tasks. BayesVPGM is not applied in this setting, as such a model assumes the output to be a categorical distribution. For more details on experiments and implementation, refer to the Appendix.

Baseline Methods For comparative analysis, we benchmark vPGM against these approaches:

- **Vanilla Instruction Prompting:** This method involves prompting the LLM with direct instructions for dialogue generation.
- **Zero-shot Chain of Thought (CoT) (Kojima et al., 2022):** A straightforward CoT approach where the LLM is prompted to sequentially articulate a reasoning chain.
- **Vanilla CoT (Wei et al., 2022):** This method builds upon the basic CoT by providing the LLM with a set of examples that include detailed reasoning steps.
- **Generalized CoT (GCoT) (Huang et al., 2024):** An advanced version of CoT, designed to improve the generation of structured feedback and integration of external knowledge effectively. It represents a state-of-the-art method in the ChatCoach benchmark.

Evaluation Metrics We follow (Huang et al., 2024) to employ conventional automated metrics **BLEU-2**, **ROUGE-L**, and **BERTScore**. BLEU-2 is employed to measure the precision of bi-gram overlaps, offering insights into the lexical accuracy of the generated text against reference answers. ROUGE-L is used to assess sentence-level similarity, focusing on the longest common subsequence to evaluate structural coherence and the alignment of sequential n-grams. Additionally, BERTScore is applied for a semantic similarity assessment, utilizing BERT embeddings to compare the generated outputs and reference texts on a deeper semantic level. As specified in (Huang et al., 2024), we use GPT-4 to extract medical terminology errors and corresponding corrections in the feedback from Coach Agents. Automated metrics are then calculated based on these extracted elements in comparison to human annotations.

Results We present the performance of various methods in Table 5. The noticeable difference between machine-generated outputs and human benchmarks across all metrics highlights the inherent challenges in communicative medi-

cal coaching. In the detection of medical terminology errors, vPGM leads with superior BLEU-2 (37.2) and BERTScore (76.3), underscoring its proficiency in identifying inaccuracies. In the correction task, while vPGM achieves a standout BERTScore of 68.3, surpassing all baselines, it scores lower on BLEU-2 and ROUGE-L. This variation is attributed to the ambiguity in doctors’ inputs, which can yield multiple valid responses, affecting metrics that rely on exact matches.

6.3. A-OKVQA Negative Control: Studying Latent Variables under Misinformation

Data Simulation A-OKVQA (Schwenk et al., 2022) is a Visual Question Answering dataset that challenges models to perform commonsense reasoning about a scene, often beyond the reach of simple knowledge-base queries. Crucially, it provides ground-truth image captions and rationales for each question. We leverage these annotations to construct a negative control experiment: **A-OKVQA-clean** (603 data points) retains the correct image caption and rationale (near single-hop reasoning), while **A-OKVQA-noisy** (603 data points) randomly shuffles the rationale, thus introducing misinformation and forcing a multi-hop check for consistency. In this experiment, we adopt a vPGM with two latent variables (see Table 12 for the inference prompt and Table 11 for an example query). Refer to the Appendix for more details on data configurations.

Overall Performance under Clean vs. Noisy Conditions. Table 6 shows the overall accuracy (Acc.) and expected calibration error (ECE) on both subsets. When the rationale is clean, Chameleon+ achieves lower ECE (2.75) than vPGM or BayesVPGM, reflecting that single-hop reasoning does not strongly benefit from latent structure. However, in the *Noisy* subset, both vPGM and BayesVPGM outperform Chameleon+ on accuracy (61.03% vs. 59.04%) and yield lower ECE, indicating that latent variables help detect mismatch and improve confidence calibration.

Table 6: General Performance on A-OKVQA-clean (Clean) vs. A-OKVQA-noisy (Noisy).

Method	Clean		Noisy	
	Acc.	ECE	Acc.	ECE
Chameleon+	95.02	2.75	59.04	11.75
vPGM (Ours)	95.02	5.56	61.03	10.54
BayesVPGM (Ours)	95.02	5.30	61.03	9.85

Mismatch Detection through Z_2 . To investigate how latent variables facilitate mismatch detection, we track $P(Z_2 | \text{Pa}(Z_2))$, where Z_2 indicates whether the rationale is aligned with the image caption. As shown in Table 7, the mean probability of Z_2 is considerably higher in the *Clean* set than in the *Noisy* set (0.86 vs. 0.42), and mismatch

identification accuracy in the *Noisy* condition reaches 87%. These findings demonstrate BayesVPGM’s capacity to robustly detect cases with inconsistencies or irrelevant content (i.e., cases with $Z_2 = 0$).

Latent Variable Correlation Analysis. We additionally compute Pearson correlations (Pcc.) between numerical conditional probabilities of the latent variables (Z_1 and Z_2) and the final answer \mathbf{Y} . In the *Noisy* case, $\text{Pcc}(Z_2, \mathbf{Y})$ surpasses $\text{Pcc}(Z_1, \mathbf{Y})$ (0.55 versus 0.35), indicating that Z_2 exerts a stronger influence on the final prediction when mismatches are present. Conversely, in the *Clean* subset, Z_1 and Z_2 exhibit nearly equal correlation with \mathbf{Y} , yet about 22% of the *Clean* data is incorrectly flagged by Z_2 as mismatched, potentially introducing noisy confidence adjustments at \mathbf{Y} and thereby increasing the overall ECE relative to **Chameleon+**. This suggests a trade-off: while latent variables excel at detecting misinformation in *Noisy* settings, they can slightly degrade calibration when no mismatch actually exists.

Table 7: Analysis of the latent variables on A-OKVQA-clean (Clean) and A-OKVQA-noisy (Noisy). Accuracy (Acc.) values are reported as fractions.

	Clean	Noisy
Mean $P(Z_2 \text{Pa}(Z_2))$	0.86	0.42
Noise Identification Acc.	0.78	0.87
$\text{Pcc}(Z_1, \mathbf{Y})$	0.50	0.35
$\text{Pcc}(Z_2, \mathbf{Y})$	0.51	0.55

7. Conclusion

We introduce verbalized Probabilistic Graphical Model (vPGM), a Bayesian prompting framework that directs Large Language Models (LLMs) to simulate core principles of Probabilistic Graphical Models (PGMs) through natural language. This approach discovers latent variables and dependencies without requiring extensive domain expertise or specialized training, making it well-suited to settings with limited assumptions or data. Our empirical results on compositional reasoning tasks demonstrate substantial improvements in terms of both confidence calibration and text generation quality. These results highlight the potential of merging Bayesian principles with LLMs to enhance AI systems’ capacity for modeling uncertainty and reasoning under uncertainty. While vPGM reduces the need for expert-driven model design, it still depends on prompt engineering and on the LLM’s ability to reliably interpret and execute Bayesian instructions. Future work could explore methods to automate prompt optimization, further enhancing the applicability of this approach across varied scenarios.

Impact Statement

This work's integration of Bayesian principles with Probabilistic Graphical Models (PGMs) into Large Language Models (LLMs) primarily enhances the reliability of AI in processing complex reasoning tasks. While the societal impacts may unfold gradually, the potential for these advancements to improve decision-making accuracy and reduce over-confidence issues in LLMs is significant. By fostering more reliable AI language models, this research aims to set a foundation for safer AI deployments, thereby contributing to the progress of AI technologies that societies and industries can confidently utilize.

References

- Abdullah, A. A., Hassan, M. M., and Mustafa, Y. T. A review on bayesian deep learning in healthcare: Applications and challenges. *IEEE Access*, 10:36538–36562, 2022.
- Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F. L., Almeida, D., Altenschmidt, J., Altman, S., Anadkat, S., et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.
- Bender, E. M. and Koller, A. Climbing towards nlu: On meaning, form, and understanding in the age of data. In *Proceedings of the 58th annual meeting of the association for computational linguistics*, pp. 5185–5198, 2020.
- Bielza, C. and Larrañaga, P. Bayesian networks in neuroscience: a survey. *Frontiers in computational neuroscience*, 8:131, 2014.
- Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33: 1877–1901, 2020.
- Broyden, C. G. Quasi-newton methods and their application to function minimisation. *Mathematics of Computation*, 21(99):368–381, 1967.
- Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.
- Griffiths, T. L., Kemp, C., and Tenenbaum, J. B. Bayesian models of cognition. In *Annual Meeting of the Cognitive Science Society, 2004; This chapter is based in part on tutorials given by the authors at the aforementioned conference as well as the one held in 2006*. Cambridge University Press, 2008.
- Guo, C., Pleiss, G., Sun, Y., and Weinberger, K. Q. On calibration of modern neural networks. In *International Conference on Machine Learning*, pp. 1321–1330. PMLR, 2017.
- Hao, S., Sukhbaatar, S., Su, D., Li, X., Hu, Z., Weston, J., and Tian, Y. Training large language models to reason in a continuous latent space. *arXiv preprint arXiv:2412.06769*, 2024.
- Huang, H., Wang, S., Liu, H., Wang, H., and Wang, Y. Benchmarking large language models on communicative medical coaching: a novel system and dataset. In *Findings of the Association for Computational Linguistics: ACL 2024*, 2024.
- Kitson, N. K., Constantinou, A. C., Guo, Z., Liu, Y., and Chobtham, K. A survey of bayesian network structure learning. *Artificial Intelligence Review*, 56(8):8721–8814, 2023.
- Kojima, T., Gu, S. S., Reid, M., Matsuo, Y., and Iwasawa, Y. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35: 22199–22213, 2022.
- Koller, D. and Friedman, N. *Probabilistic graphical models: principles and techniques*. MIT press, 2009.
- Kull, M., Perello Nieto, M., Kängsepp, M., Silva Filho, T., Song, H., and Flach, P. Beyond temperature scaling: Obtaining well-calibrated multi-class probabilities with dirichlet calibration. *Advances in neural information processing systems*, 32, 2019.
- Lake, B. M., Ullman, T. D., Tenenbaum, J. B., and Gershman, S. J. Building machines that learn and think like people. *Behavioral and brain sciences*, 40:e253, 2017.
- Lu, P., Mishra, S., Xia, T., Qiu, L., Chang, K.-W., Zhu, S.-C., Tafjord, O., Clark, P., and Kalyan, A. Learn to explain: Multimodal reasoning via thought chains for science question answering. *Advances in Neural Information Processing Systems*, 35:2507–2521, 2022.
- Lu, P., Peng, B., Cheng, H., Galley, M., Chang, K.-W., Wu, Y. N., Zhu, S.-C., and Gao, J. Chameleon: Plug-and-play compositional reasoning with large language models. *arXiv preprint arXiv:2304.09842*, 2023.
- Murphy, K. P. *Machine learning: a probabilistic perspective*. MIT press, 2012.
- Naeini, M. P., Cooper, G., and Hauskrecht, M. Obtaining well calibrated probabilities using bayesian binning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 29, 2015.

- Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., et al. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744, 2022.
- Ozturkler, B., Malkin, N., Wang, Z., and Jojic, N. Thinksum: Probabilistic reasoning over sets using large language models. *arXiv preprint arXiv:2210.01293*, 2022.
- Schwenk, D., Khandelwal, A., Clark, C., Marino, K., and Mottaghi, R. A-okvqa: A benchmark for visual question answering using world knowledge. In *European Conference on Computer Vision*, pp. 146–162. Springer, 2022.
- Shum, K., Diao, S., and Zhang, T. Automatic prompt augmentation and selection with chain-of-thought from labeled data. *arXiv preprint arXiv:2302.12822*, 2023.
- Sumers, T. R., Yao, S., Narasimhan, K., and Griffiths, T. L. Cognitive architectures for language agents. *arXiv preprint arXiv:2309.02427*, 2023.
- Tenenbaum, J. B., Kemp, C., Griffiths, T. L., and Goodman, N. D. How to grow a mind: Statistics, structure, and abstraction. *science*, 331(6022):1279–1285, 2011.
- Wang, H. and Yeung, D.-Y. A survey on bayesian deep learning. *ACM computing surveys (csur)*, 53(5):1–37, 2020.
- Wang, X., Wei, J., Schuurmans, D., Le, Q., Chi, E., Narang, S., Chowdhery, A., and Zhou, D. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*, 2022a.
- Wang, Y., Kordi, Y., Mishra, S., Liu, A., Smith, N. A., Khashabi, D., and Hajishirzi, H. Self-instruct: Aligning language model with self generated instructions. *arXiv preprint arXiv:2212.10560*, 2022b.
- Wei, J., Wang, X., Schuurmans, D., Bosma, M., Xia, F., Chi, E., Le, Q. V., Zhou, D., et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35:24824–24837, 2022.
- Xiong, M., Hu, Z., Lu, X., Li, Y., Fu, J., He, J., and Hooi, B. Can llms express their uncertainty? an empirical evaluation of confidence elicitation in llms. *arXiv preprint arXiv:2306.13063*, 2023.
- Yao, S., Yu, D., Zhao, J., Shafran, I., Griffiths, T., Cao, Y., and Narasimhan, K. Tree of thoughts: Deliberate problem solving with large language models. *Advances in Neural Information Processing Systems*, 36, 2024.
- Zhang, Z., Zhang, A., Li, M., and Smola, A. Automatic chain of thought prompting in large language models. *arXiv preprint arXiv:2210.03493*, 2022.
- Zheng, L., Guha, N., Anderson, B. R., Henderson, P., and Ho, D. E. When does pretraining help? assessing self-supervised learning for law and the casehold dataset of 53,000+ legal holdings. In *Proceedings of the eighteenth international conference on artificial intelligence and law*, pp. 159–168, 2021.

A. More Detailed Experiment Setup

LLM Configuration We use GPT-4 for PGM discovery and constructing Bayesian inference prompts for **vPGM**, while GPT-3.5-turbo-1106 serves as our test-time engine for all prompting-based methods. Unless otherwise specified, the temperature is fixed at 0.2. We generate three candidate responses for vPGM and BayesVPGM to estimate confidence.

A.1. Dataset

ScienceQA To accommodate **BayesVPGM** — which requires a development set to optimize the hyperparameter λ — we randomly sample 3568 data points from ScienceQA. Among these, 2563 form the test set, while the remaining 1005 comprise the development set used to tune λ .

A-OKVQA Negative Control For our A-OKVQA-based experiment, we include 1206 data points (both *clean* and *noisy* subsets) for testing and allocate 1005 data points to the development set for hyperparameter tuning.

ChatCoach Since **BayesVPGM** is not applied to ChatCoach (which produces open-ended, non-categorical outputs), we use the entire dataset for evaluation. Additionally, due to ChatGPT’s safety mechanisms, any prompts or responses flagged as potentially inappropriate are excluded from our reported results.

A.2. Training with BayesVPGM

We employ L-BFGS to optimize λ through the reparameterization $\tau = 1/\lambda$, thus constraining the search space.

ScienceQA We initialize τ to 2×10^{-5} , adopt a learning rate of 1×10^{-8} , and fix the maximum number of iterations at 1000.

A-OKVQA Negative Control We initialize τ to 2×10^{-12} , use a learning rate of 1×10^{-6} , and allow up to 20000 iterations.

A.3. Prompt

ScienceQA We provide a detailed example of inference using the vPGM, as shown in Table 8. Additionally, Table 10 demonstrates the prompt for a vPGM with 2 latent variables.

A-OKVQA Negative Control Here, we provide a detailed example of inference using the vPGM, as shown in Table 11. Additionally, Table 12 illustrates the inference prompt for vPGM with 2 latent variables.

ChatCoach We present the prompts used for each baseline approach: Instruction Prompting (see Table 13), Vanilla CoT (see Table 14), Zero-shot CoT (see Table 15), and GCoT (see Table 16). Table 17 shows the prompt of vPGM inference for ChatCoach.

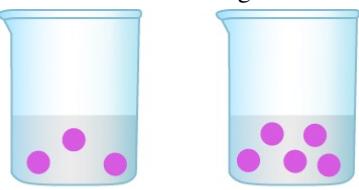
Question	<p>Which solution has a higher concentration of pink particles?</p>  <p>Solvent volume: 25 mL Solvent volume: 25 mL</p> <p>Solution A Solution B</p>
Answer options	<p>(A) Neither; their concentrations are the same ✗</p> <p>(B) Solution B ✓</p> <p>(C) Solution A ✗</p>
Image caption	A close up picture of a wii game controller.
Chameleon+	<p>To determine which solution has a higher concentration of pink particles, we need some experimental data to compare their concentrations...Therefore, the answer is A. Probability: 0.852</p> <p><i>(Continue to obtain 2 more samples)</i></p> <p>Final Answer: (A) ✗</p> <p>Marginalized Probability Across 3 Samples:</p> $P(\mathbf{Y} = \mathbf{A} \mathbf{X}) = 0.852$
vPGM	<p>Understanding Knowledge and Context (Variable Z_1):</p> <p>Comparing the concentration of pink particles in two solutions...Given the lack of retrieved knowledge and Bing search response, the probability of Z_1 capturing the essential knowledge and context accurately is low.</p> $P(Z_1 \mathbf{X}) = 0.200$ <p>Analyzing Textual and Visual Information (Variable Z_2):</p> <p>Detected Text: None provided. Image Caption: Mentions a wii game controller, which is not relevant to the question or the context...Given the lack of detected text and the irrelevant image caption, the probability of Z_2 accurately reflecting the meaning difference and assigning appropriate weightage is low.</p> $P(Z_2 Z_1, \mathbf{X}) = 0.200$ <p>Make a Decision with Confidence:</p> <p>Option A (neither): Not supported. Without any information about the concentration of pink particles in the solutions, it cannot be concluded...Option B (Solution B): Not supported. Without any information about the concentration of pink particles...Option C (Solution A): Not supported...Given the lack of information about the concentration of pink particles in the solutions, the confidence in each option being correct is as follows.</p> $P(\mathbf{Y} = \mathbf{A} Z_1, Z_2) = 0.333$ $P(\mathbf{Y} = \mathbf{B} Z_1, Z_2) = 0.333$ $P(\mathbf{Y} = \mathbf{C} Z_1, Z_2) = 0.333$ <p><i>(Continue to obtain 2 more samples)</i></p> <p>Final Answer: (B) ✓</p> <p>Marginalized Probability Across 3 Samples:</p> $P(\mathbf{Y} = \mathbf{A} Z_1, Z_2) = 0.239$ $P(\mathbf{Y} = \mathbf{B} Z_1, Z_2) = 0.522$ $P(\mathbf{Y} = \mathbf{C} Z_1, Z_2) = 0.239$

Table 8: ScienceQA case study example. vPGM captures discrepancies and produces a lower confidence comparing to Chameleon+.

Prompt for Inference of vPGM with 4 Latent Variables for ScienceQA

Using Bayesian inference, evaluate the probability of each answer option being correct. Consider step-by-step reasoning the following conditional distributions:

1. **Estimate $P(Z_1 | X)$:** Analyze the image caption, OCR results, and text data. Determine the relevance of each piece of data to the question.
 2. **Estimate $P(Z_2 | X)$:** Assess the provided external knowledge and determine its adequacy in interpreting the question and data.
 3. **Estimate $P(Z_3 | Z_1, Z_2, X)$:** Using the relevance of input data and the quality of external knowledge, evaluate how clear the question is.
 4. **Estimate $P(Z_4 | Z_2, Z_3, X)$:** Perform logical reasoning for answer option with input data and qualified knowledge.
 5. **Estimate $P(Y | Z_4)$:** Combine the evaluations to construct a probability distribution over the answer options.
-

Table 9: This is an example prompt for vPGM (with 4 latent variables) inference in the context of compositional reasoning tasks.

Prompt for Inference of vPGM with 2 Latent Variables for ScienceQA

Perform step-by-step probabilistic reasoning given the following conditional distributions:

1. **Estimate $P(Z_1 | X)$:** Begin by analyzing the question and its associated context. Consider the relevance of external knowledge derived from knowledge retrieval outputs to the posed question.
 2. **Estimate $P(Z_2 | Z_1, X)$:** Review the OCR results and image caption. Use the insights from Z_1 to evaluate the likelihood of discrepancies between the visual information and the textual context or question.
 3. **Estimate $P(Y | Z_1, Z_2)$:** Combine the assessments and reasoning derived from Z_1 and Z_2 . For each answer option, assign a probability reflecting your confidence in its correctness.
-

Table 10: This is an example prompt for vPGM (with 2 latent variables) inference in the context of compositional reasoning tasks.

Question	What setting to the animals shown here prefer? 
Answer options	(A) Desert ✗ (B) Tropical ✗ (C) Savannah ✓ (D) Tundra ✗
Image caption	Two giraffes and other animals at a zoo.
Rationale generator	The man is wearing a sailor hat.
Chameleon+	Identify the preferred setting for the animals shown. The question asks about the preferred setting for the animals shown in the image...Therefore, the answer is B . Probability: 0.250 (Continue to obtain 2 more samples) Final Answer: (B) ✗ Marginalized Probability Across 3 Samples: $P(\mathbf{Y} = \text{B} \mathbf{X}) = 0.425$
vPGM	Understanding Knowledge and Context (Variable Z_1): Identifying the preferred habitat for the animals shown in the image...Given the complete lack of relevant information in the rationale, the probability that Z_1 captures the essential knowledge and context required for answering the question correctly is very low. $P(Z_1 \mathbf{X}) = 0.050$ Analyzing Textual and Visual Information (Variable Z_2): The image caption mentions two giraffes and other animals at a zoo...Given the lack of specific details about the animals' preferred habitat in the image caption, the probability that Z_2 accurately reflects this information is low. $P(Z_2 Z_1, \mathbf{X}) = 0.150$ Make a Decision with Confidence: Options A, B, D: These options are not directly supported by the information available, and there is no clear indication of the animals' preferred habitat. Option C (Savannah): While not explicitly mentioned, giraffes are commonly associated with Savannah habitats. $P(\mathbf{Y} = \text{A} Z_1, Z_2) = 0.100$ $P(\mathbf{Y} = \text{B} Z_1, Z_2) = 0.100$ $P(\mathbf{Y} = \text{C} Z_1, Z_2) = 0.650$ $P(\mathbf{Y} = \text{D} Z_1, Z_2) = 0.150$ (Continue to obtain 2 more samples) Final Answer: (C) ✓ Marginalized Probability Across 3 Samples: $P(\mathbf{Y} = \text{A} Z_1, Z_2) = 0.237$ $P(\mathbf{Y} = \text{B} Z_1, Z_2) = 0.237$ $P(\mathbf{Y} = \text{C} Z_1, Z_2) = 0.285$ $P(\mathbf{Y} = \text{D} Z_1, Z_2) = 0.241$

Table 11: A-OKVQA case study example. vPGM captures discrepancies between the rationale and the question, hence it produces a lower confidence comparing to Chameleon+.

Prompt for Inference of vPGM with 2 Latent Variables for A-OKVQA

We have a question that requires a careful analysis to identify the correct answer. The decision-making process is structured into a series of steps, each focusing on specific aspects of the information provided. Let's approach this systematically:

1. **Estimate $P(Z_1 | X)$:** Start by analyzing the question and the provided context. What is the main topic, and what specific knowledge does it require? Consider the retrieved knowledge and the Bing search response. What essential information do these sources provide? Calculate the probability of Z_1 capturing the essential knowledge and context required for solving the question.
2. **Estimate $P(Z_2 | Z_1, X)$:** Examine the detected text in the image and the image caption. What are the key pieces of information each source provides? Are there any discrepancies between them? Estimate the probability of Z_2 accurately reflecting the meaning difference between detected text and image caption, and deciding the weight-age of each based on the discrepancy.
3. **Estimate $P(Y | Z_1, Z_2)$:** Integrate the evaluations and reasoning from Z_1 to Z_2 . For each answer option, provide a probability that represents your confidence in the option being correct. Ensure the probabilities sum up to 1. Proceed with the analysis, ensuring that each variable is logically derived from the provided information and the outcomes of the dependent variables.

Table 12: This is an example prompt for vPGM (with 2 latent variables) inference for A-OKVQA reasoning tasks.

Vanilla Instruction Prompting

Instruction: As a linguistic coach for a junior doctor, evaluate the doctor's statement: {doctor's statement} against the given medical context: {medical context}. If there are discrepancies, guide the doctor. If not, provide positive feedback.

Table 13: Instruction prompting for ChatCoach.

Vanilla Chain-of-thought

Instruction: As a linguistic coach for a junior doctor, evaluate the doctor's statement: {doctor's statement} against the given medical context: {medical context}. You should provide your response based on the following examples of input, thinking steps and output.

Example 1:

Input:

{doctor's statement for Example 1}
{medical context for Example 1}

Thinking steps:

{thinking steps for Example 1}

Output:

{coach's feedback for Example 1}

Example 2: {example2}

Example 3: {example3}

Input:

{doctor's statement}
{medical context}

Table 14: Vanilla CoT for ChatCoach.

Zero-shot Chain-of-thought

Instruction: As a linguistic coach for a junior doctor, evaluate the doctor's statement: {doctor's statement} against the given medical context: {medical context}. If there are discrepancies, guide the doctor. If not, provide positive feedback.

Please think step by step.

Table 15: Zero-shot CoT for ChatCoach

Generalized Chain-of-thought (GCoT)

Instruction: As a linguistic coach for a junior doctor, your task is to evaluate the doctor's statement: {doctor's statement} against the provided medical context: {Medical Context}. Your evaluation should identify any discrepancies within the doctor's communication. Where discrepancies arise, guide the doctor towards more accurate medical terminology and understanding. If the statements align well with the medical context, provide positive reinforcement and additional advice if necessary.

Thinking steps:

Identify Key Medical Terms:

Extract medical terms from the doctor's statement, including diseases, symptoms, medications, and treatments.

Compare with Medical Context:

Check these terms against the medical context for accuracy in:

- Disease/symptom identification.
- Medication/treatment recommendation.

Feedback:

- *If Incorrect:* Point out the error and provide the correct term from the medical context. Use simple corrections like "Instead of [incorrect symptom], it should be [correct symptom]", "Instead of [incorrect medication name], it should be [correct medication name]" or "Instead of [incorrect disease name], it should be [correct disease name]."

- *If Correct:* Affirm with "Your diagnosis/treatment aligns well with the medical context. Good job."

Note: `[correct symptom]`, `[correct medication name]` and `[correct disease name]` are extracted from medical context

Table 16: GCoT prompt for ChatCoach.

Prompt for vPGM inference for ChatCoach

Given the {doctor's statement} and the {medical context} provided:

Assess the Probability of Incorrect Terminology ($P(Z_1)$):

Analyze the medical terms used in the {doctor's statement}. Estimate the probability that any given medical term is used incorrectly based on the medical context.

If medical term is irrelevant to medical context then it was considered incorrect. List the medical terms along with their corresponding numerical probability of being incorrect.

Identify Specific Errors ($P(Z_2|Z_1)$):

For medical terms with a high probability of being incorrect, identify the specific term(s) that are used inappropriately. Provide a brief explanation for each identified error, referencing the {medical context}.

Determine Correction Requirement ($P(Z_3|Z_2, Z_1)$):

Based on the errors identified, decide if a correction is needed for each term. For each term that requires correction, state the appropriate medical term extracted from medical context that should be used. For each step, provide your reasoning and the associated probabilities (give real numbers ranging from 0 to 1), if applicable, to mimic the process of Bayesian inference.

Conclude by generating the coach feedback (in Chinese) that assesses the doctor's statement against a provided medical context and guides the physician by pointing out the particular medical terminology errors and providing the corresponding corrections if discrepancies arise, if no mistakes occurred, then encouraging the doctor and provide further medical advice.

Table 17: Prompt of vPGM inference for ChatCoach