How Faithfully do Language Models Acquire Probabilistic Knowledge?

Anonymous authors
Paper under double-blind review

Abstract

Many high-stakes decisions require calibrated distributions over plausible outcomes, not just top-ranked answers. LLMs are increasingly used to support probabilistic reasoning, yet it remains unclear how faithfully they acquire probabilistic knowledge during pretraining. We study this question in a controlled setting by generating synthetic Bayesian networks with known ground-truth distributions, verbalizing samples from these networks into text, and continually pretraining on the resulting corpora. We measure how well the models recover the underlying conditional distributions and the relations among them while varying network size, documentation policy, model scale, and training exposure. Models recover individual conditional probabilities closely, but the relationships among them exhibit what we term *relational compression*: models largely preserve which outcome is more probable and in which direction evidence moves a prediction, but they systematically underestimate the difference in likelihood between pairs of outcomes and how strongly evidence should shift a prediction.

1 Introduction

High-stakes decisions often require reasoning over the entire distribution of possible outcomes, not just the most likely outcome. In medicine, clinicians consider multiple possible diagnoses, and they implicitly revise their relative likelihoods as evidence accumulates. In finance, risk assessment depends on uncertainty and low-probability tail events. Large language models (LLMs) are increasingly used in settings that require this kind of probabilistic reasoning. Existing work has shown that they can estimate simple distributions, generate plausible priors, and make coarse probability comparisons (Paruchuri et al., 2024; Renda et al., 2026; Huang, 2025; Liu et al., 2024). As a result, LLMs are being incorporated into structured probabilistic inference and Bayesian-network-style estimation pipelines (Feng et al., 2024; Nafar et al., 2025). Despite this usage, it remains unclear what probabilistic knowledge LLMs acquire during pretraining and how faithfully they represent it. This includes not only individual probabilities, but also higher-order relationships among them, such as relative likelihoods and probability shifts under new evidence, especially as the underlying distributions grow more complex.

This question is especially relevant because empirical studies probing existing pretrained LLMs have identified several limitations in their probabilistic behavior. Models under-update from new evidence (Imran et al., 2025), exhibit biased priors that are not reliably corrected by explicit instructions (Gupta et al., 2025), and verbalize probabilities that are inconsistent with their own next-token distributions (Hu & Levy, 2023; Yamin et al., 2026). These limitations are especially consequential in high-stakes domains such as medicine. Clinical evaluations show that models can achieve high final-diagnosis accuracy while producing poor differential diagnoses (Rao et al., 2026), can overweight the likelihood of rare diseases relative to prevalence (Omar et al., 2025), and fail to appropriately update probabilities when presented with new evidence (McCoy et al., 2025). Probabilistic inference is further complicated when information is incomplete or ambiguous. For example, a symptom that is not mentioned in a patient note may be because it is absent, or it may have simply not been assessed. As a result, what is *not* said can also carry latent but noisy probabilistic information.

While these empirical findings raise major concerns about the probabilistic behavior of LLMs, they do not reveal why these failures arise. For example, the training data may provide distorted evidence about the underlying distribution, or models may fail to faithfully acquire complex probabilistic

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

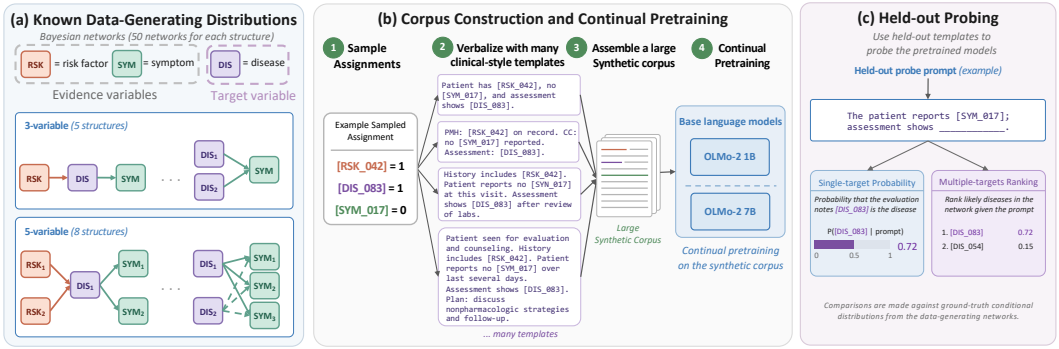


Figure 1: **High-level pipeline.** a) Known data-generating networks define ground-truth distributions. b) Sampled assignments are verbalized into a synthetic corpus for continual pretraining. c) Held-out templates probe target probabilities and rankings against ground-truth conditional references.

information from that data. However, it is difficult to disentangle these issues with natural corpora, since the distribution that generated the training data is unknown.

We therefore introduce a controlled framework in which we first construct Bayesian networks of synthetic entities with known probabilistic relationships (Fig. 1a). We then sample assignments from these networks, verbalize each assignment as a text snippet, assemble these snippets into a corpus, and then continually pretrain OLMo-2 models on the resulting corpus (Fig. 1b) (OLMo et al., 2024). Finally, after pretraining, we probe the models to see how well they recover conditional probabilities of the target variables given evidence in held-out prompts (Fig. 1c). Because we know the distributions that generated the training text, we can compute the exact conditional probability corresponding to each probe and compare it with the model's prediction. Importantly, the models are only ever exposed to the generated corpus and never observe the underlying networks or their probabilities.

Using this framework, we find that models recover individual conditional probabilities closely: across model scales and network sizes, their predictions deviate from the reference by 0.04–0.06 on average. While individual probabilities are relatively accurate, models struggle more with the difference in likelihood between candidate targets, the effect of introducing new evidence, and the interpretation of unmentioned findings. Models largely preserve the qualitative structure of these relations, such as which target is more probable and in which direction evidence moves a prediction, but systematically understate their magnitude. The gaps between candidate targets, in particular, are compressed more strongly than the individual probabilities themselves. We refer to this pattern as *relational compression*. Because aggregate error and ranking accuracy are largely insensitive to it, relational compression can go undetected under standard evaluation.

Our main contributions are as follows:

- We introduce a **controlled continual-pretraining framework** for measuring LLMs' acquisition of probabilistic knowledge from text, using synthetic generating distributions to provide exact ground truth knowledge both for individual conditional probabilities and for the relations among them.
- We show that strong acquisition of individual probabilities coexists with **relational compression** (§4.1): models rank candidate targets correctly and move predictions in the right direction, but understate the gaps between targets and the size of large updates.
- We show that relational compression extends to **what the text leaves unsaid** (§4.2): models learn how the documentation policy determines what an unmentioned finding implies, but shift their predictions only part of the way.

2 Related Work

Probabilistic reasoning and calibration in LLMs. Prior work has shown that pretrained LLMs exhibit some ability to reason about probabilities across a range of tasks. Models can answer some questions accurately about idealized distributions, but performance varies considerably across tasks and distribution families (Paruchuri et al., 2024). When asked instead to estimate distributions over

real-world quantities, their estimates can be inaccurate and poorly calibrated (Renda et al., 2026). Other work studies how models update beliefs as new evidence is provided. In controlled settings, these updates can be approximately Bayesian, although models may under-update or begin from biased priors (Gupta et al., 2025; Imran et al., 2025). Different ways of probing model behavior can also give different answers. Prompted judgments about linguistic patterns can diverge from conclusions based on the model's next-token probabilities (Hu & Levy, 2023). In decision settings, probabilities elicited from a model can also be imperfect summaries of the information reflected in its choices (Yamin et al., 2026). Probability judgments can furthermore be internally inconsistent across related questions (Zhu & Griffiths, 2024). Some work has sought to improve these probabilistic behaviors directly through fine-tuning (Qiu et al., 2026). These studies evaluate or modify trained models; we ask how faithfully pretraining acquires probabilistic information from text.

Learning and reasoning under incomplete information. Reasoning from incomplete information is hard because the meaning of an absence depends on why it occurred, and that reason is rarely recorded. For example, in clinical records, whether a measurement is taken can depend on a patient's condition, so the absence of a measurement may itself be predictive of downstream outcomes (Groenwold, 2020). In narrative text, a piece of information that is never mentioned may be absent, unassessed, or simply undocumented, as clinical records cannot mention every single symptom and risk factor (Weiskopf et al., 2013). Recent work suggests that LLMs do not always handle missing information reliably. Fu et al. (2025) show that even strong models struggle to identify content deliberately omitted from a text, even when given the original for comparison. Kobayashi et al. (2025) find LLMs do not natively make effective use of informative missingness in clinical prediction, although explicit instructions and in-context examples can improve their probabilistic predictions. While these studies probe existing trained models at inference time, we instead vary how the training corpus records absent and unobserved evidence to study learning and reasoning under incomplete information.

Knowledge acquisition during pretraining. A growing body of work studies how pretraining data shapes what models learn. One line of research relates model output to statistics of the pretraining corpus. For example, Wang et al. (2024) correlate model output probabilities with task-specific co-occurrence frequencies in the training data. Another line uses controlled experiments to study how individual facts are acquired, retained, and retrieved during pretraining (Chang et al., 2024; Kim et al., 2024; Allen-Zhu & Li, 2023). These settings provide corpus statistics or known answers for individual facts, but not a known distribution, so they do not test whether a model recovers the conditional probabilities underlying its training data.

3 FRAMEWORK

As shown in Figure 1, we use Bayesian networks (BNs) with known reference probabilities to generate text. We sample binary assignments, verbalize them as short snippets of text, and continually pretrain language models on the resulting corpus. Held-out prompts then test whether the model's continuation probabilities recover the corresponding conditional probabilities and the relations among them. We instantiate our domain-general framework using clinical note documentation, because clinical decision-making is probabilistic, high-stakes, and suitable for incomplete information settings. Clinical decision-making depends on distributions over target diseases, while evidence may be stated, negated, omitted, or never observed. Variables are synthetic tokens, and the generator-to-text mapping can be replaced with templates from other domains, such as finance or law.

3.1 CONTROLLED TEXT CORPORA FROM KNOWN DISTRIBUTIONS

Each Bayesian network contains binary variables assigned with different clinical roles: risk factors (RSK), diseases (DIS), and symptoms (SYM). To reduce reliance on prior semantic knowledge, each variable is mapped to a synthetic entity token such as `[RSK_042]`, `[DIS_083]`, or `[SYM_017]`. Edges follow role-compatible directions, with risk factors influencing diseases and diseases influencing symptoms. In this instantiation, diseases are the *target variables*, while risk factors and symptoms serve as *evidence variables* that may be stated, negated, or left unmentioned in the text.

We instantiate five 3-variable and eight 5-variable network topologies, covering both single-target and multi-target settings; an enumeration of topologies can be found in Figure 6. Each topology is instantiated as 50 networks, yielding 250 three-variable and 400 five-variable networks. Each

network has its own entity tokens, disjoint from those of all other networks. The parent priors and conditional-probability-table entries of each network are also sampled independently. Appendix A.1 provides an example network with its corresponding conditional probability tables. We draw 30,000 assignments from each network by ancestral sampling and verbalize each as a clinical note, using templates generated by GPT-5-mini in four language styles, from simple statements to rich narratives. A subset of templates is held out from training and split evenly between validation and test. We report results on held-out test templates, since knowledge acquired from the corpus should be expressible beyond the phrasings seen in training. Appendix B details template generation, and Table 6 compares recovery on training and held-out wording.

3.2 Documentation policies

A documentation policy determines whether each evidence variable is stated, negated, or omitted in the training text. Figure 2 illustrates the three policies using the same sampled assignment. Target (disease) variables are explicitly mentioned, e.g., `[DIS_i]` or `no [DIS_i]`, and only evidence variables (risk factors and symptoms) can be omitted.

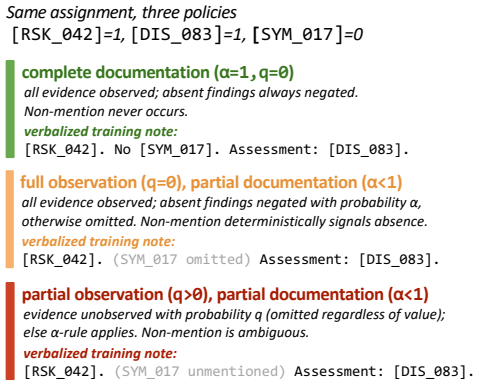


Figure 2: **Text realization policies.** The same sampled assignment is verbalized under three policies: complete documentation (§4.1), full observation with partial documentation (RQ4), and partial observation with partial documentation (RQ5).

We study two documentation regimes. Under *complete documentation* (§4.1), every evidence variable is observed and explicitly stated or negated. Under *incomplete documentation* (§4.2), an evidence variable may be omitted because the finding is negative but undocumented, or because it was never observed. This distinction is central: the same missing mention can correspond to a negative finding, unknown evidence, or a mixture of the two, depending on the training-time documentation policy. Under full observation with partial documentation (RQ4), all evidence is still observed, but an absent variable is explicitly negated with probability α and otherwise omitted, while present evidence is always stated. On the other hand, for partial observation with partial documentation (RQ5), each evidence variable is first unobserved with probability q, independently of its value, and omitted if unobserved. Observed variables then follow the same α rule (Appendix B.4).

3.3 Probing acquired distributions

Disease variables D_i are the probed *targets*. In complete-documentation probes, any other variable, including another disease, can supply evidence in the prefix $\mathcal{I}$. In the omission experiments, documentation policies act only on risk factors and symptoms; disease targets remain explicit. We evaluate held-out prompts that state the evidence information available at inference time, denoted by $\mathcal{I}$, and ask the model to continue with a disease token or its negation. We compare the resulting next-token probabilities to ground truth computed from the known generator. For complete-documentation probes, the reference conditions on the stated variables and marginalizes the rest. For incomplete documentation, the reference depends on the documentation policy (§4.2). Because the generator is known, we can also compute exact references for relations among these probabilities, such as the gap between two targets or the change in a target's probability when evidence is added.

Per-target probability probes. For a target disease D_i, we contrast the positive disease token with its explicit negation and define the normalized posterior estimate

$$p^{\text{norm}}(D_i{=}1 \mid \mathcal{I}) = \frac{P_{\text{LM}}(\texttt{[DIS_i]} \mid \mathcal{I})}{P_{\text{LM}}(\texttt{[DIS_i]} \mid \mathcal{I}) + P_{\text{LM}}(\texttt{no [DIS_i]} \mid \mathcal{I})}. \tag{1}$$

We evaluate against the ground-truth probability $P^\star(D_i{=}1 \mid \mathcal{I})$. Details on the computation are in Appendix D.2.

Multi-target ranking analysis. For networks with multiple target disease variables, we can additionally evaluate relative ranking. Targets are not assumed to be mutually exclusive: multiple target

variables may be simultaneously present in the ground-truth assignment. For each candidate disease target D_i, we produce $p^{\text{norm}}(D_i{=}1 \mid \mathcal{I})$ per Equation 1. We rank the candidate targets by their normalized scores p^{norm} and compare this ordering with that induced by the ground-truth posteriors $P^\star(D_i{=}1 \mid \mathcal{I})$. Thus, the multi-target ranking analysis evaluates conditional probability ranking and gap recovery given the evidence variables (Appendix D).

4 Experiments and Results

We organize our experiments around five research questions. Under complete documentation (§4.1), we first examine how faithfully models recover individual conditional probabilities (RQ1). We then ask whether models preserve the gaps between candidate targets under the same evidence (RQ2), and the change in a target's probability when evidence is added (RQ3). Under incomplete documentation (§4.2), we ask whether models learn how likely an unmentioned finding is to be absent rather than unobserved, while we vary the documentation policies (RQ4–5).

4.1 Probability recovery under complete documentation

We continually pretrain OLMo-2 1B and 7B separately on the three-variable and five-variable complete-documentation corpora, each containing 30,000 notes per network, and one epoch is one full pass over a corpus. Training notes mention all variables, including targets, in randomized order. For evaluation, we sample 10 networks per topology (50 three-variable and 80 five-variable networks) and probe them with held-out test templates.

For each disease target, we evaluate every binary assignment to every subset of the other variables, including the empty subset. We average p^{norm} (Eq. 1) across test templates and mention orders within each network, target, evidence assignment, and language style, and call each such combination a *query*, with averaged prediction $\hat{p}$ and reference $p^\star = P^\star(D{=}1 \mid \mathcal{I})$. We compare $\hat{p}$ with $p^\star$ using mean absolute error (MAE) and Pearson correlation r.

RQ1: How faithfully do models recover individual conditional probabilities?

Finding: Individual probabilities are recovered closely on average, with error concentrated on conditional probabilities given rare evidence.

The ability to faithfully recover conditional probabilities is a fundamental need in order to use language models as sources of probabilistic knowledge and for probabilistic reasoning. We begin by comparing model predictions with the reference probabilities across all probes. Both models recover these probabilities closely on both corpora, with MAE between 0.04 and 0.06 and Pearson correlation between 0.95 and 0.97 across the four configurations (Table 1). The effect of additional training epochs differs between the two corpora. On the three-variable corpus, recovery saturates within a few epochs: the two-epoch 1B run nearly matches the ten-epoch run, and 7B reaches similar performance after one and five epochs. On the five-variable corpus, convergence takes longer: gains diminish near the end of the ten-epoch 1B run, and 7B saturates after 5 epochs, with 7B reaching lower error than 1B (Appendix F.1, Figure 8).

Aggregate error averages over queries that differ both in how much evidence they state and in how often it occurs in the training corpus. We examine these two factors on the five-variable corpus, whose queries span conditioning levels L_0 to L_4, where L_k states k non-target variables. For example, with two risk factors and two symptoms as evidence, an L_1 prompt might read "History includes `[RSK_1]`; assessment shows", whereas an L_4 prompt states all four: "History includes `[RSK_1]` and no `[RSK_2]`; the patient reports `[SYM_1]` and no `[SYM_2]`; assessment shows".

Conditioning level. Most of the gains from further training on the five-variable corpus come from conditional queries: marginal probabilities already have low error after one epoch, whereas queries that condition on several variables continue to improve. Marginal queries (L_0) have low error from the first checkpoint onward, while conditional queries start with larger errors that decrease with training. By the final checkpoint, errors at L_2 through L_4 have converged to similar values, although a gap to marginal queries remains. Figure 3 shows this pattern within a single 7B run with full results in Appendix Table 5.

Table 1: **Models recover probabilities well, but with systematic compression.** MAE, r, and the per-target slope b use all targets; b_{pair} refits b on pair queries, and β (β_{correct}) is the gap slope on all (correctly ranked) pairs. Slopes below 1 indicate compression. This table shows 1B trained for ten epochs and 7B for five; other budgets are in Appendix F.1.

		Overall recovery		Per-target log-odds		Two-target ranking and gap		
Variables	Model	MAE ↓	r ↑	b	b_{pair}	Top-1 (%)	β	β_{correct}
3-var	1B	0.05	0.95	0.83	0.65	78.3	0.34	0.39
3-var	7B	0.04	0.97	0.84	0.71	87.5	0.50	0.51
5-var	1B	0.06	0.95	0.78	0.72	90.6	0.59	0.60
5-var	7B	0.05	0.96	0.92	0.88	92.8	0.77	0.78

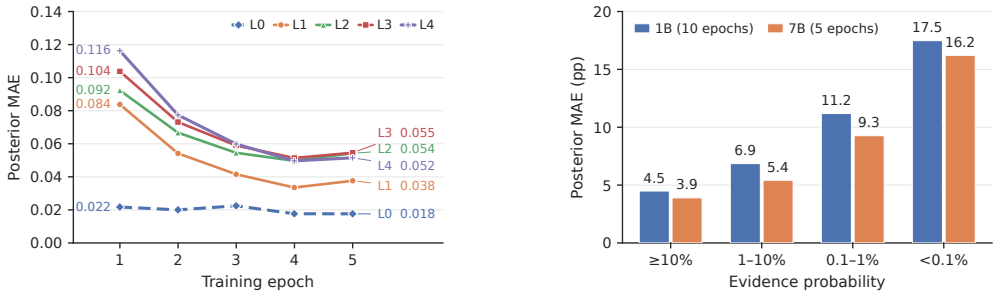
Evidence rarity. Conditioning level captures how much evidence a query states, but not how typical that evidence is. We therefore also group the conditional queries at L_1 through L_4 by $P^\star(E)$, the generator probability of the stated evidence assignment E, which determines how often that evidence appears in the training corpus. There is some overlap between the two groupings, since stating more variables generally makes an assignment rarer, but they capture different sources of difficulty. For both 1B and 7B, MAE increases from about 0.04 for the most common assignments ($P^\star(E) \geq 0.1$) to 0.16–0.18 for the rarest ($P^\star(E) < 0.001$) as shown in Figure 3(b). Additional training thus narrows the gap between conditioning levels, but conditional probabilities given rare evidence remain substantially less accurate. Low average error, moreover, does not establish that the relations among these probabilities are preserved, which we examine next.

RQ2: Do models preserve the magnitudes of probabilities and of the gaps between them?

Finding: Individual probabilities are mildly compressed on the log-odds scale. Across networks with multiple targets, models largely preserve the ranking but compress the gaps between targets beyond what per-target compression implies.

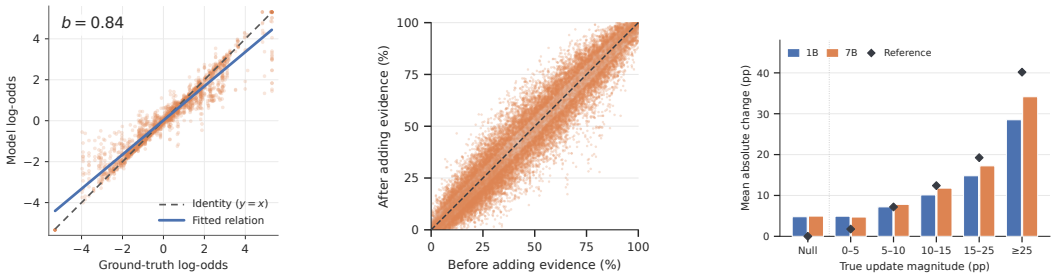
The small average error shown earlier can arise from random scatter around the reference or from a systematic distortion, and only the latter biases predictions in a consistent direction. We therefore test whether model probabilities are systematically compressed, first for individual targets and then for the gaps between candidate targets. The second directly probes *relational compression*: whether the relation between two probabilities is distorted more than the probabilities themselves. We find that the error is partly systematic and larger for relations than for individual probabilities.

Individual probabilities. We first ask whether model probabilities are as extreme as the reference. On the log-odds scale, which spreads out probabilities near 0 and 1, we fit $\text{logit}(\hat{p}) = a + b\ \text{logit}(p^\star)$ across all queries, where $0 < b < 1$ indicates compression and a captures an overall shift. Figure 4(a) shows this fit for three-variable 7B. Across the four configurations, b ranges from 0.78 to 0.92 (Table 1), indicating compression of individual log-odds after sufficient training. This compression is concentrated in multi-disease networks: b is lower there than in single-disease networks in every configuration, and single-disease slopes are close to one for 7B (Appendix F.3). This further motivates examining the gaps between candidate diseases directly.



(a) Conditioning level (b) Evidence rarity

Figure 3: **Recovery error by conditioning level and evidence rarity on five-variable corpora**. (a) MAE of 7B across checkpoints of the 5-epoch training under different conditioning levels, where L_k states k non-target variables. (b) MAE of both models for queries binned by the different generator probability $P^\star(E)$ of the stated evidence.



(a) Log-odds compression (b) Null contexts (c) Update magnitude

Figure 4: **Compression and evidence updates.** (a) Model versus reference log-odds for all test queries of three-variable 7B; the fitted slope $b = 0.84$ lies below the identity line. (b) Predictions of five-variable 7B before and after adding a variable in Null contexts, where the reference update is zero; shading marks ± 0.05. (c) Mean model update $|\hat{d}|$ (bars) and reference update $|d^\star|$ (diamonds) for five-variable models, binned by $|d^\star|$.

Gaps between diseases. A differential diagnosis depends not only on which disease is most probable, but also on how far apart the candidate diagnoses are. In two-disease networks, we pair the two diseases' queries that share the same non-disease evidence and state neither disease. Since the diseases can co-occur, the two probabilities need not sum to one. For each pair, Top-1 agreement records whether the model ranks the more probable disease first, and the gap $\Delta = \mathrm{logit}(p_1) - \mathrm{logit}(p_2)$ takes disease 1 to be the one with the higher reference probability. We fit $\hat{\Delta} = \beta\,\Delta^\star$ across pairs, so that $0 < \beta < 1$ indicates compressed gaps. To compare β with per-target compression on the same data, we also fit b only on the queries that form these pairs, which gives b_{pair}. If both diseases followed $\mathrm{logit}(\hat{p}) = a + b_{\mathrm{pair}}\,\mathrm{logit}(p^\star)$, the intercept would cancel in $\hat{\Delta}$, and β would equal b_{pair}. Models rank the diseases correctly in 78–93% of pairs, yet β is only 0.34–0.77. It falls below b_{pair} in every configuration (Table 1), so gaps are compressed beyond what per-target compression explains. Restricting the fit to correctly ranked pairs (β_{correct}) raises β by at most 0.05, so gaps remain compressed even when the ranking is correct. Models thus identify the more probable disease but understate how far apart candidate diseases are. This is relational compression in its most direct form, and ranking accuracy alone does not reveal it.

RQ3: Do models preserve how probabilities should change when evidence changes?

Finding: Models largely preserve the direction of evidence updates but understate the magnitude of large updates. They also shift predictions in response to conditionally independent findings, by amounts comparable to their response to weakly informative evidence.

Reasoning under uncertainty requires revising a target's probability as evidence accumulates, akin to when a clinician updates a differential diagnosis with each new finding. The update that a new finding induces is itself a relation between two probabilities, and a model can misestimate it even when each probability is close to the reference. We therefore ask whether models preserve both the direction and the magnitude of these updates, and whether they leave their predictions unchanged when the added finding is conditionally independent of the target. We use the five-variable corpus, whose queries span the widest range of conditioning levels. For a disease D, retained evidence E, and an added variable $X{=}x$, we compare the reference update $d^\star = p^\star(D{=}1 \mid E, X{=}x) - p^\star(D{=}1 \mid E)$ with the analogous model update $\hat{d}$.

Update magnitude. We group all one-variable evidence additions by $|d^\star|$ and compare the mean $|\hat{d}|$ with the mean $|d^\star|$ in each bin (Appendix F.4 gives the selection and counts). For reference updates between 0.05 and 0.10, the mean model update is close to the reference, but for updates of at least 0.25 it reaches only 0.70–0.85 of the reference. The direction of these large updates is nonetheless almost always correct, in 95.5% (1B) and 98.0% (7B) of additions. As with the gaps between targets in RQ2, models recover which way strong evidence pushes a prediction far better than how far it should move it, another instance of *relational compression*.

Invariance. Sometimes, added findings should leave the prediction unchanged. In Null contexts, the graph implies $D \perp X \mid E$, so $d^\star = 0$. Predictions nevertheless move by about 0.05 on average (mean $|\hat{d}|$) for both models, with a long tail of larger updates (Figure 4(b)). This is as large as the models' response to evidence that is genuinely but weakly informative: for reference updates below

0.05, the mean model update is also about 0.05 (Figure 4(c)). From the size of an update alone, one therefore cannot tell whether a finding is irrelevant or weakly relevant, and scaling from 1B to 7B does not reduce these spurious updates. Low aggregate error thus guarantees neither the evidential strengths nor the conditional independencies of the training distribution.

4.2 Learning the meaning of non-mention

A piece of information (e.g., a symptom) that goes unmentioned could either imply that symptom is absent, or just that the status of that symptom is unknown. Here we study how LLMs learn under incomplete documentation policies.

Documentation policies. Under the policies of §3.2, a present finding goes unmentioned only if it is unobserved (probability q), whereas an absent finding goes unmentioned both when it is unobserved (probability q) and when it is observed but not explicitly negated (probability $(1-q)(1-\alpha)$). At $q=0$, non-mention is therefore equivalent to explicit negation; for $q>0$, it becomes a weaker cue for absence as q grows. In training notes, diseases are always mentioned after all evidence variables, similar to how clinical notes present a diagnosis of disease following observed evidence.

Measuring the absence reading. For a disease D, stated evidence E, and an unmentioned evidence variable Y, let $g_{\text{marg}} = p^\star(D{=}1 \mid E)$ treat Y as uninformative and $g_{\text{abs}} = p^\star(D{=}1 \mid E, Y{=}0)$ treat it as absent. For each policy, we fit the *absence coefficient* by least squares over queries j,

$$\hat{p}_{\text{NM},j} - g_{\text{marg},j} = \widehat{\lambda}_{\text{model}}\big(g_{\text{abs},j} - g_{\text{marg},j}\big) + \epsilon_j, \tag{2}$$

which gives the fraction of the absence shift the model applies: 0 if it marginalizes over Y and 1 if it reads non-mention as explicit absence. The correct coefficient depends on the policy. We obtain it by fitting the exact posterior $g_{\text{NM}} = p^\star(D{=}1 \mid E, \text{NM})$, which follows from the BN and the policy by Bayes' rule (Appendix E), in place of the model predictions. The resulting reference coefficient $\widehat{\lambda}^\star$ equals 1 at $q=0$ and decreases as q grows.

Setup. We train separate three-variable 1B models for two epochs under each policy with $\alpha \in \{0, 0.25, 0.5, 0.75, 1\}$ and $q \in \{0, 0.1, 0.25, 0.5, 0.75\}$, excluding $\alpha = 1$, $q = 0$, which never produces non-mention evidence. Each query states one positive finding and leaves one evidence variable unmentioned. We only keep queries with $|g_{\text{abs}} - g_{\text{marg}}| \geq 0.05$, so that the two readings are distinguishable. This yields 248 queries from 42 BNs, shared across policies (Appendix G).

RQ4: Do models treat non-mention as equivalent to explicit absence?

Finding: When non-mention certainly means absence, models apply only about three quarters of the implied shift, and respond differently to non-mention than to an equivalent explicit negation.

At $q=0$, non-mention is a certain sign of absence for every $\alpha < 1$, so $\widehat{\lambda}^\star = 1$ regardless of α. Varying α nonetheless changes how often non-mention occurs: an absent finding goes unmentioned with probability $1-\alpha$, so from $\alpha=0$ to $\alpha=0.75$ non-mention becomes four times rarer while keeping the same meaning. This setting separates what non-mention means from its frequency.

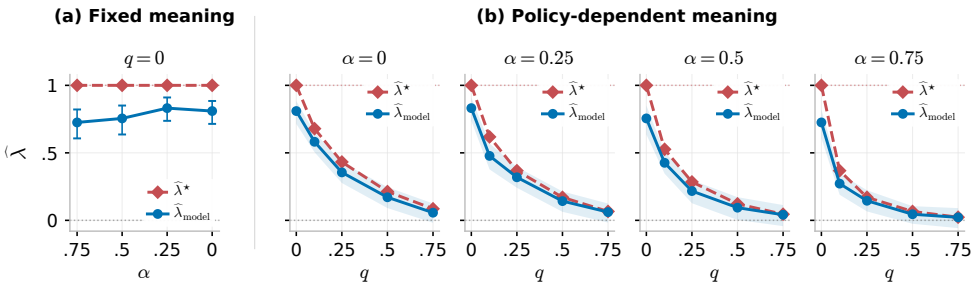


Figure 5: **Models learn what non-mention implies but apply a compressed shift.** The two BN references anchor the vertical scale at zero (marginalization) and one (explicit absence). (a) Varying α at $q=0$ changes omission frequency while preserving its meaning as absence. (b) Varying q at fixed α weakens the absence cue. All panels use the same 248 test points from three-variable 1B models trained for two epochs. Error bars and shading give 95% BN-bootstrap intervals.

Agreement with the reference. Models read non-mention as absence, but only in part. Their coefficients lie between 0.73 and 0.83 at every α (Figure 5(a)), so they apply roughly three-quarters of the shift that explicit absence would produce. Although non-mention becomes four times rarer from $\alpha = 0$ to $\alpha = 0.75$, the coefficient varies only modestly, and is slightly lower where non-mention is rarest: models extract its meaning nearly as well from a corpus in which it is rare as from one in which it is common. This holds even at $\alpha = 0$, where no evidence variable is ever negated and non-mention is the only way the corpus expresses absence.

Agreement with explicit negation. Equivalence also requires the model's own predictions to agree across the two descriptions. For $\alpha > 0$, where negated evidence appears in training, we compare $\hat{p}_{\mathrm{NM}}$ with the same model's prediction $\hat{p}_0$ when the prompt negates Y explicitly, matching BN, disease, remaining evidence, and style. The two differ by about 0.11 on average, more than twice the average recovery error of three-variable 1B under complete documentation, although under the training policy they carry exactly the same information. Models thus learn that non-mention signals absence, yet they do not treat it as interchangeable with stating absence.

RQ5: Do models treat non-mention as weaker evidence of absence when findings may go unobserved?

Finding: As more findings go unobserved, models weaken their response to non-mention along with the reference, but still apply only 73–87% of the reference shift wherever that shift is non-negligible.

When $q > 0$, a present finding can also go unmentioned, so the absence cue weakens and $\widehat{\lambda}^{\star}$ decreases with q. No note marks a finding as unobserved, so a model can recover this weakening only from how non-mention co-occurs with the other variables across the corpus. Figure 5(b) varies q at each $\alpha < 1$.

Tracking the cue. Along all four curves, the model coefficient decreases with q, following the reference coefficient. At $\alpha = 1$, every observed absence is negated, so non-mention indicates only a missing observation and $\widehat{\lambda}^{\star} = 0$; model coefficients stay between -0.02 and 0.02 for every $q > 0$ (Table 9). The models thus infer, from corpus statistics alone, that non-mention becomes weaker evidence of absence once observations can go missing.

Compression persists. The gap between model and reference is largest at $q = 0$ and narrows as q grows; by $q = 0.5$, the two differ by at most 0.05 at every α. This narrowing, however, largely reflects the shrinking reference shift rather than improved recovery. Measured relative to the reference, models apply 73–87% of the reference shift across the 15 policies where that shift is non-negligible ($\widehat{\lambda}^{\star} \geq 0.1$; Table 10), a fraction that is stable across these policies. Non-mention thus exhibits the same *relational compression* as explicit evidence: models learn how likely an unmentioned finding is to be absent, but apply only part of the implied shift.

5 Discussion and Conclusion

We studied how faithfully language models acquire probabilistic knowledge from text generated by known distributions. Models recover individual conditional probabilities closely but exhibit *relational compression*: they preserve which target is more probable and which way evidence moves a prediction, while underestimating the gaps between targets, the size of strong updates, and the shift implied by non-mention. Because aggregate error and ranking accuracy are largely insensitive to this pattern, evaluations of probabilistic knowledge should test the relations among probabilities directly. Since our training text is sampled without bias from the known distribution, these results also suggest that part of the under-updating reported for pretrained LLMs (Imran et al., 2025; McCoy et al., 2025) may arise when acquired probabilistic knowledge is expressed beyond the training wording (Table 6), although establishing this for natural corpora remains open.

Limitations and future work. Our controlled setting uses synthetic text so that the underlying probabilistic relationships are known exactly. This allows us to isolate relational compression, but leaves open how strongly the effect extends to natural text. Because we study base models before post-training, future work could also examine whether instruction tuning or RLHF amplifies or corrects the effect. More broadly, this work opens a line of inquiry into how faithfully language models acquire probabilistic relationships as the underlying system becomes more complex, including larger and more densely connected networks, nonbinary variables, richer forms of evidence, and other model families.

AI USE STATEMENT

We used generative AI tools for language editing and LaTeX formatting assistance. As part of the method, a language model (GPT-5-mini) generated the note templates used to verbalize synthetic samples (Appendix B); all variable assignments and reference probabilities come from our own generators. We did not use generative AI tools to generate experimental results, fabricate data, or replace author verification of the methods and claims. All AI-assisted outputs were reviewed by the authors, who take responsibility for the final content of this work.

ETHICS STATEMENT

This work uses only synthetic data: variables are synthetic entity tokens sampled from Bayesian networks, and the clinical setting serves only as a template domain. The study involves no human subjects, real patient records, or personally identifying information. The trained models are research artifacts and are not intended for clinical use. We follow the ICLR Code of Ethics.

REPRODUCIBILITY STATEMENT

Section 3 describes the data generators, documentation policies, and probing procedure, and Section 4 describes the experimental setup. Network structures, templates, training configuration, probing and metric details, and the derivation of the omission references are given in Appendices A–E. All experiments start from publicly available OLMo-2 base checkpoints. We plan to release the code for corpus generation, training, and evaluation.

REFERENCES

Zeyuan Allen-Zhu and Yuanzhi Li. Physics of language models: Part 3.1, knowledge storage and extraction. *arXiv preprint arXiv:2309.14316*, 2023.

Hoyeon Chang, Jinho Park, Seonghyeon Ye, Sohee Yang, Youngkyung Seo, Du-Seong Chang, and Minjoon Seo. How do large language models acquire factual knowledge during pretraining? *Advances in neural information processing systems*, 37:60626–60668, 2024.

Yu Feng, Ben Zhou, Weidong Lin, and Dan Roth. Bird: A trustworthy bayesian inference framework for large language models. *arXiv preprint arXiv:2404.12494*, 2024.

Harvey Yiyun Fu, Aryan Shrivastava, Jared Moore, Peter West, Chenhao Tan, and Ari Holtzman. Absencebench: Language models can't tell what's missing. *arXiv preprint arXiv:2506.11440*, 2025.

Rolf HH Groenwold. Informative missingness in electronic health record systems: the curse of knowing. *Diagnostic and prognostic research*, 4(1):8, 2020.

Ritwik Gupta, Rodolfo Corona, Jiaxin Ge, Eric Wang, Dan Klein, Trevor Darrell, and David M Chan. Enough coin flips can make llms act bayesian. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 7634–7655, 2025.

Jennifer Hu and Roger Levy. Prompting is not a substitute for probability measurements in large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 5040–5060, 2023.

Yongchao Huang. Llm-bi: Towards fully automated bayesian inference with large language models. *arXiv preprint arXiv:2508.08300*, 2025.

Sohaib Imran, Ihor Kendiukhov, Matthew Broerman, Aditya Thomas, Riccardo Campanella, Rob Lamb, and Peter M Atkinson. Are llm belief updates consistent with bayes' theorem? *arXiv preprint arXiv:2507.17951*, 2025.

Jiyeon Kim, Hyunji Lee, Hyowon Cho, Joel Jang, Hyeonbin Hwang, Seungpil Won, Youbin Ahn, Dohaeng Lee, and Minjoon Seo. Knowledge entropy decay during language model pretraining hinders new knowledge acquisition. *arXiv preprint arXiv:2410.01380*, 2024.

Yuta Kobayashi, Vincent Jeanselme, and Shalmali Joshi. Aligning probabilistic beliefs under informative missingness: Llm steerability in clinical reasoning. *arXiv preprint arXiv:2512.00479*, 2025.

Tennison Liu, Nicolás Astorga, Nabeel Seedat, and Mihaela van der Schaar. Large language models to enhance bayesian optimization. *arXiv preprint arXiv:2402.03921*, 2024.

Liam G McCoy, Rajiv Swamy, Nidhish Sagar, Minjia Wang, Stephen Bacchi, Jie Ming Nigel Fong, Nigel CK Tan, Kevin Tan, Thomas A Buckley, Peter Brodeur, et al. Assessment of large language models in clinical reasoning: a novel benchmarking study. *NEJM AI*, 2(10):AIdbp2500120, 2025.

Aliakbar Nafar, Kristen Brent Venable, Zijun Cui, and Parisa Kordjamshidi. Extracting probabilistic knowledge from large language models for bayesian network parameterization. *arXiv preprint arXiv:2505.15918*, 2025.

Team OLMo, Pete Walsh, Luca Soldaini, Dirk Groeneveld, Kyle Lo, Shane Arora, Akshita Bhagia, Yuling Gu, Shengyi Huang, Matt Jordan, et al. 2 olmo 2 furious. *arXiv preprint arXiv:2501.00656*, 2024.

Mahmud Omar, Reem Agbareia, Alon Gorenshtein, and Girish N Nadkarni. Large language models chase zebras: Salient cues overrule base rates in clinical diagnosis. *Available at SSRN 5988435*, 2025.

Akshay Paruchuri, Jake Garrison, Shun Liao, John B Hernandez, Jacob Sunshine, Tim Althoff, Xin Liu, and Daniel McDuff. What are the odds? language models are capable of probabilistic reasoning. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 11712–11733, 2024.

Linlu Qiu, Fei Sha, Kelsey Allen, Yoon Kim, Tal Linzen, and Sjoerd van Steenkiste. Bayesian teaching enables probabilistic reasoning in large language models. *Nature Communications*, 2026.

Arya S Rao, Kaiz P Esmail, Richard S Lee, Sharon Jiang, Bianca Arraiza Carlo, Jasleen Gill, Praneet Khanna, Ezra Kalmowitz, Basile Montagnese, Kimia Heydari, et al. Large language model performance and clinical reasoning tasks. *JAMA Network Open*, 9(4):e264003, 2026.

Alana Renda, Jillian Ross, and Jacob Andreas. Openestimate: Evaluating llms on reasoning under uncertainty with real-world data. In *International Conference on Learning Representations*, volume 2026, pp. 2350–2368, 2026.

Xinyi Wang, Antonis Antoniades, Yanai Elazar, Alfonso Amayuelas, Alon Albalak, Kexun Zhang, and William Yang Wang. Generalization vs memorization: Tracing language models' capabilities back to pretraining data. *arXiv preprint arXiv:2407.14985*, 2024.

Nicole G Weiskopf, George Hripcsak, Sushmita Swaminathan, and Chunhua Weng. Defining and measuring completeness of electronic health records for secondary use. *Journal of biomedical informatics*, 46(5):830–836, 2013.

Khurram Yamin, Jingjing Tang, Santiago Cortes-Gomez, Amit Sharma, Eric Horvitz, and Bryan Wilder. Do llms act like rational agents? measuring belief coherence in probabilistic decision making. *arXiv preprint arXiv:2602.06286*, 2026.

Jian-Qiao Zhu and Thomas L Griffiths. Incoherent probability judgments in large language models. *arXiv preprint arXiv:2401.16646*, 2024.

APPENDIX

A GENERATOR AND NETWORK STRUCTURES

Figure 6 enumerates the 13 Bayesian network topologies used in the experiments. All variables are binary and assigned one of three roles: risk factors (RSK), diseases (DIS), or symptoms (SYM). Edges are restricted to the role-compatible directions RSK $\rightarrow$ DIS and DIS $\rightarrow$ SYM. Each topology is instantiated as 50 independent networks with independently sampled priors and conditional probability tables. Entity tokens are unique to each network.

The 3-variable structures instantiate all role-compatible chain, fork, and collider motifs. The 5-variable structures cover larger diagnosis-oriented patterns: single-disease evidence accumulation, overlapping symptoms, competing diseases, risk-based disambiguation, mixed evidence, and comorbidity.

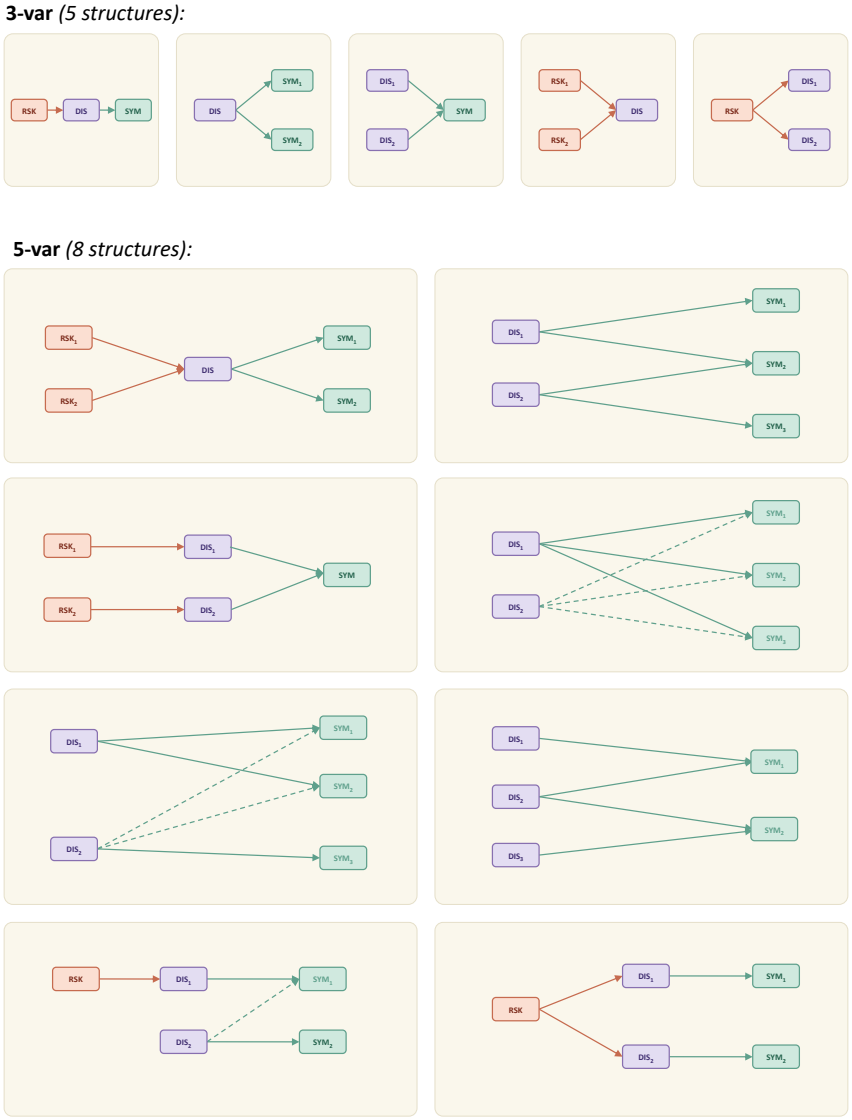


Figure 6: **Structure catalogue.**

A.1 EXAMPLE NETWORK

To make the catalogue concrete, Figure 7 sketches a single-DIS network instance and Table 2 lists its conditional probability tables. The DAG follows the role-compatible direction RSK $\rightarrow$ DIS $\rightarrow$ SYM: two risk factors are independent parents of the disease, which in turn is the parent of two symptoms. CPT entries and priors are sampled independently per network so that no two BNs share parameter values.

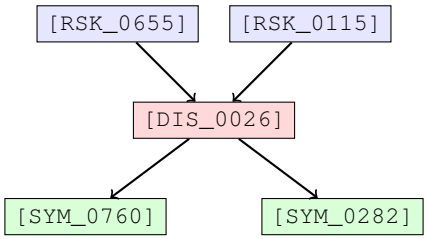


Figure 7: **Single-DIS Example Network.** Two risk factors RSK feed a single disease DIS, which feeds two symptoms SYM. The marginal $P(\text{DIS}{=}1) \approx 0.72$ for the parameters in Table 2.

Table 2: Conditional probability tables for the example. Parents are listed in the order they appear in the entity ordering; configurations are read in the same order (e.g. "10" under `[DIS_0026]` means `[RSK_0655]`=1, `[RSK_0115]`=0).

Variable	Parents	$P(\cdot{=}1 \mid \text{parents})$
`[RSK_0655]`	—	0.8169
`[RSK_0115]`	—	0.8013
`[DIS_0026]`	`[RSK_0655]`, `[RSK_0115]`	{00:0.138, 01:0.250, 10:0.641, 11:0.874}
`[SYM_0760]`	`[DIS_0026]`	{0:0.555, 1:0.103}
`[SYM_0282]`	`[DIS_0026]`	{0:0.846, 1:0.851}

We draw CPT entries and parentless priors independently per network from $\text{Uniform}[0, 1)$ and round them to four decimal places. This random parameterization creates heterogeneous ground-truth probabilities across networks, prompts, and target diseases.

B TEXT REALIZATION AND DOCUMENTATION POLICIES

B.1 SAMPLING FROM BAYESIAN NETWORKS

For each instantiated BN, we draw 30,000 joint binary configurations by ancestral sampling and verbalize every configuration as a clinical-note-style text. The fixed corpora contain 250 three-variable networks (7.5 million notes) and 400 five-variable networks (12 million notes). Additional epochs repeat these corpora. For example, 150k samples per network denotes five exposures to 30k distinct sampled notes. Complete-documentation training uses all six or 120 mention orders, respectively, and all four styles. Incomplete documentation training uses all orders in which the disease variables are the last to be mentioned.

B.2 ENTITY TOKENS

Each variable is mapped to a synthetic entity token, e.g., `[RSK_042]`, `[DIS_083]`, or `[SYM_017]`. Entity pools are disjoint across networks, preventing distributional information about a token from transferring across different Bayes Networks.

We add each entity token to the tokenizer as a single token before continual pretraining. For each role, we average the input embeddings of the base-tokenizer pieces for the corresponding word: `symptom`, `disease`, and `risk`. We initialize each entity's input-embedding and output-projection rows with the same reference vector plus Gaussian noise with standard deviation 0.01 per coordinate. Each entity receives its own noise draw to ensure a different randomly sampled starting point for each entity.

B.3 TEMPLATE FAMILIES

The four styles are `simple`, `structured`, `medium_narrative`, and `rich_narrative`. Each structure/order/style stratum supplies 50 training templates and 10 held-out templates generated with GPT-5-mini. For 3-var and 5-var networks, we jointly generate training and held-out templates and then divide them into training and held-out templates, with the 10 held-out templates further divided into a validation and a test split. For all the templates, we ensure that they are all distinct by de-duplicating repetitive templates every time new templates are generated. Therefore, the held-out template could serve as an in-distribution held-out template that generalizes to different wording while maintaining similar writing styles.

Validation and test wording partitions. Within each stratum, five full templates are assigned to the validation split and five to the test split. The validation split is used to assess whether performance has saturated over training. For both the 3-variable and 5-variable settings, we observe that validation performance has saturated over the final several checkpoints by epoch 10 for the 1B model and by epoch 5 for the 7B model, which are the training endpoints used in our reported runs for Section 4.1. All main recovery-table entries use held-out templates in the test split. We report training and validation controls separately. All splits probe BNs represented in training; "held-out" refers to wording, not unseen networks.

Examples of the four styles for one five-variable configuration follow.

> **simple.** History includes `[RSK_0]` and `[RSK_1]`, and the patient reports `[SYM_3]` and `[SYM_4]`; assessment shows `[DIS_2]` on current evaluation.
>
> **structured.** PMH: history includes `[RSK_0]` and `[RSK_1]`. HPI: patient reports `[SYM_3]` and `[SYM_4]`. Assessment: `[DIS_2]` noted during today's exam.
>
> **medium_narrative.** History includes `[RSK_0]` and `[RSK_1]` in the chart. The patient reports `[SYM_3]` and `[SYM_4]` over the past week. On exam, assessment shows `[DIS_2]`.
>
> **rich_narrative.** Visit for symptom review and medication check. History includes `[RSK_0]`. History includes `[RSK_1]` as well. Subjective: reports `[SYM_3]` and `[SYM_4]` that have progressed. Assessment shows `[DIS_2]`. Plan discussed and referrals offered.

The four styles share the same evidence content, entity assignments, and mention orders but differ in surrounding lexical context, sentence structure, and the presence of irrelevant clinical chrome. Each style is rendered with all valid mention orders so that any single entity appears in every position with the same expected count under all shuffled orders.

B.4 DOCUMENTATION POLICY FOR ABSENCE AND NON-MENTION

Disease variables are always explicitly realized during training. If $D_i = 1$, the note contains `[DIS_i]`; if $D_i = 0$, it contains `no [DIS_i]`. The negation fraction α applies only to absent evidence variables, i.e., risk factors and symptoms.

We define documentation in two stages. First, each evidence variable is observed with probability $1 - q$ and unobserved with probability q. If an evidence variable is unobserved, it is omitted from the note regardless of its true value. If it is observed, present evidence variables are always mentioned, while absent evidence variables follow the α-rule:

$$X = 0 \quad \mapsto \quad \begin{cases} \text{explicit negation} & \text{with probability } \alpha, \\ \text{omission} & \text{with probability } 1 - \alpha. \end{cases}$$

Thus, $q = 0$ gives the full observation with partial documentation regime: all evidence is observed, and non-mention can only arise because an absent variable was omitted rather than explicitly negated. When $q > 0$, non-mention can also arise because the variable was never observed, making omission ambiguous between absence and unknown evidence.

The q draw is taken independently per evidence variable per note. Conditional on the variable being observed, the α Bernoulli draw is taken independently per absent evidence variable, so a single note

may contain a mix of negated and omitted absent variables. Disease variables are always mentioned as present or explicitly negated: an absent disease variable is always realized as `no [DIS_i]` regardless of α or q, so the disease polarity contrast is preserved for p^{norm} in every training regime.

The full set of cases for an evidence variable Y under the joint (α, q) policy is summarized in Table 3.

Table 3: Documentation policy for an evidence variable Y under negation fraction α and unobserved-rate q. Disease variables are always mentioned as present or explicitly negated and never appear in the non-mention path.

Y value	Observed?	Documented?	Realization in note
$Y = 1$	yes (w.p. $1 - q$)	always	present mention of Y
$Y = 1$	no (w.p. q)	cannot	non-mention
$Y = 0$	yes (w.p. $1 - q$)	w.p. α	explicit negation of Y
$Y = 0$	yes (w.p. $1 - q$)	w.p. $1 - \alpha$	non-mention
$Y = 0$	no (w.p. q)	cannot	non-mention

C Training configuration

All experiments use continual pretraining from pretrained OLMo-2 checkpoints. We continue pretraining 1B and 7B models on the generated corpora and save checkpoints throughout training for acquisition-dynamics analyses.

Each complete-documentation corpus contains $30{,}000$ notes per network: 7.5 million notes for three-variable networks and 12 million for five-variable networks. One epoch presents each corpus once, so two, five, and ten epochs give 60k, 150k, and 300k note presentations per network. The omission grid uses two epochs for every policy.

Table 4: Training hyperparameters.

Hyperparameter	3-variable	5-variable
Base model	OLMo-2 1B / 7B	OLMo-2 1B / 7B
Notes per network per epoch	30K	30K
Networks	250	400
Effective batch	1B: 512; 7B: 1024	same
Sequence length	512	512
Optimizer	AdamW	AdamW
Learning rate	1B: 7.4487×10^{-5}; 7B: 6.1499×10^{-5}	same
Learning-rate schedule	Linear decay; no warmup	same
Weight decay	0.1	0.1
Adam (β_1, β_2)	$(0.9, 0.95)$	$(0.9, 0.95)$
Adam ϵ	10^{-8}	10^{-8}
Gradient clipping norm	1.0	1.0
Precision	bf16	bf16

We use `OLMo-2-0425-1B` and `OLMo-2-1124-7B` as the base checkpoints. For the main-table models, three-variable 1B uses four GPUs and three-variable 7B uses eight NVIDIA H200 GPUs. Five-variable 1B uses four NVIDIA A100-SXM4 GPUs (80 GB each) for ten epochs, with a recorded training-loop runtime of 48.6 hours. Its per-device batch size of 32 and four gradient-accumulation steps give an effective batch size of 512. Five-variable 7B uses eight NVIDIA H200 GPUs for five epochs.

D Probing and metric details

D.1 Held-out probe construction

For complete documentation, the evaluation panels contain ten BNs per topology: 50 three-variable BNs and 80 five-variable BNs, all drawn from the corresponding training corpora. For every disease target, we enumerate all subsets of non-target variables and all binary assignments to those subsets; another disease may be an observed variable. We render valid mention orders through each language

style, truncating before the target. The target posterior marginalizes the remaining unobserved variables. Predictions from the five full templates assigned to each held-out split are aggregated separately.

The all-disease panels contain 70 and 160 disease targets, giving $4 \times 70 \times 3^2 = 2,520$ and $4 \times 160 \times 3^4 = 51,840$ semantic/style analysis points per split. The single-disease subsets contain 30 3-var and 10 5-var BNs, while the multi-disease subsets contain 20 3-var and 70 5-var BNs. All probing outputs are probed and saved in BF16, and the inference precision is held fixed.

D.2 Phrase probabilities for negated continuations

We score each positive and negated continuation using its full sequence probability, including every token in a multi-token phrase. The normalization below compares these two sequence probabilities. It conditions on the two candidate continuations to calculate the normalized probabilities.

D.3 Per-target normalized probability estimate p^{norm}

For each disease target, the single-DIS probe contrasts the positive disease token with its explicit negation:

$$p^{\mathrm{norm}}(D_i \mid \mathcal{I}) = \frac{P_{\mathrm{LM}}(\texttt{[DIS_i]} \mid \mathcal{I})}{P_{\mathrm{LM}}(\texttt{[DIS_i]} \mid \mathcal{I}) + P_{\mathrm{LM}}(\texttt{no [DIS_i]} \mid \mathcal{I})}.$$

$P_{\mathrm{LM}}(\cdot \mid \mathcal{I})$ denotes sequence probability. Equivalently, if ℓ_+ and ℓ_- are the summed token log-probabilities of the two continuations, stable normalization computes $p^{\mathrm{norm}} = \exp(\ell_+ - \mathrm{logaddexp}(\ell_+, \ell_-))$. p^{norm} is compared to the ground-truth posterior $P^\star(D_i = 1 \mid \mathcal{I})$.

D.4 Template and order aggregation

An analysis point is indexed by BN, target disease, observed-variable set, its binary state assignment, and language style. We average normalized model probabilities over the selected templates and mention orders within this cell before computing errors. The four style means remain separate. This removes within-cell wording variation from the point estimate while retaining between-style differences. Equal weighting of cells also gives BNs with more disease targets more per-target observations.

D.5 Posterior recovery metrics

For cell i, let $p_i^\star$ be the stored posterior reference and $\hat{p}_i$ the template/order-averaged model estimate. The primary MAE is $N^{-1} \sum_i |\hat{p}_i - p_i^\star|$; Pearson correlation uses the same cells. Conditioning depth L_i is the number of visible non-target variables.

D.6 Multi-target ranking metrics

The pairwise analysis includes only networks with exactly two diseases, both unobserved. The targets are scored independently and matched by BN, visible non-disease evidence, and style after template/order averaging. The diseases can co-occur, and their two binary probabilities are not renormalized to sum to one. The panels contain 240 pairs from 20 three-variable BNs and 6,480 pairs from 60 five-variable BNs. Three-disease networks are excluded from these pairwise metrics, while retained in multi-disease per-target summaries. Reference ties are excluded; model ties count as incorrect rankings.

Top-1 agreement. Top-1 measures whether the model assigns the largest score to the disease with the largest ground-truth marginal posterior:

$$\mathrm{Top1}^{\mathrm{norm}} = \mathbb{1}\left[\arg\max_{D_i \in \mathcal{D}} p^{\mathrm{norm}}(D_i \mid \mathcal{I}) = \arg\max_{D_i \in \mathcal{D}} P^\star(D_i = 1 \mid \mathcal{I})\right].$$

Two-disease probability gaps.

The supplementary probability-gap analyses below use the raw-probability slope β_p; the main text uses the log-odds gap slope β.

For two-target analyses, we order each pair so that D_{high} has the larger ground-truth posterior and define

$$\Delta^\star = P^\star(D_{\text{high}} = 1 \mid \mathcal{I}) - P^\star(D_{\text{low}} = 1 \mid \mathcal{I}),$$
$$\Delta^{\text{norm}} = p^{\text{norm}}(D_{\text{high}} \mid \mathcal{I}) - p^{\text{norm}}(D_{\text{low}} \mid \mathcal{I}).$$

Top-1 measures agreement between the signs of the model and reference contrasts; the through-origin OLS slope β_p of Δ^{norm} on $\Delta^\star$ summarizes their relative scale. A slope $0 < \beta_p < 1$ indicates aggregate gap compression, but does not imply correct ranking on every pair or compression conditional on a correct ranking. We report Top-1 agreement separately.

E Omission references under incomplete documentation

The reported omission analyses in RQ4–5 evaluate one omitted non-disease variable per query. We therefore derive the reference for a single omitted variable Y.

The non-mention coefficient $\widehat{\lambda}$ in §E.3 measures whether the model's posterior under non-mention sits closer to a marginalization reading (λ=0, g_{marg}) or an absence-imputation reading (λ=1, g_{abs}). The diagnostic target is the corpus-implied posterior g_{NM}, the prediction an optimal frequency-matching learner of the training distribution should recover under non-mention, and its corresponding projection $\widehat{\lambda}^\star$. The behaviour of $\widehat{\lambda}^\star$ depends on which documentation regime the training corpus uses:

- *Full observation with partial documentation* ($q = 0$): for $\alpha < 1$, non-mention implies absence and $\widehat{\lambda}^\star = 1$. At $\alpha = 1$, omission has zero probability, so g_{NM} and $\widehat{\lambda}^\star$ are undefined.
- *Partial observation with partial documentation* (q>0): $\widehat{\lambda}^\star$ is *graded*, taking values in $[0, 1]$ that depend on both α and q. As q grows, more non-mention events come from unobserved evidence rather than documented absence, and $\widehat{\lambda}^\star$ is pulled back toward marginalization.

The model's distance from $\widehat{\lambda}^\star$ indicates the diagnostic quantity.

E.1 Documentation policy likelihoods

Let $Y \in \{0, 1\}$ be a non-disease evidence variable. The training pipeline treats Y in two stages:

- With probability q, Y is *unobserved* (drawn from the BN but not eligible to appear in the note regardless of value).
- Otherwise Y is observed; if Y=1 it is mentioned with the present template; if Y=0 it is mentioned with explicit negation with probability α and omitted with probability $1 - \alpha$.

The corpus likelihoods of the omission event therefore satisfy

$$\ell_1 = P(\text{omit} \mid Y{=}1) = q, \qquad \ell_0 = P(\text{omit} \mid Y{=}0) = q + (1 - q)(1 - \alpha).$$

The two regimes correspond to qualitatively different (ℓ_0, ℓ_1) patterns:

q=0, α=1 : $\ell_0 = \ell_1 = 0$ (no non-mention in the training distribution).
q=0, α<1 : $\ell_0 = 1 - \alpha > 0,\ \ell_1 = 0$ (non-mention deterministically implies absence).
q>0, $\alpha \in [0, 1]$: $\ell_0, \ell_1 > 0$ (non-mention is a graded mixture of absence and unknown).

E.2 Corpus-implied posterior under non-mention

For a probe prefix e, the ideal frequency-matching learner of this corpus predicts

$$g_{\text{NM}} = P^\star(D = 1 \mid e,\ Y \text{ omitted}) = \frac{\ell_0\, P^\star(D = 1, e, Y = 0) + \ell_1\, P^\star(D = 1, e, Y = 1)}{\ell_0\, P^\star(e, Y = 0) + \ell_1\, P^\star(e, Y = 1)},$$

which is the ground-truth update of D given the omission event under the training-time documentation policy. The two endpoints used in the $\widehat{\lambda}$ projection are special cases of g_{NM} at the extremes of the documentation policy: $g_{\text{marg}} = g_{\text{NM}}\big|_{\ell_0 = \ell_1 > 0}$ (omission carries no information about Y) and $g_{\text{abs}} = g_{\text{NM}}\big|_{\ell_1 = 0, \ell_0 > 0}$ (omission implies absence).

E.3 Projection onto the marginalization-to-absence axis

The main paper reports $\widehat{\lambda}_{\text{model}}$ and the policy-specific reference $\widehat{\lambda}^{\star}$ on a shared axis so that the two are directly comparable. For each analysis cell c containing probes $i \in c$, define

$$x_i = g_{\text{abs},i} - g_{\text{marg},i}, \qquad y_i^{\text{model}} = p_i^{\text{norm}} - g_{\text{marg},i}, \qquad y_i^{\star} = g_{\text{NM},i} - g_{\text{marg},i}.$$

The model coefficient is the through-origin OLS slope of y^{model} on x,

$$\widehat{\lambda}_{\text{model},c} = \frac{\sum_{i \in c} x_i \, y_i^{\text{model}}}{\sum_{i \in c} x_i^2},$$

i.e. the projection of the model's posterior residuals onto the $(g_{\text{abs}} - g_{\text{marg}})$ axis. The reference coefficient is the analogous projection of the optimal-learner target onto the same axis,

$$\widehat{\lambda}_c^{\star} = \frac{\sum_{i \in c} x_i \, y_i^{\star}}{\sum_{i \in c} x_i^2}.$$

The two are apples-to-apples summaries on the same axis; their difference

$$\Delta\lambda_c = \widehat{\lambda}_{\text{model},c} - \widehat{\lambda}_c^{\star}$$

captures whether the model over-imputes absence ($\Delta\lambda_c > 0$) or underuses the absence signal ($\Delta\lambda_c < 0$).

E.4 Fit quality and filtering

For the model fit we additionally report the through-origin coefficient of determination

$$R_c^2 = 1 - \frac{\sum_{i \in c} \left(y_i^{\text{model}} - \widehat{\lambda}_{\text{model},c} x_i\right)^2}{\sum_{i \in c} (y_i^{\text{model}})^2}.$$

A low R_c^2 indicates that a single projection coefficient is a poor summary of the cell's per-probe behavior even when the average $\widehat{\lambda}_{\text{model},c}$ matches $\widehat{\lambda}_c^{\star}$. We therefore read $\widehat{\lambda}_{\text{model}}$ alongside R_c^2, and prefer direct MAE against g_{NM} when R_c^2 is low.

Denominator filter. We retain probes with $|x_i| = |g_{\text{abs},i} - g_{\text{marg},i}| \geq 0.05$ to focus on queries where the two reference interpretations differ. Both projections and the direct MAE use this same filter. Appendix G reports sensitivity to removing the filter.

F Extended complete-documentation results

F.1 Recovery across training budgets

For each complete-documentation corpus, we train separate 1B models for one, two, five, and ten epochs, and separate 7B models for one and five epochs. Each run starts from pretrained OLMo-2 weights and uses a linear learning-rate schedule over its specified budget. Each epoch presents the same 30,000 notes per BN. Table 5 and Figure 8 report the twelve final models. The main table uses ten epochs for 1B and five epochs for 7B.

Three-variable recovery shows limited gains beyond the first few epochs. For 1B, the two- and ten-epoch runs have similar overall MAE (0.0495 and 0.0483), although the five-epoch run achieves the lowest error (0.0429). The 7B runs have similar recovery after one and five epochs. These comparisons support a plateau in aggregate recovery, with variation across budgets. For five-variable models, within-run probing on held-out validation prompts shows diminishing gains in recovery near the end of the ten-epoch 1B run. The five-epoch 7B run shows no further improvement in its final epoch.

The five-variable gains concentrate in conditional queries. For 1B, increasing the budget from one to ten epochs reduces MAE for queries conditioning on all four other variables from 0.2197 to 0.0649, while marginal MAE increases from 0.0215 to 0.0358. The 7B results show the same pattern: conditional estimates improve with longer training while marginal error remains low. These comparisons link the aggregate gains to improved conditional recovery.

Table 5: **Probability recovery across all evaluated budgets.** Final models from independent runs on held-out test templates, including all disease targets. MAE and Pearson r weight template/order-averaged semantic/style cases equally. Each epoch repeats a fixed corpus of 30,000 notes per BN. L_k denotes k stated non-target variables; dashes mark inapplicable depths.

			MAE ↓						Pearson r ↑					
Variables	Model	epochs	All	L_0	L_1	L_2	L_3	L_4	All	L_0	L_1	L_2	L_3	L_4
3-var	1B	1	0.0675	0.0397	0.0743	0.0676	–	–	0.9411	0.9804	0.9266	0.9470	–	–
3-var	1B	2	0.0495	0.0381	0.0639	0.0380	–	–	0.9555	0.9826	0.9263	0.9765	–	–
3-var	1B	5	0.0429	0.0512	0.0513	0.0325	–	–	0.9645	0.9665	0.9435	0.9838	–	–
3-var	1B	10	0.0483	0.0688	0.0566	0.0349	–	–	0.9536	0.9270	0.9251	0.9846	–	–
3-var	7B	1	0.0410	0.0309	0.0466	0.0379	–	–	0.9757	0.9892	0.9671	0.9811	–	–
3-var	7B	5	0.0402	0.0357	0.0452	0.0362	–	–	0.9714	0.9853	0.9592	0.9804	–	–
5-var	1B	1	0.1852	0.0215	0.1411	0.1694	0.1958	0.2197	0.6357	0.9953	0.7694	0.6836	0.6138	0.5557
5-var	1B	2	0.1682	0.0198	0.1235	0.1530	0.1783	0.2025	0.6940	0.9964	0.8118	0.7347	0.6760	0.6252
5-var	1B	5	0.0965	0.0292	0.0653	0.0895	0.1025	0.1147	0.8903	0.9921	0.9459	0.9048	0.8834	0.8605
5-var	1B	10	0.0621	0.0358	0.0486	0.0622	0.0647	0.0649	0.9495	0.9876	0.9712	0.9524	0.9465	0.9425
5-var	7B	1	0.1368	0.0157	0.1243	0.1301	0.1419	0.1507	0.7941	0.9973	0.8208	0.8082	0.7881	0.7769
5-var	7B	5	0.0517	0.0176	0.0376	0.0543	0.0546	0.0515	0.9642	0.9967	0.9821	0.9642	0.9618	0.9611

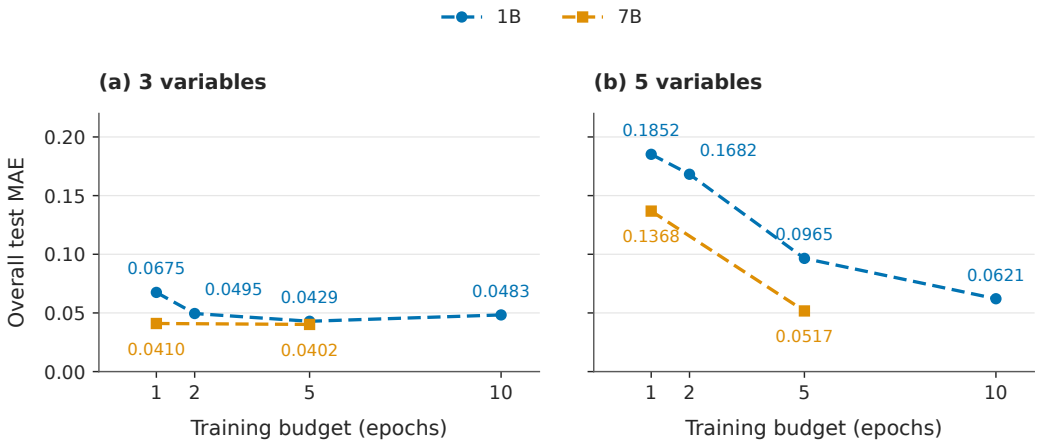


Figure 8: **Three-variable gains diminish; five-variable recovery improves with longer training.** Overall test MAE for the twelve final models in Table 5; lower values indicate better recovery. Each point represents an independent run, and dashed lines connect training budgets for comparison. Labels give MAE.

F.2 Conditioning level and evidence probability

Figure 3 (a) follows the five-variable 7B model across checkpoints of its five-epoch run on held-out test templates. Figure 3 (b) uses the final five-variable models at ten epochs for 1B and five epochs for 7B, also on test templates. Figure 3 averages absolute errors within each conditioning level after template/order averaging, retaining the four language styles as separate cases.

For Figure 3, we compute $P^\star(E)$ by summing the generator's joint distribution over variables outside the stated evidence. We exclude L_0, whose empty evidence has probability one, leaving 51,200 conditional semantic/style cases per model. The bins $[0.1, 1]$, $[0.01, 0.1)$, $[0.001, 0.01)$, and $[0, 0.001)$ contain 27,268, 18,248, 4,340, and 1,344 cases, covering 80, 80, 78, and 40 BNs, respectively. We compute equal-weight MAE within each bin against the stored posterior references used throughout the recovery analysis. For five-variable recovery, these saved references come from the cached, rounded joint distribution; evidence probabilities and RQ3 reference updates use the generator CPTs. Bin composition differs in conditioning level, networks, and target probabilities; the comparison describes an association with evidence probability.

F.3 Probability magnitudes and ranking

For per-target fits, we clip reference and model probabilities to $[0.005, 0.995]$ and fit $\mathrm{logit}(\hat{p}) = a + b\,\mathrm{logit}(p^\star) + \epsilon$. The intercept allows an overall shift. A slope below one describes compression

Table 6: **Three-variable recovery and compression across wording splits.** Both panels use the same final models: ten epochs for 1B and five for 7B. Panel (a) reports recovery over all disease targets, with equal weight per semantic/style case. Panel (b) gives training/test slopes. Training refers to corpus-generator wording; Appendix B defines validation and test partitions. Per-target log-odds slopes b_{all}, b_{single}, and b_{multi} cover all, single-disease, and multi-disease cases. Two-disease pairs supply b_{pair}, log-odds gap slopes β and β_{correct}, and raw-probability gap slope β_p; β_{correct} uses correctly ranked pairs.

(a) Probability recovery

		Training templates		Validation		Test	
Model	Epochs	r	MAE	r	MAE	r	MAE
1B	10	0.9985	0.0091	0.9503	0.0508	0.9536	0.0483
7B	5	0.9985	0.0084	0.9708	0.0408	0.9714	0.0402

(b) Compression: training/test slopes

Model	Epochs	b_{all}	b_{single}	b_{multi}	b_{pair}	β	β_{correct}	β_p
1B	10	1.003/0.830	1.005/0.923	1.002/0.731	1.000/0.650	0.997/0.338	0.999/0.392	0.997/0.391
7B	5	1.000/0.836	1.004/0.957	0.997/0.708	1.000/0.707	0.996/0.503	0.997/0.514	0.998/0.576

Table 7: **Probability magnitudes and ranking agreement.** Test-only final checkpoints. Per-target log-odds slopes b include an intercept; gap slopes β_p fit through the origin. Multi-disease per-target metrics include three-disease networks, whereas gap and Top-1 metrics require exactly two diseases, both unobserved. All fits are unweighted. Top-1 reports ranking agreement as a percentage.

			Per-target b			Two diseases	
Variables	Model	Samples/net.	All	Single	Multi	β_p	Top-1 (%)
3-var	1B	30k	0.7308	0.7973	0.6618	0.6000	80.42
3-var	1B	60k	0.8302	0.8946	0.7629	0.4845	78.75
3-var	1B	150k	0.8661	0.9459	0.7829	0.4641	80.42
3-var	1B	300k	0.8298	0.9227	0.7314	0.3912	78.33
3-var	7B	30k	0.8425	0.9338	0.7468	0.6611	82.92
3-var	7B	150k	0.8355	0.9574	0.7077	0.5762	87.50
5-var	1B	30k	0.3332	-0.0085	0.3523	0.2012	73.95
5-var	1B	60k	0.4025	0.1635	0.4160	0.1915	72.75
5-var	1B	150k	0.6446	0.6378	0.6449	0.6426	84.31
5-var	1B	300k	0.7760	0.8416	0.7723	0.7299	90.57
5-var	7B	30k	0.4965	0.4231	0.5007	0.3677	78.80
5-var	7B	150k	0.9187	1.0401	0.9119	0.8527	92.79

on the log-odds scale; a unit slope can coexist with offset and residual error. Single- and multi-disease groups contain networks with one or multiple disease variables, respectively.

The supplementary raw-gap fit (β_p) uses unclipped probability differences and no intercept. It includes only networks with two diseases, both unobserved, and aligns their predictions under the same non-disease evidence and language style. We exclude reference ties and count model ties as incorrect. Appendix D.6 gives the gap orientation and query counts. Every compression fit gives equal weight to its analysis cases.

For the main-text log-odds gap fit, we first average probabilities over templates and orders, then clip them to $[0.005, 0.995]$, as for b. Let $\ell(p)$ denote the resulting logit. For each reference-ordered pair, we compute $\Delta^\star_\ell = \ell(p^\star_{\text{high}}) - \ell(p^\star_{\text{low}})$ and $\hat{\Delta}_\ell = \ell(\hat{p}_{\text{high}}) - \ell(\hat{p}_{\text{low}})$, and fit $\beta = \sum_j \Delta^\star_{\ell,j}\hat{\Delta}_{\ell,j} / \sum_j (\Delta^\star_{\ell,j})^2$. The matched per-target slope b_{pair} includes an intercept and uses the two target probabilities from each of these same pairs. Ranking and the correctly ranked subset use unclipped probabilities. Clipping gives eight five-variable pairs zero reference log-odds contrast; these pairs contribute zero to both sums defining β.

Whole-BN paired bootstrap intervals for $b_{\text{pair}} - \beta$ are $[0.215, 0.391]$, $[0.133, 0.271]$, $[0.098, 0.168]$, and $[0.084, 0.139]$, in the configuration order of Table 1. All four intervals lie above zero. Excluding pairs with any endpoint outside $(0.005, 0.995)$ gives gap slopes of 0.34, 0.51, 0.61, and 0.79.

Whole-BN bootstrap intervals use 1,000 resamples with seed 42. For the 7B five-epoch models, the single-disease slope intervals are $[0.9368, 0.9847]$ and $[0.9200, 1.1206]$ for three and five variables. Multi-disease intervals are $[0.6873, 0.7277]$ and $[0.8855, 0.9362]$; two-disease raw-probability gap intervals are $[0.5443, 0.6059]$ and $[0.8283, 0.8730]$. These intervals describe network variation within a checkpoint, excluding training-seed uncertainty.

Both three-variable models have near-unit per-target and gap slopes on training wording (Table 6). On test wording, per-target slopes decrease, and log-odds gap slopes fall to 0.338 for 1B and 0.503 for 7B. Restricting the gap fit to correctly ranked pairs retains compression, with slopes of 0.392 and 0.514.

F.4 Evidence updates and Null selection

Each evidence comparison fixes the BN, disease target, retained evidence assignment, and language style, then adds one previously unstated variable. We average model predictions over templates and orders at each endpoint before taking their difference. The five-variable analysis includes both values of every added variable and all four styles, yielding $138{,}240$ comparisons per model over 80 BNs. Counts describe coverage; shared endpoints and networks make comparisons statistically dependent.

We select Null contexts by graph separation: the target and added variable are conditionally independent given the retained evidence variables. This selects $33{,}200$ comparisons across 70 BNs per model; direct calculation from the generator CPTs confirms zero reference changes up to floating-point precision. Figure 4 shows all these comparisons for 7B. Its diagonal represents unchanged model probabilities, and the shaded band marks deviations within five percentage points as a visual guide.

Figure 4 groups the remaining comparisons by absolute reference updates in the intervals $(0, 5)$, $[5, 10)$, $[10, 15)$, $[15, 25)$, and $[25, 100]$, in percentage points. The five bins contain $48{,}076$, $17{,}396$, $11{,}640$, $12{,}452$, and $15{,}476$ comparisons. The first bin includes 128 nonstructural cases with $0 < |d^\star| < 10^{-5}$ in probability units; these remain distinct from structural Null contexts. Each bar averages $|\hat{d}|$ and each diamond averages $|d^\star|$ over the same comparisons, with equal weight per comparison. These absolute magnitudes discard update direction, and bin composition differs across networks, conditioning levels, and evidence probabilities.

F.5 Ranking and magnitude can diverge during one run

The three-variable 7B five-epoch run provides a within-run comparison on a fixed test bank. Top-1 remains at 87.5% from epochs three through five while the raw-probability gap slope falls. The posterior, ranking, and update metrics thus need not improve together. This example describes one run and does not establish a general learning order.

The update slope γ regresses model log-odds changes on reference changes through the origin over informative additions. We clip endpoint probabilities to $[0.005, 0.995]$ before taking log-odds.

Table 8: **Different aspects of recovery can diverge within one run.** Three-variable 7B, five-epoch run, held-out test wording. Posterior MAE uses single- or multi-disease cells; γ uses multi-disease updates. Top-1 and β_p require exactly two unobserved diseases.

Epoch	Single MAE	Multi MAE	Multi γ	Top-1 (%)	Gap β_p
1	0.0391	0.0701	0.6409	85.00	0.6920
2	0.0265	0.0501	0.7285	86.25	0.6604
3	0.0192	0.0489	0.7647	87.50	0.6278
4	0.0177	0.0560	0.7057	87.50	0.5907
5	0.0150	0.0591	0.6833	87.50	0.5762

G EXTENDED INCOMPLETE-DOCUMENTATION RESULTS

Table 9 reports the test results behind Figure 5, using only three-variable OLMo-2 1B models trained for two epochs. We average test templates within each BN, target, evidence condition, and style. The pool contains 400 points across 50 BNs. Each point has one positive stated variable and one omitted non-disease variable. The common filter $|g_{\text{abs}} - g_{\text{marg}}| \geq .05$ retains 248 points across 42 BNs for every policy and metric.

Appendix E defines the policy posterior and the unconstrained projection. At $\alpha = 1, q = 0$, omission has zero probability, so we exclude this cell from the table and assign it no policy reference. Figure 5 uses 5,000 bootstrap resamples of whole retained BNs with seed 42. Each resample retains all points from each sampled BN. The intervals measure variation across BNs within the available runs and exclude training-seed uncertainty.

Removing the endpoint filter retains all 400 points across 50 BNs and changes the model coefficient by at most 0.007 across the 24 supported policies. At $q = 0$, coefficients range from 0.699 to 0.826 for single-disease BNs and from 0.753 to 0.874 for multi-disease BNs under the main filter. Both groups retain a shortfall from the absence reference of one.

Table 9: **Incomplete-documentation results.** All rows use the same 248 test points. MAE compares omission predictions with g_{NM} on the probability scale; R^2 measures the uncentered projection fit.

q	α	$\widehat{\lambda}_{\text{model}}$	$\widehat{\lambda}^{\star}$	MAE	R^2
0	0	0.810	1.000	0.086	0.610
0	0.25	0.831	1.000	0.088	0.598
0	0.5	0.755	1.000	0.099	0.517
0	0.75	0.725	1.000	0.107	0.492
0.1	0	0.582	0.679	0.072	0.481
0.1	0.25	0.477	0.618	0.096	0.345
0.1	0.5	0.426	0.526	0.085	0.297
0.1	0.75	0.272	0.368	0.090	0.129
0.1	1	-0.019	0.000	0.097	0.001
0.25	0	0.355	0.432	0.077	0.245
0.25	0.25	0.319	0.368	0.077	0.213
0.25	0.5	0.217	0.285	0.088	0.101
0.25	0.75	0.145	0.171	0.097	0.042
0.25	1	-0.016	0.000	0.088	0.001
0.5	0	0.171	0.213	0.080	0.068
0.5	0.25	0.143	0.171	0.086	0.044
0.5	0.5	0.094	0.122	0.091	0.019
0.5	0.75	0.045	0.066	0.079	0.005
0.5	1	-0.001	0.000	0.087	0.000
0.75	0	0.057	0.086	0.082	0.008
0.75	0.25	0.061	0.066	0.086	0.010
0.75	0.5	0.041	0.045	0.090	0.004
0.75	0.75	0.022	0.023	0.089	0.001
0.75	1	0.015	0.000	0.092	0.000

The model coefficient follows the policy-dependent absence cue, with attenuated shifts when the cue is strongest. At $q = 0$, model coefficients range from 0.725 to 0.831 against the reference of one. For $\alpha < 1$, the coefficient gap narrows as q increases. MAE and R^2 describe query-level accuracy and projection fit, respectively.

Fraction of the reference shift. The coefficients in Table 9 share a common scale, on which 0 corresponds to ignoring the unmentioned finding and 1 to treating it as absent. Because $\widehat{\lambda}^{\star}$ itself decreases with q, the absolute difference $\widehat{\lambda}^{\star} - \widehat{\lambda}_{\text{model}}$ shrinks as non-mention becomes less informative, even if the model's relative shortfall is unchanged. To compare the shortfall across policies, Table 10 reports the ratio $\widehat{\lambda}_{\text{model}}/\widehat{\lambda}^{\star}$, the fraction of the reference shift that the model applies. For the 15 policies with $\widehat{\lambda}^{\star} \geq 0.1$, the ratio lies between 0.73 and 0.87, with no clear trend in q or α. When $\widehat{\lambda}^{\star} < 0.1$, small absolute differences produce large changes in the ratio: for example, two

policies with the same $\widehat{\lambda}^\star = 0.066$ yield ratios of 0.68 and 0.92. We therefore exclude these policies from the range reported in the main text.

Several policies share the same reference coefficient: $\widehat{\lambda}^\star = 0.368$ for $(q, \alpha) = (0.1, 0.75)$ and $(0.25, 0.25)$, 0.171 for $(0.25, 0.75)$ and $(0.5, 0.25)$, and 0.066 for $(0.5, 0.75)$ and $(0.75, 0.25)$. This follows from the policy posterior g_{NM}, which depends on the documentation policy only through the likelihood ratio $\ell_0/\ell_1 = [q + (1-q)(1-\alpha)]/q$ (Appendices E.1–E.2); the three pairs have equal ratios of 3.25, 1.75, and 1.25, respectively. Their model coefficients differ because each policy corresponds to a separately trained model.

Table 10: **Fraction of the reference shift applied by the model.** Each ratio divides the model coefficient by the reference coefficient from Table 9, using the three-decimal values reported there. Ratios in parentheses have $\widehat{\lambda}^\star < 0.1$, where the reference shift is small and the ratio is unstable; they are excluded from the 73–87% range reported in the main text. Policies with $\alpha = 1$ are omitted because $\widehat{\lambda}^\star = 0$.

q	α	$\widehat{\lambda}_{\mathrm{model}}/\widehat{\lambda}^\star$
0	0	$0.810/1.000 = 0.81$
0	0.25	$0.831/1.000 = 0.83$
0	0.5	$0.755/1.000 = 0.76$
0	0.75	$0.725/1.000 = 0.73$
0.1	0	$0.582/0.679 = 0.86$
0.1	0.25	$0.477/0.618 = 0.77$
0.1	0.5	$0.426/0.526 = 0.81$
0.1	0.75	$0.272/0.368 = 0.74$
0.25	0	$0.355/0.432 = 0.82$
0.25	0.25	$0.319/0.368 = 0.87$
0.25	0.5	$0.217/0.285 = 0.76$
0.25	0.75	$0.145/0.171 = 0.85$
0.5	0	$0.171/0.213 = 0.80$
0.5	0.25	$0.143/0.171 = 0.84$
0.5	0.5	$0.094/0.122 = 0.77$
0.5	0.75	$(0.045/0.066 = 0.68)$
0.75	0	$(0.057/0.086 = 0.66)$
0.75	0.25	$(0.061/0.066 = 0.92)$
0.75	0.5	$(0.041/0.045 = 0.91)$
0.75	0.75	$(0.022/0.023 = 0.96)$