

Patient-level Information Extraction by Consistent Integration of Textual and Tabular Evidence with Bayesian Networks

Paloma Rabaey^{1,2*}†, Adrick Tench^{1,2*}†, Stefan Heytens³ and Thomas Demeester^{1,2}

¹Faculty of Engineering and Architecture, Ghent University,
Ghent, Belgium.

²imec, Leuven, Belgium.

³Ghent University Hospital, Ghent, Belgium.

*Corresponding author(s). E-mail(s): paloma.rabaey@ugent.be;
adrick.tench@ugent.be;

†These authors contributed equally to this work.

Abstract

Electronic health records (EHRs) form an invaluable resource for training clinical decision support systems. To leverage the potential of such systems in high-risk applications, we need large, structured tabular datasets on which we can build transparent feature-based models. While part of the EHR already contains structured information (e.g. diagnosis codes, medications, and lab results), much of the information is contained within unstructured text (e.g. discharge summaries and nursing notes). In this work, we propose a method for multi-modal patient-level information extraction that leverages both the tabular features available in the patient’s EHR (using an expert-informed Bayesian network) as well as clinical notes describing the patient’s symptoms (using neural text classifiers). We propose the use of *virtual evidence* augmented with a *consistency node* to provide an interpretable, probabilistic fusion of the models’ predictions. The consistency node improves the calibration of the final predictions compared to virtual evidence alone, allowing the Bayesian network to better adjust the neural classifier’s output to handle missing information and resolve contradictions between the tabular and text data. We show the potential of our method on the SimSUM dataset, a simulated benchmark linking tabular EHRs with clinical notes through expert knowledge.

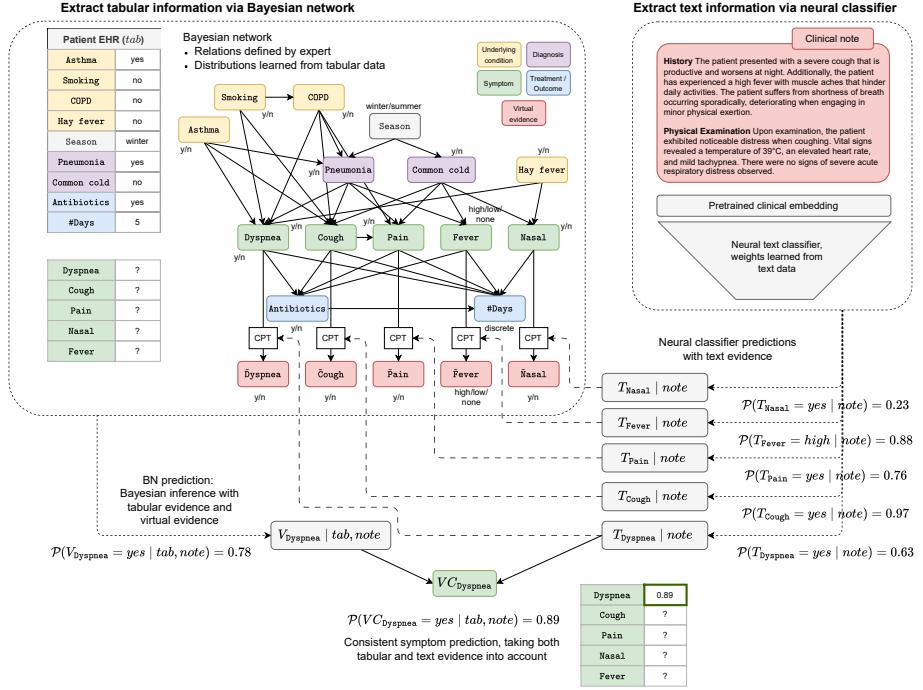


Fig. 1: Overview of our patient-level information extraction method which integrates both tabular and text evidence. As an example, we show how to extract the probability that a patient suffers from **Dyspnea**, given tabular evidence that is already encoded in the EHR, a clinical note describing the patient’s symptoms, and an expert-defined **Bayesian network (BN)** structure. On the right, the **neural classifiers** infer probabilities that the **text** mentions each symptom, with a 63% confidence for **Dyspnea** (in this case, “dyspnea” is not mentioned verbatim). The classifiers’ probabilities for each symptom are provided as **virtual evidence** to the **BN**, via the red “virtual” nodes in the network. Given all **tabular** and **virtual evidence**, the **BN** infers that the patient has a 78% chance of **Dyspnea** – since the patient has both **Pneumonia** and **Asthma**, this probability is high. The consistency node VC_{Dyspnea} combines these probabilities, arriving at an 89% chance that this patient has **Dyspnea**. Part of this figure is adapted from Rabaey et al. [1].

Keywords: Bayesian networks, Virtual evidence, Clinical Decision Support, Information Extraction, Multi-modal integration, Electronic Health Records

1 Introduction

In healthcare, storing patient information in a structured format is essential for ensuring continuity of care. This information, such as diagnosis codes, medication, and lab results, is usually stored in Electronic Health Records (EHRs). Often, these also contain a large quantity of unstructured free text, such as discharge summaries, nursing notes, procedural descriptions, and more [2]. All this information forms an invaluable resource for training clinical decision support systems, which have the potential to partially automate care processes such as diagnosis and treatment planning [3–5]. While the recent rise in large language models has shown great potential for processing clinical notes [6–8], these black-box systems lack interpretability [9–11]. In high-risk clinical applications, one should prefer robust and transparent systems built on simpler, feature-based models, like regression models, decision trees or Bayesian networks [12–14]. However, these models require large structured tabular datasets as a training resource to ensure their generalization to broader patient populations. As such, developing methods to extend the tabular portion of the EHR with as much information as possible could benefit a wide range of downstream applications.

In this work, we propose a method for patient-level information extraction that leverages both the structured tabular features available in a patient’s EHR and the unstructured clinical notes documenting physician encounters. We model the tabular portion of the EHR using a Bayesian network (BN) with an expert-defined structure and learned probabilities, enabling interpretable integration of background information. This network is then connected to the predictions of a text classifier that interprets the clinical notes. By linking these two modalities, the BN can adjust the text classifier’s predictions when they conflict with evidence from the tabular data, and infer missing information when the text is incomplete, all in a manner that remains transparent to the end user.

As a proof-of-concept for our method, we focus on a specific use-case built on the SimSUM dataset [1]. SimSUM is a simulated benchmark of 10,000 artificial patient records, linking tabular variables (like symptoms, diagnoses and underlying conditions) to associated clinical notes describing the patient encounter in the domain of respiratory diseases. To our knowledge, SimSUM is the only available clinical dataset that provides both tabular and textual data connected through a shared and fully known data-generating process. This inherent alignment between modalities, together with the known BN structure underlying the data, enables fundamental research on integrating tabular and text-based models.

Figure 1 shows the overview of our setup and proposed method. On one side, we have a tabular EHR containing encoded background, diagnoses, and treatment. In addition, we have a clinical text describing the symptoms experienced by this patient. Since such symptom information can be valuable for downstream applications (e.g., automated diagnostic systems), our goal is to incorporate it into the EHR in a structured format. In a standard information extraction pipeline, a neural text classifier – however advanced – is typically used to predict whether a symptom is mentioned in the text. Our approach additionally exploits the tabular information already present in the EHR by connecting it to the target concepts for extraction (i.e., symptoms) through a BN. The relations in the BN (i.e. which underlying conditions may give

rise to which symptoms) are provided by an expert, while the exact probabilities are learned from the tabular portion of the EHR.

Our main contribution lies in **enabling patient-level information extraction**, by **combining the predictions made by a Bayesian network and a text classifier in a probabilistic manner**, augmenting the established approach of virtual evidence with an additional *consistency node*. This combination enables our method to correct for inconsistencies and missing information in the text by leveraging tabular data and background knowledge through the BN. The consistency node improves the calibration of the final predictions compared to using virtual evidence alone, especially in abnormal cases where information is missing from the text, while retaining high predictive power in common cases. Moreover, by obtaining a probabilistic rather than deterministic encoding of the symptoms within the tabular record, our approach yields extracted information that is more robust for downstream use. Consequently, users of the final tabular dataset can either apply a threshold to obtain hard labels for these concepts or directly utilize the probabilistic outputs, depending on their specific application and system requirements.

The remainder of our work is structured as follows. In Section 2 we discuss related work, including an in-depth comparison of our work with previous approaches for integrating BNs with neural text classifiers. Then, Section 3 explains the building blocks of our multimodal approach and how these are connected together using the consistency node. In Section 4, we report the performance of our method on the SimSUM dataset for various sample sizes, and take a closer look at how our multimodal approach improves over uni-modal baselines. Finally, Sections 5 and 6 summarize the potential of our method, while also addressing its limitations. The code for this project can be found at <https://github.com/AdrickTench/patient-level-IE>.

2 Related work

In Section 2.1, we first introduce related work on representation learning for electronic health records (EHRs), in particular focusing on multimodal approaches that integrate tabular data and text, as well as those that incorporate background knowledge. Section 2.2 then zooms in on two closely related methods that integrate text in BNs, contrasting them with our approach. Finally, Section 2.3 explains the concept of virtual evidence.

2.1 Multimodal EHR Representation Learning

Modeling tabular data and text

Many studies focus on combining two modalities commonly found in EHRs — structured tabular data (e.g. disease codes, patient demographics, treatment codes, lab results) and unstructured text (e.g. discharge summaries or nursing notes) — for representation learning. Most methods use state-of-the-art encoders to learn a representation for each modality, combining both through simple concatenation [15], ensemble methods [16] or an attention mechanism [17–19]. While our work also focuses on combining these two modalities, our goal is not to learn a black-box patient-level representation for downstream prediction tasks. Instead, we aim to enrich the

structured portion of the EHR with information extracted from the text, thereby facilitating its use in interpretable downstream prediction models such as regression models, decision trees, or BNs [12]. Furthermore, by providing a probabilistic view on the extracted structured variables, our approach enables intermediate manual inspection of the dataset before it is used in downstream models, which is not possible with black-box patient representations.

Integrating background knowledge

Particularly interesting to our work are representation learning methods that include some form of background knowledge to improve the learned multimodal representations. Nguyen et al. [18] enhance their multimodal encoding for the structured and text portions of the EHR with a clinical reasoning embedding, which is obtained by retrieving relevant clinical documents and asking an LLM to reason over them. Xu et al. [20] use a similar information retrieval module to enhance disease code embeddings.

Instead of incorporating relevant information from clinical corpora, we focus on incorporating graph-based background knowledge. In this line of work, both Choi et al. [21] and Park et al. [22] model the structured portion of the EHR with a graph attention network, where each admission is represented as a hierarchical graph containing procedure, diagnoses and prescription nodes. As an extension, Choi et al. [23] learn to automatically construct the causal structure of a patient admission (in particular, modeling which diagnoses lead to the prescription of which treatment), by training a graph convolutional transformer jointly with supervised medical prediction tasks. This approach is conceptually similar to the BN component of our method. However, in our case, we (i) learn a BN rather than a graph convolutional transformer, and (ii) rely on domain experts to specify the medical relationships symbolically, learning only the underlying probability distributions from the data rather than the relations among variables.

Other works incorporate graph-based background knowledge into EHR representation learning by leveraging medical ontologies such as ICD-10, SNOMED-CT, or UMLS, which encode hierarchical clinical concepts and relationships among medical entities [24]. Most approaches employ graph neural networks to model the relationships between medical codes, integrating these representations into the structured EHR embedding at various stages of the representation learning pipeline [17, 25, 26]. In contrast, we do not rely on knowledge graphs to model relationships between medical variables. Instead, we represent background knowledge through a BN, which not only captures clinical dependencies among variables (as knowledge graphs do) but also enables probabilistic reasoning and prediction.

2.2 Integrating Bayesian networks and text

Bayesian networks

A BN has the ability to model complex problems involving uncertainty, while combining data with expert knowledge in an interpretable graphical structure [28]. It is made up of two parts: a Directed Acyclic Graph (DAG) modeling the relations between the variables, and a joint probability distribution factorizing into a set of conditional distributions, one for each variable. BNs have proven useful to model a wide range of

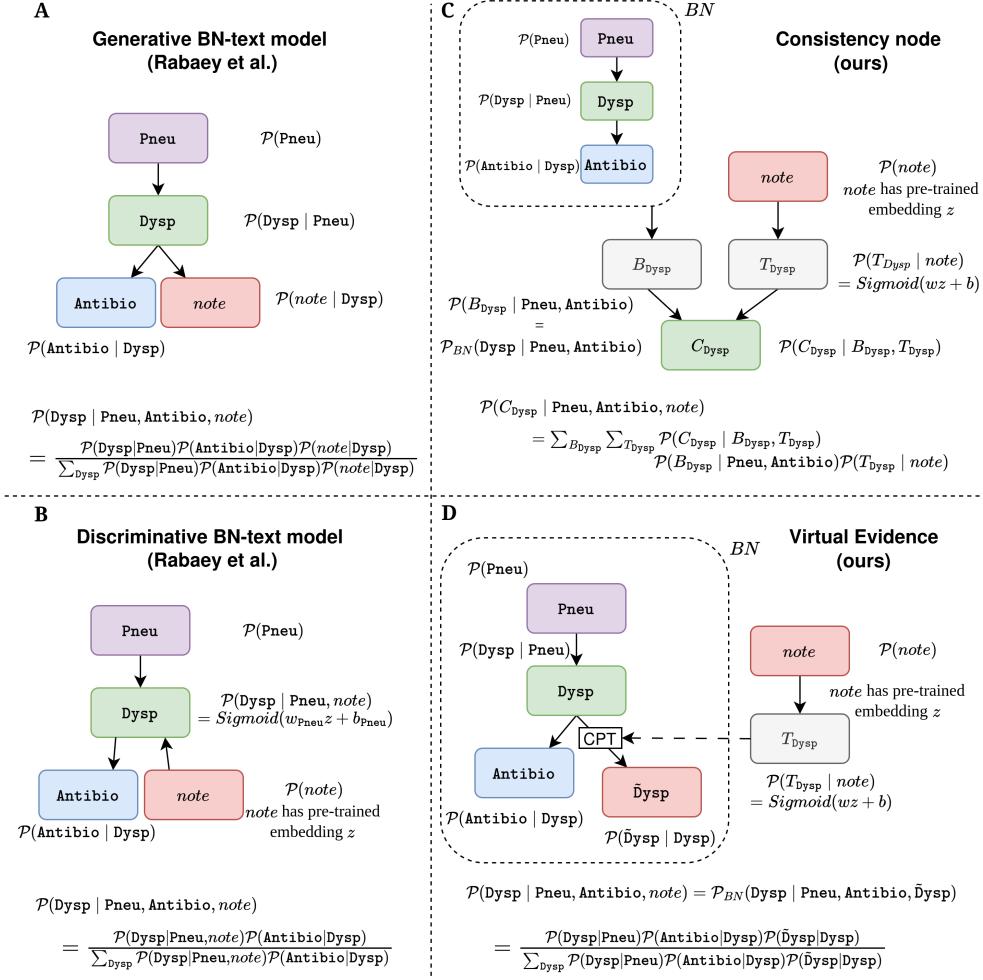


Fig. 2: Comparison of our work with Rabaey et al. [27]. In this running example, we aim to predict the probability $\mathcal{P}(\text{Dysp} \mid \text{Pneu}, \text{note})$ that a patient suffers from the symptom dyspnea (Dysp), given both tabular information (whether the patient has pneumonia, or Pneu, and was prescribed antibiotics, or Antibio) and a clinical note (note). In the generative and discriminative BN-text models proposed by Rabaey et al. [27] (**A** and **B**), a *note* node is directly integrated in the BN, allowing one to perform Bayesian inference with mixed textual and tabular evidence. In our method, we split off the BN – where Bayesian inference is performed only over tabular evidence – and the neural text classifier, integrating their predictions through the consistency node C_{Dysp} (**C**) and virtual evidence (**D**). Our method improves on the poor performance of **A** and poor scalability, interpretability, and causal structure of **B**.

medical conditions in clinical research settings [29, 30], including respiratory diseases such as pneumonia and Covid-19 [31]. While one could automatically learn the structure of the BN from data [23, 32], a particular asset of the BN is the possibility to integrate expert knowledge in the prediction process. In our case, we indeed assume that the relations between the clinical variables have been provided by an expert, while we learn the conditional probabilities from the data. Despite their potential, the clinical adoption of BNs remains limited, largely because they struggle to handle realistic medical data, where unstructured text is abundant [33]. Consequently, there is significant potential in developing methods to effectively integrate textual information into BNs. Despite this, there has been limited prior work on integrating text data into BNs. We now zoom in on two highly relevant contributions.

***DeepProbLog* [34]**

One contribution comes from the field of Neuro-Symbolic AI, in the form of the DeepProbLog framework [34]. Here, a probabilistic logic program is extended with neural predicates, whereby a neural network is used to learn a representation of an unstructured concept (in our case, clinical text), which is further treated as a regular predicate in the program. Since BNs are a symbolic method, they can be naturally formulated as probabilistic logic programs. However, one limitation of DeepProbLog is that neural predicates can only serve as root nodes within the BN, from which probabilities propagate downward to fully symbolic, categorical variables. In other words, these neural predicates cannot have any symbolic parent variables. In our case, the tabular nodes we aim to predict – namely, the symptoms – occupy arbitrary positions within the BN and typically have multiple parent variables.

***BN-text model* [27]**

To tackle this limitation, Rabaey et al. [27] propose two approaches – a *generative* and a *discriminative* BN-text model – which integrate text directly into a clinical BN, allowing the text to be part of the evidence in the Bayesian inference procedure. Figure 2 compares both approaches to our method, making use of a simple example. In this small BN, the disease pneumonia (**Pneu**) gives rise to the symptom dyspnea (**Dysp**), with a clinician deciding whether or not to prescribe antibiotics (**Antibio**) based on the presence of this symptom in the patient. A clinical *note* may describe information about the symptom dyspnea. The first approach proposed by Rabaey et al. [27], called the generative BN-text model (Figure 2A), directly includes a text node in the BN and models its conditional distribution $\mathcal{P}(\text{note} \mid \text{Dysp})$ by learning the parameters of a multivariate Gaussian over pre-trained text embeddings. This parametric assumption proved too stringent to work in practice, leading to inferior performance of this method. Rabaey et al. [27] therefore propose a second approach, called the discriminative BN-text model (Figure 2B). This method deviates from the causal structure of the problem – the symptom dyspnea influencing the content of the text – and instead uses a neural classifier to model $\mathcal{P}(\text{Dysp} \mid \text{Pneu}, \text{note})$. In other words, it extends the DeepProbLog framework [34] by learning multiple neural predicates, conditional on the possible values of the parent variable **Pneu**. However, this approach comes with two disadvantages: (i) learning a text classifier for every

combination of parent values per symptom does not scale well when symptoms have many parents, and (ii) the effect of the diagnosis Pneu is implicitly encoded in the text classifiers, rather than explicitly injected through the conditional distributions in the BN.

As illustrated in Figure 2, our solution addresses these limitations by assuming a less stringent connection between the BN modeling the structured tabular variables on the one hand, and the neural text classifier modeling the clinical notes on the other. With *virtual evidence* (Figure 2D, Sections 2.3 and 3.4), a child node is introduced to the BN with a CPT determined by the neural classifier. With the *consistency node* (Figure 2C, Section 3.5), the individual predictions of the BN and neural classifier are integrated by a separate module trained to assign appropriate weights to their contributions. Our final method incorporates both *virtual evidence* and a *consistency node* to combine the predictions of the BN with the neural classifiers (see Figure 1 for an overview of the combined model).

2.3 Virtual evidence

Virtual evidence is an established method to incorporate the predictions of neural networks with a BN [35, 36], and can be understood as evidence with uncertainty [37]. Pearl [38] first introduced virtual evidence as a convenient way to incorporate uncertain evidence into a BN. Pearl’s method treats virtual evidence as likelihood information. As opposed to the standard hard presence or absence of a piece of evidence in a regular BN, a piece of virtual evidence is represented by a real number in $[0, 1]$ that indicates the confidence in observing a particular value for a variable V .¹

To incorporate virtual evidence into a BN, a child node is added to the variable for which virtual evidence is provided. The child node itself is “observed” – i.e., provided as standard, hard evidence – while its CPT is determined by the uncertain evidence. For example, suppose a binary variable A is observed with a probability of 0.8. To include this uncertain observation as virtual evidence, a child node \tilde{A} is added to the BN with a CPT encoding $P(\tilde{A}|A)$:²



This method of incorporating uncertain evidence retains the prior probability distribution of the BN. It is possible to provide virtual evidence for multiple variables, or even multiple pieces of virtual evidence for the same variable [39]. Furthermore, it is still sensible to query the BN for the variable(s) for which virtual evidence is provided.

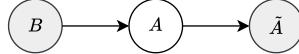
To make this concrete, consider a slight extension of our example above. Now A has a single parent B , with the following CPT:³

¹Note that virtual evidence is sometimes referred to as soft evidence. We avoid using the term soft evidence here, as it also refers to a different type of uncertain evidence [37].

²Here we use the convention of shading in the observed nodes in the BN.

³See footnote 2.

	B	$\neg B$
A	0.3	0.7



We can then use standard variable elimination [40] to calculate $P(A|B, \tilde{A}) \approx 0.6316$ (Equation 1), where both B and virtual evidence \tilde{A} are provided as evidence to the BN. Note that the final probability is lower than the uncertain evidence of 0.8 provided through \tilde{A} , because it factors in the prior probability $P(A|B)$.

$$P(A | B, \tilde{A}) = \frac{P(\tilde{A} | A) P(A | B)}{\sum_{a'} P(\tilde{A} | a') P(a' | B)} = \frac{0.8 \cdot 0.3}{(0.8 \cdot 0.3) + (0.2 \cdot 0.7)} \approx 0.6316 \quad (1)$$

As shown in Figure 1, our method uses virtual evidence but extends it with a *consistency node* (Section 3.5), further improving the calibration of the full model.

3 Methods

Following the introduction of the dataset and our experimental setup in Section 3.1, we present the fundamental components of our model: a Bayesian network (Section 3.2) and a neural network (Section 3.3). The core contribution of this work is the investigation of two strategies for integrating these models’ predictions: virtual evidence (Section 3.4) and a consistency node (Section 3.5).

3.1 Preliminaries

Dataset

The SimSUM dataset links artificial tabular patient records with artificial clinical notes describing a patient’s visit to the general practitioner’s office. By design, the clinical concepts expressed in the text (the symptoms experienced by the patient) and the tabular background information are connected through a BN representing domain knowledge. As shown in Figure 1, the BN relates two respiratory diseases (**Pneumonia** and **Common cold**) with their associated symptoms (**Dyspnea**, **Cough**, **Pain**, **Nasal symptoms** and **Fever**), as well as including underlying respiratory conditions (**Asthma**, **Smoking**, **COPD** and **Hay fever**), some background (**Season** of the year), and a treatment and outcome variable (whether **Antibiotics** were prescribed and the **#Days** the patient ended up staying at home as a result of their symptoms).⁴ All variables are binary, except for **Fever**, which has three levels (none, low and high) and **#Days**, which is discrete and bounded $(0, \dots, 15)$.

The SimSUM notes are generated in such a way that they describe the five symptoms and occasionally mention underlying respiratory conditions, but they do not explicitly mention the diagnoses. In this way, both modalities (tabular features and

⁴In the original SimSUM dataset, two other variables are included (**policy** and **self-employed**). We leave these out of our setup as they are non-clinical variables that would not be encoded in the patient record in a realistic setting.

clinical notes) contain complementary information. Every patient has one unique note, with two versions: “normal” and “compact”. We will report results on the normal notes.

Setup

As shown in Equation 2, we have a set \mathcal{X} of n patient records where each patient $X^{(i)}$ is described by their tabular record $tab^{(i)}$, their clinical note $note^{(i)}$ and their set of symptoms $s^{(i)}$.

$$\begin{aligned} \mathcal{X} &= \{X^{(0)}, \dots, X^{(n-1)} \mid X^{(i)} = \{tab^{(i)}, note^{(i)}, s^{(i)}\}\} \\ tab^{(i)} &= \{\text{Asthma} = t_0^{(i)}, \text{Smoking} = t_1^{(i)}, \text{COPD} = t_2^{(i)}, \text{Hay fever} = t_3^{(i)}, \text{Season} = t_4^{(i)}, \\ &\quad \text{Pneumonia} = t_5^{(i)}, \text{Common cold} = t_6^{(i)}, \text{Antibiotics} = t_7^{(i)}, \#\text{Days} = t_8^{(i)}\} \\ s^{(i)} &= \{\text{Dyspnea} = s_0^{(i)}, \text{Cough} = s_1^{(i)}, \text{Pain} = s_2^{(i)}, \text{Nasal} = s_3^{(i)}, \text{Fever} = s_4^{(i)}\} \\ &= \{s_j = s_j^{(i)} \mid j = 0, \dots, 4\} \end{aligned} \tag{2}$$

Here, $s_j^{(i)} \in \mathcal{V}_{s_j}$, where $\mathcal{V}_{s_j} = \{\text{yes}, \text{no}\}$ for $j = 0 \dots 3$ (symptoms Dyspnea, Cough, Pain and Nasal), and $\mathcal{V}_{s_j} = \{\text{high}, \text{low}, \text{none}\}$ for $j = 4$ (symptom Fever).

While the tabular values $t_0^{(i)}, \dots, t_8^{(i)}$ and the text note $note^{(i)}$ are available in the patient record at all times, the symptom values $s_0^{(i)}, \dots, s_4^{(i)}$ are only available during training. Out of the full set of patient records \mathcal{X} , we select a training set \mathcal{X}_{train} , and a test set \mathcal{X}_{test} . At test time, we aim to predict the probability for each of the symptoms given the tabular data and text note, i.e., the distributions $\mathcal{P}(s_j \mid tab^{(i)}, note^{(i)})$ for $X^{(i)} \in \mathcal{X}_{test}$ and $j = 0 \dots 4$.

Note that the distinction between the symptoms $s^{(i)}$ and tabular data $tab^{(i)}$ serves merely to denote $s^{(i)}$ as the targets for information extraction. While targeting $s^{(i)}$ is a natural choice given the data-generating process behind the SimSUM dataset, our method does not depend on any specific choice of target variables for extraction.

3.2 Bayesian network

We model the tabular portion of the data with a BN, where the relations between the variables (the DAG, as shown in Figure 1) are provided up-front by an expert. This DAG naturally prescribes how the joint probability distribution factorizes into conditional probability distributions (CPD), one for each variable conditional on its parents. As in Rabaey et al. [1], each CPD is learned independently by fitting it to the tabular portion of the training data \mathcal{X}_{train} . We refer to Appendix A.1 for more details on this training procedure.

When the full joint distribution defined by the BN has been learned, we can perform Bayesian inference to obtain $\mathcal{P}(s_j \mid tab^{(i)})$ for each patient $i \in \mathcal{X}_{test}$ and each symptom j , by filling in the tabular evidence available for this patient and performing variable elimination [40]. For example, for the patient in Figure 1, we would calculate $\mathcal{P}(\text{Dyspnea} = \text{yes} \mid \text{Asthma} = \text{yes}, \dots, \#\text{Days} = 5)$ by summing out the other symptoms Cough, Pain, Nasal, and Fever (which are unavailable at test time) from the

joint distribution and normalizing. From now on, we will denote the BN’s prediction for the symptom s_j as $\mathcal{P}(B_{s_j} | tab^{(i)})$.

3.3 Neural network

To model the text portion of the data, we use a lightweight neural text classifier. For simplicity, we use the same encoder and architecture as SimSUM ([1]), described below.⁵ We train separate classifiers for each symptom. At the input, the clinical note is first split into sentences. Each sentence is transformed into an embedding using the pretrained clinical representation model BioLORD-2023 [41], after which all sentence embeddings are averaged to obtain a single representation for the full note. This note embedding is then fed into a multi-layer perceptron with one hidden layer, of which the weights are trained using the cross-entropy objective over the symptom labels in \mathcal{X}_{train} . We finally obtain class probabilities by applying a Sigmoid activation (or Softmax for **Fever**, which has three classes) to the output layer. At test-time, each trained classifier can provide predictions $\mathcal{P}(s_j | note^{(i)})$ for symptom j , for each patient $i \in \mathcal{X}_{test}$. From now on, we will denote the text classifier’s prediction for the symptom s_j as $\mathcal{P}(T_{s_j} | note^{(i)})$.

3.4 Virtual evidence

To combine the probabilities of the BN and neural classifiers, we can provide the latter’s predictions to the BN as virtual evidence. That is, for a patient $i \in \mathcal{X}$, the predictions of the neural classifiers for each symptom $\mathcal{P}(T_{s_j} | note^{(i)})$ are provided to the BN as virtual evidence, as outlined in Section 2.3. We refer to the virtual evidence for a symptom s with \tilde{s} . For example, $\tilde{\text{Dyspnea}}^{(i)}$ refers to the virtual evidence for the presence of **Dyspnea** in patient i . The combined tabular and text prediction for each symptom can be noted as follows, where both the tabular evidence $tab^{(i)}$ and the virtual evidence for all symptoms are provided:

$$\mathcal{P}(s_j | tab^{(i)}, \tilde{\text{Dyspnea}}^{(i)}, \tilde{\text{Cough}}^{(i)}, \tilde{\text{Pain}}^{(i)}, \tilde{\text{Nasal}}^{(i)}, \tilde{\text{Fever}}^{(i)})$$

This probability can be obtained through variable elimination, again by summing out the other (non-virtual) symptom nodes, apart from target symptom s_j . From now on, we will denote the prediction of the BN with virtual evidence for the symptom s_j as $\mathcal{P}(V_{s_j} | tab^{(i)}, note^{(i)})$.

3.5 Consistency node

After training both the BN and the neural text classifier over \mathcal{X}_{train} , we can obtain $\mathcal{P}(B_{s_j} | tab^{(i)})$ (prediction based on tabular evidence) and $\mathcal{P}(T_{s_j} | note^{(i)})$ (prediction based on text evidence), for any patient $i \in \mathcal{X}_{test}$ and any symptom j . From now on, we will leave out the patient index i to avoid cluttering the notation. We now combine both probabilities through a consistency node C_{s_j} , as illustrated in the overview in Figure 1. The nodes B_{s_j} , T_{s_j} and C_{s_j} form a probabilistic graphical model, with joint distribution as shown in Equation 3.

⁵Note that we only intend to provide an example of a lightweight text classifier for comparison and do not require the best possible model.

$$\mathcal{P}(C_{s_j}, B_{s_j}, T_{s_j} \mid tab, note) = \mathcal{P}(B_{s_j} \mid tab) \mathcal{P}(T_{s_j} \mid note) \mathcal{P}(C_{s_j} \mid B_{s_j}, T_{s_j}) \quad (3)$$

To obtain $\mathcal{P}(C_{s_j} \mid tab, note)$, we simply need to marginalize out the BN and text classifier predictions from the joint distribution, as in Equation 4. We do this by summing over all possible values of the symptom s_j :

$$\mathcal{P}(C_{s_j} \mid tab, note) = \sum_{b'} \sum_{t'} [\mathcal{P}(B_{s_j} = b' \mid tab) \mathcal{P}(T_{s_j} = t' \mid note) \mathcal{P}(C_{s_j} \mid b', t')], \text{ with } b', t' \in V_{s_j} \quad (4)$$

The conditional distribution $\mathcal{P}(C_{s_j} \mid B_{s_j}, T_{s_j})$ can be computed over the training set \mathcal{X}_{train} .⁶ We first use the BN and neural classifier to obtain the probabilities $\mathcal{P}(B_{s_j} = b' \mid tab^{(k)})$ and $\mathcal{P}(T_{s_j} = t' \mid note^{(k)})$ for each patient $k \in \mathcal{X}_{train}$, each symptom j and each label $b', t' \in V_{s_j}$. As shown in Equations 5 and 6, we then calculate the agreement of the predicted probabilities with respect to the ground truth label c' as observed in \mathcal{X}_{train} :

$$\mathcal{P}(C_{s_j} = c' \mid b', t') = \frac{W\{c', b', t'\}}{\sum_{c''} W\{c'', b', t'\}}, \text{ with } c', c'' \in V_{s_j} \quad (5)$$

where the weights are obtained as:

$$W\{c', b', t'\} = \sum_{k \in \mathcal{X}_{train}} \mathcal{P}(B_{s_j} = b' \mid tab^{(k)}) \cdot \mathcal{P}(T_{s_j} = t' \mid note^{(k)}) \cdot \mathbb{1}[s_j^{(k)} = c'] \quad (6)$$

Intuitively, this process calculates the agreement between the BN and the text classifier on the training set \mathcal{X}_{train} . When they disagree (meaning $b' \neq t'$), one of the two models will be right more often. For example, if the text classifier agrees with the ground truth label in the training set more often, this means $\mathcal{P}(c' = t' \mid b', t') > \mathcal{P}(c' = b' \mid b', t')$. The terms containing the factor $\mathcal{P}(T_{s_j} = t' \mid note)$ will then receive a higher weight in Equation 4, pushing the prediction towards the label t' . We provide a simple numerical example in Appendix B.

From Section 3.4 we can also obtain $\mathcal{P}(V_{s_j} \mid tab, note)$, denoting the prediction of the BN with virtual evidence for the symptom s_j . In the discussion above, $\mathcal{P}(V_{s_j} \mid tab, note)$ can be used in place of $\mathcal{P}(B_{s_j} \mid tab)$, in particular in Equation 3. This leads to our final model shown in Figure 1, where we calculate the consistency between the virtual evidence-enhanced BN and the text classifier prediction, also called **V-C-BN-text** in the next section.

4 Empirical Results

We split the SimSUM dataset into a training set \mathcal{X}_{train} of 8000 samples and a test set \mathcal{X}_{test} of 2000 samples. To investigate the performance of our method in various training regimes, we subsample different training sets \mathcal{X}_{train}^n from \mathcal{X}_{train} , logarithmically sized between $n = 100$ and 8000, i.e. $n \in \{100, 187, 350, 654, 1223, 2287, 4278, 8000\}$, using 20 different seeds per n . We also use these seeds for initialization of the model parameters and weights. For more details on hyperparameter tuning, we refer to Appendix A. The BN and neural text classifier are trained on the tabular and text portion of \mathcal{X}_{train}^n , respectively, across these 20 seeds. We compare the following model architectures:

⁶To prevent overfitting on the training set, which would taint the conditional distribution, we train the neural networks using 5-fold cross-validation as described in Appendix A.2.

- **BN-only**: The BN predicts $\mathcal{P}(B_{s_j} \mid tab^{(i)})$ for every patient i , taking tabular evidence $tab^{(i)}$ as input (Section 3.2).
- **text-only**: The neural text classifier predicts $\mathcal{P}(T_{s_j} \mid note^{(i)})$ for every patient i , taking text evidence $note^{(i)}$ as input (Section 3.3).
- **V-BN-text**: The **BN-only** and **text-only** predictions are combined by using the text-only predictions as virtual evidence, obtaining the combined prediction $\mathcal{P}(V_{s_j} \mid tab^{(i)}, note^{(i)})$ (Section 3.4).
- **C-BN-text**: We learn the distribution of the consistency node on \mathcal{X}_{train}^n for the **BN-only** and **text-only** predictions, obtaining a final consistent prediction that symptom s_j is present in patient i : $\mathcal{P}(C_{s_j} \mid tab^{(i)}, note^{(i)})$ (Section 3.5).
- **V-C-BN-text**: This approach uses both virtual evidence and the consistency node, corresponding to our final model in Figure 1. We learn the distribution of the consistency node on \mathcal{X}_{train}^n for the **V-BN-text** and **text-only** predictions, obtaining the combined prediction $\mathcal{P}(VC_{s_j} \mid tab^{(i)}, note^{(i)})$ (Section 3.4 and 3.5). Note the contrast with **C-BN-text**, which uses $\mathcal{P}(B_{s_j} \mid tab^{(i)})$ rather than $\mathcal{P}(V_{s_j} \mid tab^{(i)}, note^{(i)})$.

The models **C-BN-text** and **V-BN-text**, can be viewed as ablations of the final model **V-C-BN-text**, using only one of the consistency node or virtual evidence to combine the neural classifiers with the BN.

In addition to these ablations, we consider the following additional baseline:

- **Concat-tab-text**: This early-fusion baseline concatenates the tabular features and the text embedding at the input of an MLP with the same architecture as the **text-only** classifier. We use the same implementation as described in Rabaey et al. [1], where binary variables are transformed into a one-hot encoding, and the variable `#Days` is preprocessed using standard scaling. In contrast to the other fusion models we explore, this black-box model treats the tabular and text features as a single entity, and does not involve a BN. For more details on the implementation of this baseline, we refer to Appendix A.3.

To evaluate these models, we calculate the **average precision** and **Brier score** over \mathcal{X}_{test} , across 20 seeds. We report average precision (equivalent to area under the precision-recall curve), rather than some threshold-based metrics like F1 score or accuracy, because our goal is to provide a probabilistic estimate of the symptom s_j being present in the patient, rather than a hard decision. We choose average precision over area under the ROC curve since the former is better equipped to deal with imbalanced datasets [42], which is indeed the case for the symptom labels in SimSUM. Since our classifiers for **Fever** have three classes, we report the macro average precision in that case.

We also report the Brier score [43] (equivalent to the MSE between the predicted probabilities and ground truth labels for binary symptoms) to reflect the accuracy of our models' predicted probabilities. It provides a combined measure of calibration and confidence. Like average precision, this metric quantifies the correctness of the predicted probabilities without requiring a hard threshold be set for classification.

Table 1: Average precision ($\uparrow\uparrow$) for the predictions of the **text-only** model over \mathcal{X}_{test} , averaged over 20 seeds for various training sizes n . The symptoms **Pain** and **Fever** show the most room for improvement.

	Training size n							
	100	187	350	654	1223	2287	4278	8000
dysp	92.46 \pm 1.88	94.82 \pm 1.14	95.78 \pm 0.49	96.7 \pm 0.3	97.31 \pm 0.28	97.98 \pm 0.16	98.31 \pm 0.13	98.78 \pm 0.12
cough	90.2 \pm 3.11	94.52 \pm 0.65	95.71 \pm 0.48	96.89 \pm 0.38	97.8 \pm 0.23	98.26 \pm 0.14	98.63 \pm 0.11	98.9 \pm 0.08
pain	61.81 \pm 9.03	72.52 \pm 4.21	76.48 \pm 2.61	80.4 \pm 0.95	82.5 \pm 0.85	83.77 \pm 0.88	84.71 \pm 0.89	86.08 \pm 0.24
nasal	95.12 \pm 0.59	95.84 \pm 0.56	96.57 \pm 0.25	97.06 \pm 0.17	97.43 \pm 0.18	97.79 \pm 0.11	98.0 \pm 0.1	98.16 \pm 0.03
fever	69.05 \pm 4.81	75.01 \pm 4.12	79.73 \pm 2.34	86.46 \pm 0.98	89.86 \pm 0.65	91.64 \pm 0.51	93.15 \pm 0.25	93.93 \pm 0.21

Note that while our Brier scores for the binary symptoms lie in the $[0, 1]$ range, we did not scale the score for **Fever** (ternary), so it remains in the $[0, 2]$ range.

Section 4.1 presents our main results, where we compare our multimodal consistency method with the uni-modal and multimodal baselines. Then, Section 4.2 unravels where the improvements lie by examining the models’ performance on distinct subsets of the test set. Following this, we investigate how our consistency method handles a shift in text data distribution during inference in Section 4.3. Our key takeaways are summarized in Section 4.4.

4.1 Overall model comparison

As shown in Table 1, the naive **text-only** method already performs quite well for the symptoms **Dysp**, **Cough**, and **Nasal**. Therefore, we focus our analysis on the more difficult symptoms **Pain** and **Fever** that show more room for improvement. Tables 2 and 3, respectively, report the average precision and Brier scores for these symptoms. Detailed results for the other symptoms are available in Appendix D.1.

Examining Tables 2 and 3, we note that the **BN-only** model performs sub-par to the models involving text, which is unsurprising, as this model has no access to the clinical notes containing ample detail on the presence of the symptoms. More importantly, the results show that the **V-C-BN-text** model almost always outperforms the **text-only** and **Concat-tab-text** models. In terms of Brier scores, the **V-C-BN-text** model always outperforms its ablated versions **V-BN-text** and **C-BN-text**. While the improvements in average precision and Brier score over the baselines are sometimes marginal, they are mostly significant across 20 seeds according to a paired statistical test. As we will show in Section 4.2, including knowledge from the tabular portion of the data with help of the BN, allows the **V-C-BN-text** model to flag mistakes made by the **text-only** classifier and thereby reliably improve upon its predictions, without impacting its performance on more straightforward cases.

While one might expect lower training sizes to have the most potential for improvement, as the text classifier does not have many examples to learn from in that case, this is not reflected in the results. This is because the BN’s performance suffers in these low regimes as well, rendering its predictions less reliable and negatively affecting the performance of the combined models. However, note that we do see the greatest improvements on “middling” training sizes (654, 1223, and 2287 samples), where the BN begins to fit the data quite well but the neural classifiers still struggle somewhat.

Table 2: Average precision ($\uparrow\uparrow$) for the predictions of our models over \mathcal{X}_{test} , averaged over 20 seeds for various training sizes n . The best model per training size and per symptom is highlighted in **bold**. The best baseline model for each class is underlined. Cases where a model outperforms the best baseline model significantly are indicated by * ($p < 0.05$ in a one-sided Wilcoxon signed-rank test over 20 seeds). **Change vs. baseline** compares the difference between the strongest baseline and non-baseline models.

		Training size n							
		100	187	350	654	1223	2287	4278	8000
pain	BN-only	0.3197	0.3277	0.3409	0.3464	0.35	0.3509	0.3517	0.3515
	text-only	0.6181	<u>0.7252</u>	0.7648	<u>0.804</u>	<u>0.825</u>	0.8377	0.8471	0.8608
	Concat-text-tab	0.5091	0.6537	0.7422	0.7858	0.8183	0.8412	<u>0.8559</u>	0.868
	C-BN-text	0.6135	*0.7271	*0.7723	*0.8124	*0.8323	0.8435	0.8521	0.8653
	V-BN-text	0.5667	0.7027	0.7532	*0.8194	*0.8463	*0.8598	*0.8699	*0.8826
	V-C-BN-text	0.6022	0.7244	0.7673	*0.8146	*0.8375	*0.8511	*0.8606	*0.8738
	change vs. baseline	-0.46%	+0.19%	+0.76%	+1.54%	+2.13%	+1.86%	+1.4%	+1.46%
fever	BN-only	0.4792	0.4983	0.5167	0.5309	0.5398	0.5437	0.5465	0.5474
	text-only	0.6905	<u>0.7501</u>	0.7973	<u>0.8646</u>	<u>0.8986</u>	0.9164	0.9315	0.9393
	Concat-text-tab	0.6605	0.7495	0.7939	0.8526	0.8951	<u>0.922</u>	<u>0.9381</u>	0.9501
	C-BN-text	0.66	0.7472	0.7999	*0.8705	*0.9017	0.9199	0.9345	0.9434
	V-BN-text	0.6521	0.7545	*0.8091	*0.8829	*0.9141	*0.931	*0.9475	*0.9562
	V-C-BN-text	0.6644	*0.7595	*0.8107	*0.8802	*0.9101	*0.9267	*0.9421	*0.9515
	change vs. baseline	-2.61%	+0.94%	+1.35%	+1.82%	+1.54%	+0.9%	+0.94%	+0.61%

Furthermore, in Appendix D.2, we show that a combined model which has access to the full ground truth BN (including the probabilities, rather than learning them from data), indeed shows larger improvements for smaller training sizes.

Zooming in on the **Concat-tab-text** baseline, we note that it suffers from overfitting for smaller train sizes, as a result of its naive concatenation of all features. We find that the combined models using the BN are more robust across train sizes and symptoms. These combined models are also much more interpretable, thanks to their modularity, their reliance on the expert-informed BN for the inclusion of the tabular features, and their interpretable late-fusion of the tabular and text predictions using the consistency node and virtual evidence.

4.2 Analysis of test subsets

To get a better idea of how our combined model **V-C-BN-text** manages to improve over the **text-only** baseline, we break the test set up into four distinct subsets based on whether the symptom is present in the patient (treating low + high as present for **Fever**) and whether the symptom is mentioned in the text: $\{(present, mentioned); (present, not mentioned); (not present, mentioned); (not present, not mentioned)\}$.⁷

As each subset now only contains either negative or positive examples, it is no longer possible to report the average precision. Therefore, we look only at the Brier score. Table 4 provides the results for these subsets for **pain** and **fever**. Results for the other symptoms across all subsets and training sizes are available in Appendix D.3.

⁷This is possible in SimSUM thanks to the addition of a label that indicates if a given symptom is mentioned in the text of a clinical note; Appendix C outlines the process for obtaining this label.

Table 3: Brier scores (\Downarrow) for the predictions of our models over \mathcal{X}_{test} , averaged over 20 seeds for various training sizes n . The best model per training size and per symptom is highlighted in **bold**. The best baseline model for each class is underlined. Cases where a model outperforms the best baseline model significantly are indicated by * ($p < 0.05$ in a one-sided Wilcoxon signed-rank test over 20 seeds). **Change vs. baseline** compares the difference between the strongest baseline and non-baseline models.

		Training size n							
		100	187	350	654	1223	2287	4278	8000
pain	BN-only	0.1138	0.1119	0.1091	0.1078	0.1074	0.1073	0.1071	0.1072
	text-only	0.0854	<u>0.0711</u>	0.0628	<u>0.0553</u>	<u>0.0491</u>	<u>0.0451</u>	<u>0.0426</u>	<u>0.0383</u>
	Concat-text-tab	0.0942	0.0789	0.0666	0.0623	0.0524	0.0478	0.0436	0.0393
	C-BN-text	0.0915	0.0734	0.0702	0.0604	0.0535	0.0501	0.047	0.0423
	V-BN-text	0.0963	0.0792	0.0717	0.0577	0.0511	0.046	0.0427	0.0385
	V-C-BN-text	0.0866	*0.0704	0.0647	0.0539	0.0483	0.0444	0.0414	0.0377
	change vs. baseline	+0.13%	-0.07%	+0.19%	-0.14%	-0.08%	-0.06%	-0.13%	-0.06%
fever	BN-only	0.3253	0.3153	0.3083	0.3049	0.3029	0.3016	0.301	0.3008
	text-only	<u>0.256</u>	0.2257	0.201	0.1744	<u>0.1448</u>	0.1256	0.1126	0.0984
	Concat-text-tab	0.2632	<u>0.2208</u>	<u>0.1978</u>	<u>0.1738</u>	0.1449	<u>0.1243</u>	<u>0.1081</u>	<u>0.0962</u>
	C-BN-text	0.2683	0.2253	0.2089	0.1709	0.1441	0.1261	0.1144	0.1038
	V-BN-text	0.2868	0.2327	0.2098	*0.1613	*0.1353	0.1242	0.1067	0.0986
	V-C-BN-text	0.2535	*0.2115	*0.1911	*0.1511	*0.1269	*0.1146	*0.1014	0.0941
	change vs. baseline	-0.25%	-0.93%	-0.67%	-2.27%	-1.79%	-0.96%	-0.68%	-0.22%

As seen in Table 4, **V-C-BN-text** reliably improves over **text-only** on the *not present, mentioned* and *not present, not mentioned* subsets, and displays varied performance on the *present, not mentioned* subset. Performance is slightly worse on the *present, mentioned* subset, but as seen in the Tables 2 and 3, **V-C-BN-text** still improves overall. (Note that the **text-only** model should be expected to make a good prediction for this subset, as the clinical note contains the required information.) These trends hold for the other symptoms and training sizes as well. Together, these results indicate that the BN provides **V-C-BN-text** with information about the prior distribution, which is biased towards the symptom(s) not being present.

The *present, not mentioned* subset – where the patient experiences a symptom which is not mentioned in the text – is of particular interest, as the BN can provide valuable complementary information in this case. Table 5 directly compares the performance of the BN and neural classifiers on this subset. As expected, the BN performs much better on this subset than the **text-only** classifiers. Intuitively, this occurs because the neural classifiers learn that a symptom is almost never present without being mentioned in the text, while the BN looks only at the other tabular data to form its prediction. The superior performance of the BN in this subset in turn should allow the combined **V-C-BN-text** model to improve more over the **text-only** baseline, as incorporating the BN’s prediction helps correct for the missing information in the text. In Appendix D.4, we break down a concrete example of **V-C-BN-text** providing a higher probability than **text-only** that **pain** is present when the symptom is not mentioned.

However, Table 4 shows that while **V-C-BN-text** performs much better than the **text-only** classifiers on the *present, not mentioned* subset for small training sizes (improving by a much larger margin than in other subsets), at larger training sizes this improvement breaks down for **pain** and **fever**. Table 5 partly explains the particularly noticeable degradation in performance for **fever**: the BN is much worse at predicting the occurrence of **fever** in this subset than the other symptoms, leaving less room for it to improve the combined **V-C-BN-text** model. The general degradation is explained by the fact that, as the training size increases, the **text-only** classifiers become more confident in their predictions, making their contributions as virtual evidence weigh more heavily. This is detrimental in the *present, not mentioned* subset where those **text-only** predictions are typically wrong.

As shown in Table 4, the consistency node **C-BN-text** significantly improves over the **text-only** classifiers for the *present, not mentioned* subset for all training sizes, while the virtual evidence-only model **V-BN-text** often performs worse than the **text-only** classifiers. **V-C-BN-text**, using both virtual evidence and the consistency node, reliably improves over **V-BN-text**, indicating the consistency node’s ability to help offset the fatal weakness of virtual evidence: confident (but wrong) virtual evidence can overwhelm the BN, leading to a more confident (and more wrong) final prediction. A more comprehensive analysis can be found in Appendix D.3.

4.3 Handling shifts in text data distribution

The performance gaps between the combined models and the **text-only** classifier in Table 2 and Table 3 are often small, which can be attributed to the limited information gap between the tabular data and what is written in the text notes in the SimSUM dataset. Furthermore, our analysis of performance on the *present, not mentioned* subset shows the potential of our method to correct for faulty predictions of the **text-only** classifiers when information is missing from the text.

To investigate how our method performs when more information is missing from the text, leaving more room for improvement over the **text-only** model, we create a new version of the test dataset, which we call \mathcal{X}_{test}^* , containing manipulated notes. In this test set, the tabular variables remain the same, but we randomly mask out sentences from the notes describing a symptom. For this, we use the annotated symptom spans that were released together with the SimSUM dataset [1]. For every note in the test set, we go over each phrase that describes any of the symptoms, and drop the sentence containing that phrase with a 50% probability. For example, the full sentence “Patient presents with low-grade fever and significant nasal symptoms” might be dropped to remove the mention of “low-grade fever” or of “significant nasal symptoms”. Note that this technique increases the gap between the information contained in the tabular portion of the data (patient background variables causing certain symptoms), and the information contained in the text portion of the data (descriptions of these symptoms in the clinical notes).

We then used the original **BN-only** and **text-only** classifiers (which were trained using the original train set \mathcal{X}_{train} with non-manipulated notes) to evaluate performance on this new test set \mathcal{X}_{test}^* . We also use the original consistency node, whose weights were set based on the original train set \mathcal{X}_{train} . Tables 6 and 7 show the average

Table 4: Brier scores ($\downarrow\downarrow$) for our models on the present vs. mentioned subsets across various training sizes. The best model per training size and per symptom is highlighted in **bold**. Cases where a model outperforms **text-only** significantly are indicated by * ($p < 0.05$ in a one-sided Wilcoxon signed-rank test over 20 seeds).

			Training size n							
			100	187	350	654	1223	2287	4278	8000
			text-only	0.3809	0.2508	0.1941	0.1393	0.1252	0.103	0.0909
<i>present, mentioned</i>	pain	C-BN-text	0.4679	0.3097	0.3086	0.2337	0.1912	0.1696	0.1489	0.1162
		V-BN-text	0.5569	0.401	0.3573	0.2387	0.1921	0.15	0.1247	0.0931
		V-C-BN-text	0.4157	0.28	0.2625	0.1915	0.1608	0.1354	0.1147	0.0885
		fever	text-only	0.7589	0.5334	0.4449	0.2503	0.1718	0.155	0.0998
	fever	C-BN-text	0.8994	0.6497	0.5861	0.3683	0.2582	0.2037	0.1579	0.1223
		V-BN-text	1.0834	0.7634	0.6551	0.3741	0.2486	0.2152	0.1294	0.0905
		V-C-BN-text	0.8151	0.5772	0.4998	0.2932	0.1965	0.1643	0.1076	0.0774
		fever	text-only	0.7786	0.8574	0.8387	0.8591	0.8822	0.8734	0.8711
<i>present, not mentioned</i>	pain	C-BN-text	*0.7251	*0.8035	*0.7825	*0.8102	*0.8236	*0.8236	*0.8206	*0.8178
		V-BN-text	0.8285	0.8962	0.8896	0.9142	0.9304	0.922	0.9188	0.9092
		V-C-BN-text	*0.724	*0.8182	*0.8038	0.8442	*0.863	0.8695	0.8682	0.8679
		fever	text-only	1.6366	1.5965	1.5683	1.4084	1.39	1.3764	1.306
	fever	C-BN-text	*1.4653	*1.4837	*1.4503	1.3786	*1.3556	*1.3157	1.2763	*1.277
		V-BN-text	1.7423	1.7089	1.701	1.5679	1.5602	1.5568	1.5109	1.5409
		V-C-BN-text	*1.511	*1.5258	*1.4959	1.4464	1.4325	1.3999	1.3855	1.4088
		fever	text-only	0.0293	0.032	0.0305	0.0275	0.0169	0.0127	0.0103
<i>not present, mentioned</i>	pain	C-BN-text	*0.0246	*0.0253	*0.0208	*0.0184	*0.0134	*0.0107	0.0089	0.007
		V-BN-text	*0.0138	*0.0144	*0.0112	*0.009	*0.0055	*0.0043	*0.0035	*0.0025
		V-C-BN-text	*0.0264	*0.0258	*0.0203	*0.0145	*0.009	*0.0063	*0.0051	*0.0036
		fever	text-only	0.0929	0.1232	0.1025	0.1117	0.0691	0.0368	0.0318
	fever	C-BN-text	*0.0728	*0.0821	*0.0697	*0.0678	*0.0467	*0.0307	*0.0244	0.0183
		V-BN-text	*0.0278	*0.0361	*0.0263	*0.0329	*0.0195	*0.01	*0.0086	*0.0057
		V-C-BN-text	*0.0724	*0.078	*0.0623	*0.0507	*0.0301	*0.0193	*0.0142	*0.0107
		fever	text-only	0.021	0.0164	0.0147	0.0131	0.0097	0.0098	0.0094
<i>not present, not mentioned</i>	pain	C-BN-text	0.0196	*0.015	*0.013	0.0106	0.0091	0.0089	0.0086	0.0082
		V-BN-text	*0.011	*0.0083	*0.0059	*0.0047	*0.0037	*0.0041	*0.0039	*0.0039
		V-C-BN-text	0.0205	*0.0147	*0.0117	*0.0081	*0.0063	*0.0058	*0.0053	*0.0049
		fever	text-only	0.0289	0.0293	0.0265	0.0527	0.05	0.0388	0.0427
	fever	C-BN-text	0.0367	0.0271	0.0278	*0.0356	*0.0359	*0.0328	*0.0331	0.0276
		V-BN-text	*0.0096	*0.0088	*0.0072	*0.0163	*0.017	*0.0122	*0.0121	*0.0076
		V-C-BN-text	0.0311	*0.0223	*0.0213	*0.0249	*0.0246	*0.0211	*0.0186	*0.0138

precision and Brier scores, respectively, of these models on \mathcal{X}_{test}^* , the set of out-of-distribution samples, for the symptoms **pain** and **fever**. These results demonstrate larger improvements than seen before in Tables 2 and 3, due to the added room for improvement in the **text-only** classifiers as a result of the text data shift in \mathcal{X}_{test}^* . This pattern also holds for the other symptoms. The full results can be found in Appendix D.5. The **V-C-BN-text** model significantly improves over the **text-only** baseline for almost all training sizes and all symptoms. This shows that by including the BN and its background knowledge, the **V-C-BN-text** model can fill in gaps of missing information in the text, allowing it to effectively handle shifts in text data distribution (compared to the training phase) during inference.

4.4 Key takeaways

Our results show that the methods combining the BN with neural classifiers outperform the **text-only** and **concat-tab-text** baselines, with the **V-C-BN-text** model performing the best overall. From our analysis of present vs. mentioned subsets, we

Table 5: Brier scores ($\downarrow\downarrow$) for BN-only vs. text-only on the *present, not mentioned* subset across various training sizes. Note that BN-only performs much better than text-only for all symptoms save fever.

		Training size n							
		100	187	350	654	1223	2287	4278	8000
dysp	BN-only	0.2892	0.2814	0.2608	0.2632	0.2582	0.2555	0.2579	0.2612
	text-only	0.8879	0.8893	0.8958	0.9192	0.8972	0.8517	0.8501	0.847
cough	BN-only	0.2019	0.1886	0.177	0.1793	0.1791	0.179	0.1803	0.1761
	text-only	0.5749	0.642	0.6555	0.644	0.66	0.6452	0.6715	0.6648
pain	BN-only	0.5822	0.5994	0.5925	0.5954	0.5975	0.5995	0.5951	0.5954
	text-only	0.7786	0.8574	0.8387	0.8591	0.8822	0.8734	0.8711	0.8674
nasal	BN-only	0.2982	0.2931	0.2868	0.2868	0.283	0.2813	0.2751	0.2827
	text-only	0.8839	0.9129	0.9198	0.9324	0.9298	0.899	0.9087	0.8948
fever	BN-only	1.1247	1.0954	1.098	1.0774	1.0818	1.0847	1.0908	1.0905
	text-only	1.6366	1.5965	1.5683	1.4084	1.39	1.3764	1.306	1.3218

Table 6: Average precision ($\uparrow\uparrow$) for the predictions of our models over the test set \mathcal{X}_{test}^* containing **manipulated text notes**. The best model per training size and per symptom is highlighted in **bold**. Cases where a model outperforms **text-only** significantly are indicated by * ($p < 0.05$ in a one-sided Wilcoxon signed-rank test over 20 seeds).

		Training size n							
		100	187	350	654	1223	2287	4278	8000
pain	text-only	0.5357	0.6195	0.6575	0.6946	0.722	0.7349	0.7422	0.7516
	C-BN-text	*0.5467	*0.6262	*0.669	*0.7031	*0.7285	*0.7404	*0.7493	*0.7595
	V-BN-text	0.5176	0.6106	0.6567	*0.714	*0.7473	*0.764	*0.775	*0.7868
	V-C-BN-text	0.538	0.6228	*0.6632	*0.7065	*0.737	*0.7528	*0.7613	*0.7723
	change vs. baseline	+1.1%	+0.67%	+1.15%	+1.94%	+2.53%	+2.91%	+3.27%	+3.52%
fever	text-only	0.6023	0.6524	0.6875	0.7361	0.7714	0.7909	0.8072	0.8161
	C-BN-text	0.5975	*0.6562	*0.6943	*0.7424	*0.7761	*0.7981	*0.8132	*0.8243
	V-BN-text	0.6047	*0.6782	*0.7202	*0.7746	*0.8086	*0.8269	*0.8439	*0.8543
	V-C-BN-text	0.6034	*0.6671	*0.7044	*0.7572	*0.7915	*0.8104	*0.829	*0.839
	change vs. baseline	+0.23%	+2.58%	+3.27%	+3.86%	+3.72%	+3.6%	+3.67%	+3.83%

found that **V-C-BN-text** reliably improves over the **text-only** classifiers in cases where the symptom is not present, indicating that incorporating the predictions of the BN better accounts for the true prior (which is biased towards a symptom not being present).

Furthermore, we found that for symptoms and training sizes where the BN performs well relative to the **text-only** classifiers, the largest improvements of the combined **V-C-BN-text** model come from cases where the symptom is present but not mentioned in the text. This hints at the ability of the BN to fill in the gaps where information is missing from the text. This ability is further demonstrated in a data

Table 7: Brier scores ($\downarrow\downarrow$) for the predictions of our models over the test set \mathcal{X}_{test}^* containing **manipulated text notes**. The best model per training size and per symptom is highlighted in **bold**. Cases where a model outperforms **text-only** significantly are indicated by * ($p < 0.05$ in a one-sided Wilcoxon signed-rank test over 20 seeds).

		Training size n							
		100	187	350	654	1223	2287	4278	8000
pain	text-only	0.0955	0.0879	0.0793	0.0753	0.0698	0.0665	0.0657	0.0621
	C-BN-text	0.0982	0.0867	0.082	0.0754	0.0703	0.0679	0.0662	0.0628
	V-BN-text	0.1043	0.0933	0.0869	0.0773	0.0733	0.0694	0.067	0.0639
	V-C-BN-text	0.0949	*0.0853	0.0789	*0.072	0.0683	0.0661	0.0641	0.0618
	change vs. baseline	-0.05%	-0.26%	-0.04%	-0.33%	-0.15%	-0.04%	-0.16%	-0.03%
fever	text-only	0.3071	0.2887	0.2695	0.2613	0.2388	0.2231	0.2134	0.2041
	C-BN-text	0.3029	*0.2743	*0.2612	*0.2401	*0.2222	*0.2102	*0.2012	*0.196
	V-BN-text	0.3233	0.2878	0.2736	*0.2437	*0.2251	0.2171	0.2051	0.2036
	V-C-BN-text	*0.294	*0.2671	*0.2515	*0.2306	*0.2131	*0.2034	*0.1949	*0.1929
	change vs. baseline	-1.31%	-2.16%	-1.8%	-3.07%	-2.57%	-1.97%	-1.85%	-1.12%

shift experiment, where information is missing from the text at a higher rate at inference time than during training. In this setting, the combined models outperform the **text-only** classifiers by a larger margin, indicating that the BN is able to help correct for the missing information.

Finally, the Brier scores of **V-C-BN-text** relative to **V-BN-text** indicate the consistency node’s ability to improve calibration over virtual evidence alone, while retaining strong overall accuracy. The difference in Brier scores are especially notable when the symptom is missing from the text (see *present, not mentioned* in Table 4), where virtual evidence is particularly susceptible to confident, incorrect predictions from the neural classifiers.

5 Conclusion

In this work, we introduced the concept of patient-level information extraction that leverages both the structured tabular features in a patient’s EHR and the unstructured clinical notes describing the patient’s symptoms. By augmenting *virtual evidence* with a *consistency node*, we achieved interpretable integration of a Bayesian network with neural text classifiers, enabling coherent and probabilistic fusion of tabular and textual information. Our method resulted in better-calibrated final predictions for the target variables that take into account the true prior encoded in the Bayesian network. At the same time, it highlighted the potential of the Bayesian network to correct for abnormal cases of missing textual data. Our method proved most effective for middling training sizes where the BN approaches its optimal performance prior to the neural classifiers, making it particularly appealing for use-cases where training data is limited.

While the current work focused on the specific use-case of predicting patient-level symptoms from tabular data and text, we foresee a broader use of our method in the future. First, any node in the Bayesian network may be the target of information extraction, as long as the text contains some information about this feature. Second, and more broadly, either of the two modalities (tabular and text) may be swapped

out for any other. Virtual evidence and the consistency node are flexible, only expecting probabilities at the input, without assuming any particular underlying method with which these probabilities are obtained. For example, another application of our combined consistency method could be the automated extraction of information from X-Ray images, where predictions of an image classifier that detects radiology features can be easily combined with a Bayesian network that includes tabular background information on the patient (such as age, previous diagnoses, etc.). Furthermore, while virtual evidence requires a Bayesian network, there is no requirement that a neural network be used to model the other modality. The consistency node is completely agnostic to the choice of models and can even be used without a Bayesian network.⁸ Finally, with minor adjustments to the consistency node, it becomes possible to include more than two modalities, while virtual evidence is inherently able to handle evidence from multiple modalities for the same node of a Bayesian network. In the previous example, radiology reports might be included as a third modality along with X-ray images and tabular background information.

Ultimately, the consistency node is a promising approach that provides an interpretable fusion of arbitrary models and modalities, while its compatibility with virtual evidence makes it particularly suitable for integration with Bayesian networks.

6 Limitations

Our method has several limitations. First, we interpret the probabilities at the output of the neural classifier as if they are a reflection of its confidence on the presence of the symptom in the text. However, this is not necessarily the case, as neural classifiers are known to have issues with calibration [44]. Still, this is offset by the consistency node, which improves calibration compared to using virtual evidence alone, as evidenced by the elevated Brier scores.

Second, we make strong assumptions on the types of conditional distributions that are learned in the BN. In our case, these assumptions match up perfectly with the true data generating process of the data as described in Rabaey et al. [1]. However, in a realistic setting, one would not have access to the true type of probability distribution for each variable in the network, and would instead need to consult an expert.

Third, it can be very challenging to come up with a DAG structure that accurately captures reality. To mitigate this, one could work with a panel of experts who iteratively improve the DAG. Furthermore, future work can focus on using (partial) structure learning algorithms [45] to learn the DAG from the data, filling in the gaps where experts are unsure, while still asking experts to validate the final DAG. Note that while the inclusion of the BN in our method might limit its generalization to broader contexts, we explicitly choose to trade in this flexibility for interpretability and expert input.

Finally, and related to the previous point, we only validated our method on a single simulated use-case. While this shows the merit of our method as a proof-of-concept, future work should focus on putting the theory into practice and applying

⁸However, we do find that having a Bayesian network as the tabular data model would be an asset in many medical use-cases, thanks to the implicit inclusion of expert knowledge.

our method to a more realistic and challenging dataset. To this end, the MIMIC-III [46] and MIMIC-IV [47] datasets come to mind.

Data and Code Availability. This paper uses the SimSUM dataset, which is freely available on Github [1]. Our code is available at <https://github.com/AdrickTench/patient-level-IE>.

Institutional Review Board (IRB). Since we conduct all experiments on a simulated dataset, our research did not require IRB approval.

Acknowledgements. Paloma Rabaey’s research is funded by the Research Foundation Flanders (FWO Vlaanderen) with grant number 1170124N. This research has also received funding from the Flemish government under the “Onderzoeksprogramma Artificiele Intelligentie (AI) Vlaanderen” programme. The authors thank Robin Manhaeve and Jaron Maene for their valuable insights during the conception of the consistency node method and their helpful feedback on early versions of the manuscript.

References

- [1] Rabaey, P., Arno, H., Heytens, S., Demeester, T.: SimSUM – Simulated Benchmark with Structured and Unstructured Medical Records. arXiv (2024). <https://arxiv.org/abs/2409.08936>
- [2] Ford, E., Carroll, J.A., Smith, H.E., Scott, D., Cassell, J.A.: Extracting information from the text of electronic medical records to improve case detection: a systematic review. *J Am Med Inform Assoc* **23**(5), 1007–1015 (2016)
- [3] Peiffer-Smadja, N., Rawson, T.M., Ahmad, R., Buchard, A., al.: Machine learning for clinical decision support in infectious diseases: A narrative review of current applications. *Clin Microbiol Infect* **26**(5), 584–595 (2020)
- [4] Rasmy, L., Xiang, Y., Xie, Z., Tao, C., Zhi, D.: Med-bert: pretrained contextualized embeddings on large-scale structured electronic health records for disease prediction. *NPJ digital medicine* **4**(1), 86 (2021)
- [5] Li, Y., Rao, S., Solares, J.R.A., Hassaine, A., Ramakrishnan, R., Canoy, D., Zhu, Y., Rahimi, K., Salimi-Khorshidi, G.: Behrt: transformer for electronic health records. *Scientific reports* **10**(1), 7155 (2020)
- [6] Lehman, E., Johnson, A.: Clinical-t5: Large language models built using mimic clinical text. *PhysioNet* (2023)
- [7] Singhal, K., Azizi, S., Tu, T., Mahdavi, S.S., Wei, J., Chung, H.W., Scales, N., Tanwani, A., Cole-Lewis, H., Pfohl, S., *et al.*: Large language models encode clinical knowledge. *Nature* **620**(7972), 172–180 (2023)

- [8] Labrak, Y., Bazoge, A., Morin, E., Gourraud, P.-A., Rouvier, M., Dufour, R.: BioMistral: A collection of open-source pretrained large language models for medical domains. In: Findings of the Association for Computational Linguistics: ACL 2024, pp. 5848–5864. Association for Computational Linguistics, Bangkok, Thailand (2024)
- [9] Quinn, T.P., Jacobs, S., Senadeera, M., Le, V., Coghlan, S.: The three ghosts of medical ai: Can the black-box present deliver? Artificial intelligence in medicine **124**, 102158 (2022)
- [10] Zhao, H., Chen, H., Yang, F., Liu, N., Deng, H., Cai, H., Wang, S., Yin, D., Du, M.: Explainability for large language models: A survey. ACM Transactions on Intelligent Systems and Technology **15**(2), 1–38 (2024)
- [11] Tian, S., Jin, Q., Yeganova, L., Lai, P.-T., Zhu, Q., Chen, X., Yang, Y., Chen, Q., Kim, W., Comeau, D.C., *et al.*: Opportunities and challenges for chatgpt and large language models in biomedicine and health. Briefings in Bioinformatics **25**(1), 493 (2024)
- [12] Rudin, C.: Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. Nature machine intelligence **1**(5), 206–215 (2019)
- [13] Sanchez, P., Voisey, J.P., Xia, T., Watson, H.I., O’Neil, A.Q., Tsafaris, S.A.: Causal machine learning for healthcare and precision medicine. Royal Society Open Science **9**(8), 220638 (2022)
- [14] Lundberg, S.M., Erion, G., Chen, H., DeGrave, A., Prutkin, J.M., Nair, B., Katz, R., Himmelfarb, J., Bansal, N., Lee, S.-I.: From local explanations to global understanding with explainable ai for trees. Nature machine intelligence **2**(1), 56–67 (2020)
- [15] Zhang, D., Yin, C., Zeng, J., Yuan, X., Zhang, P.: Combining structured and unstructured data for predictive models: a deep learning approach. BMC medical informatics and decision making **20**, 1–11 (2020)
- [16] Xu, K., Lam, M., Pang, J., Gao, X., Band, C., Mathur, P., Papay, F., Khanna, A.K., Cywinski, J.B., Maheshwari, K., *et al.*: Multimodal machine learning for automated icd coding. In: Machine Learning for Healthcare Conference, pp. 197–215 (2019). PMLR
- [17] Liu, S., Wang, X., Hou, Y., Li, G., Wang, H., Xu, H., Xiang, Y., Tang, B.: Multimodal data matters: language model pre-training over structured and unstructured electronic health records. IEEE Journal of Biomedical and Health Informatics **27**(1), 504–514 (2022)

- [18] Nguyen, T., Huynh, T., Phan, M.H., Nguyen, Q.V.H., Le Nguyen, P.: Carer-clinical reasoning-enhanced representation for temporal health risk prediction. In: Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, pp. 10392–10407 (2024)
- [19] Wang, X., Luo, J., Wang, J., Yin, Z., Cui, S., Zhong, Y., Wang, Y., Ma, F.: Hierarchical pretraining on multimodal electronic health records. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing. Conference on Empirical Methods in Natural Language Processing, vol. 2023, p. 2839 (2023). NIH Public Access
- [20] Xu, R., Shi, W., Yu, Y., Zhuang, Y., Jin, B., Wang, M.D., Ho, J., Yang, C.: RAM-EHR: Retrieval augmentation meets clinical predictions on electronic health records. In: Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pp. 754–765. Association for Computational Linguistics, Bangkok, Thailand (2024)
- [21] Choi, E., Xiao, C., Stewart, W., Sun, J.: Mime: Multilevel medical embedding of electronic health records for predictive healthcare. Advances in neural information processing systems **31**, 4552–4562 (2018)
- [22] Park, S., Bae, S., Kim, J., Kim, T., Choi, E.: Graph-text multi-modal pre-training for medical representation learning. In: Conference on Health, Inference, and Learning, pp. 261–281 (2022). PMLR
- [23] Choi, E., Xu, Z., Li, Y., Dusenberry, M., Flores, G., Xue, E., Dai, A.: Learning the graphical structure of electronic health records with graph convolutional transformer. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, pp. 606–613 (2020)
- [24] Choi, E., Bahadori, M.T., Song, L., Stewart, W.F., Sun, J.: Gram: graph-based attention model for healthcare representation learning. In: Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 787–795. Association for Computing Machinery, New York, NY, USA (2017)
- [25] Ma, F., You, Q., Xiao, H., Chitta, R., Zhou, J., Gao, J.: Kame: Knowledge-based attention model for diagnosis prediction in healthcare. In: Proceedings of the 27th ACM International Conference on Information and Knowledge Management, pp. 743–752. Association for Computing Machinery, New York, NY, USA (2018)
- [26] Ye, M., Cui, S., Wang, Y., Luo, J., Xiao, C., Ma, F.: Medpath: Augmenting health risk prediction via medical knowledge paths. In: Proceedings of the Web Conference 2021, pp. 1397–1409 (2021)
- [27] Rabaey, P., Deleu, J., Heytens, S., Demeester, T.: Clinical reasoning over tabular data and text with bayesian networks. In: International Conference on Artificial

Intelligence in Medicine, pp. 229–250 (2024). Springer

- [28] Kyrimi, E., McLachlan, S., Dube, K., Neves, M.R., Fahmi, A., Fenton, N.: A comprehensive scoping review of bayesian networks in healthcare: Past, present and future. *Artificial Intelligence in Medicine* **117**, 102108 (2021)
- [29] McLachlan, S., Dube, K., Hitman, G.A., Fenton, N.E., Kyrimi, E.: Bayesian networks in healthcare: Distribution by medical condition. *Artif Intell Med* **107**, 101912 (2020)
- [30] Arora, P., Boyne, D., Slater, J.J., Gupta, A., Brenner, D.R., Druzdzel, M.J.: Bayesian networks for risk prediction using real-world data: a tool for precision medicine. *Value in Health* **22**(4), 439–445 (2019)
- [31] Edye, E.O., Kurucz, J.F., Lois, L., Paredes, A., Piria, F., Rodríguez, J., Delgado, S.H.: Applying Bayesian networks to help physicians diagnose respiratory diseases in the context of covid-19 pandemic. In: 2021 IEEE URUCON, pp. 368–371 (2021)
- [32] Mani, S., Cooper, G.F.: Causal discovery from medical textual data. In: Proceedings of the AMIA Symposium, p. 542 (2000). American Medical Informatics Association
- [33] Kyrimi, E., Dube, K., Fenton, N., Fahmi, A., Neves, M.R., Marsh, W., McLachlan, S.: Bayesian networks in healthcare: What is preventing their adoption? *Artificial Intelligence in Medicine* **116**, 102079 (2021)
- [34] Manhaeve, R., Dumancic, S., Kimmig, A., Demeester, T., De Raedt, L.: Deep-problog: Neural probabilistic logic programming. *Advances in neural information processing systems* **31**, 103504 (2018)
- [35] Feng, X., Williams, C.K.: Training bayesian networks for image segmentation. In: Mathematical Modeling and Estimation Techniques in Computer Vision, vol. 3457, pp. 82–92 (1998). SPIE
- [36] Morgan, N., Bourlard, H.A.: Neural networks for statistical recognition of continuous speech. *Proceedings of the IEEE* **83**(5), 742–772 (1995)
- [37] Mrad, A., Delcroix, V., Maalej, M.A., Piechowiak, S., Abid, M.: Uncertain evidence in bayesian networks : Presentation and comparison on a simple example. In: *Communications in Computer and Information Science*, vol. 299, pp. 39–48 (2012). https://doi.org/10.1007/978-3-642-31718-7_5
- [38] Pearl, J.: Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. Morgan Kaufman, San Mateo, CA (1988)
- [39] Pan, R., Peng, Y., Ding, Z.: Belief update in bayesian networks using uncertain

- evidence. In: 2006 18th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'06), pp. 441–444 (2006). <https://doi.org/10.1109/ICTAI.2006.39>
- [40] Koller, D., Friedman, N.: Probabilistic Graphical Models: Principles and Techniques. Adaptive computation and machine learning. MIT Press, Cambridge, MA (2009)
- [41] Remy, F., Demuynck, K., Demeester, T.: BioLORD-2023: semantic textual representations fusing large language models and clinical knowledge graph insights. Journal of the American Medical Informatics Association **31**(9), 029 (2024)
- [42] Davis, J., Goadrich, M.: The relationship between precision-recall and roc curves. In: Proceedings of the 23rd International Conference on Machine Learning, pp. 233–240. Association for Computing Machinery, New York, NY, USA (2006)
- [43] Rufibach, K.: Use of brier score to assess binary predictions. Journal of Clinical Epidemiology **63**(8), 938–939 (2010) <https://doi.org/10.1016/j.jclinepi.2009.11.009>
- [44] Guo, C., Pleiss, G., Sun, Y., Weinberger, K.Q.: On calibration of modern neural networks. In: International Conference on Machine Learning, pp. 1321–1330. JMLR, Sydney, Australia (2017). PMLR
- [45] Scanagatta, M., Salmerón, A., Stella, F.: A survey on bayesian network structure learning from data. Progress in Artificial Intelligence **8**(4), 425–439 (2019)
- [46] Johnson, A.E., Pollard, T.J., Shen, L., Lehman, L.-w.H., Feng, M., Ghassemi, M., Moody, B., Szolovits, P., Anthony Celi, L., Mark, R.G.: Mimic-iii, a freely accessible critical care database. Scientific data **3**(1), 1–9 (2016)
- [47] Johnson, A.E., Bulgarelli, L., Shen, L., Gayles, A., Shammout, A., Horng, S., Pollard, T.J., Hao, S., Moody, B., Gow, B., *et al.*: Mimic-iv, a freely accessible electronic health record dataset. Scientific data **10**(1), 1 (2023)
- [48] Wu, Y.-C., Shih, M.-C., Tu, Y.-K.: Using normalized entropy to measure uncertainty of rankings for network meta-analyses. Medical Decision Making **41**(6), 706–713 (2021)

Appendix A Training details

A.1 Bayesian network

To model the tabular portion of the data, we use a Bayesian network. A Bayesian network is defined by a Directed Acyclic Graph (DAG), which models the relations between the variables. This DAG also prescribes how the joint distribution factorizes into conditional probability distributions, one for each variable conditional on its parents. In SimSUM, both the DAG and the probability distributions are defined by an expert, together forming a Bayesian network from which the tabular data is sampled. In a realistic setting, we cannot assume that we know the full data generating process. Instead, we assume that we can consult an expert to tell us how the variables are related, giving rise to the DAG in Figure 1, but that we need to learn the exact conditional probability distributions from data.

Formally, we model the tabular portion of the data by learning the probability distribution in Equation A1, where we abbreviated some of the variable names for ease of presentation.

$$\begin{aligned}
 & \mathcal{P}_{tab}(\text{Asthma}, \text{Smoking}, \text{COPD}, \text{Hay}, \text{Season}, \text{Pneu}, \text{Cold} \\
 & \quad \text{Dysp}, \text{Cough}, \text{Pain}, \text{Nasal}, \text{Fever}, \text{Antibio}, \#\text{Days}) \\
 & = \mathcal{P}(\text{Asthma})\mathcal{P}(\text{Smoking})\mathcal{P}(\text{COPD} \mid \text{Smoking})\mathcal{P}(\text{Season}) \\
 & \quad \mathcal{P}(\text{Hay})\mathcal{P}(\text{Cold} \mid \text{Season})\mathcal{P}(\text{Pneu} \mid \text{Asthma}, \text{COPD}, \text{Season}) \\
 & \quad \mathcal{P}(\text{Dysp} \mid \text{Asthma}, \text{Smoking}, \text{COPD}, \text{Pneu}, \text{Hay}) \\
 & \quad \mathcal{P}(\text{Cough} \mid \text{Asthma}, \text{Smoking}, \text{COPD}, \text{Pneu}, \text{Cold}) \\
 & \quad \mathcal{P}(\text{Pain} \mid \text{Cough}, \text{Pneu}, \text{COPD}, \text{Cold})\mathcal{P}(\text{Fever} \mid \text{Pneu}, \text{Cold}) \\
 & \quad \mathcal{P}(\text{Nasal} \mid \text{Cold}, \text{Hay})\mathcal{P}(\text{Antibio} \mid \text{Dysp}, \text{Cough}, \text{Pain}, \text{Fever}) \\
 & \quad \mathcal{P}(\#\text{Days} \mid \text{Antibio}, \text{Dysp}, \text{Cough}, \text{Pain}, \text{Fever}, \text{Nasal})
 \end{aligned} \tag{A1}$$

In SimSUM, the conditional probability distributions are parameterized in various ways: (i) conditional probability tables (CPTs) for the variables `Asthma`, `Smoking`, `COPD`, `Hay fever`, `Season`, `Pneumonia`, `Common cold` and `Fever`, (ii) Noisy-OR distributions for the symptoms `Dyspnea`, `Cough`, `Pain`, and `Nasal`, (iii) a logistic regression model and two Poisson regression models for the variables `Antibiotics` and `#Days`, respectively.

To obtain the full probability distribution $\mathcal{P}_{tab}(\text{Asthma}, \text{Smoking}, \dots, \text{Antibiotics}, \#\text{Days})$, we can learn the parameters for each of these conditional distributions independently, by training them on the tabular portion of the data \mathcal{X}_{train} . To this end, we follow the approach outlined in SimSUM [1]. In short, all parameters are learned through Maximum Likelihood Estimation, where the exact likelihood that is optimized depends on the particular parametrization approach. We manually tune the hyperparameters (learning rate and number of epochs) for each training set size, increasing the number of epochs and learning rate for smaller training sets. We keep the batch size fixed at 50.

When all parameters have been learned, we turn these distributions into CPTs by evaluating them for each combination of child and parent values, as described in SimSUM [1]. This allows us to do exact inference over the tabular evidence in the learned Bayesian network through variable elimination [40].

A.2 Neural text classifier

We follow the approach of Rabaey et al. [1] for training the neural text classifier. There is one hidden layer of size 256, followed by a ReLU activation. To deal with the varying training set sizes, we tune the optimal number of epochs with early stopping in a 5-fold cross-validation loop for each training set, with patience of 10 and tolerance of 10^{-3} on the cross-entropy loss over the validation set (with the maximum set to 200 epochs). We then take the median of the number of epochs at which early stopping was applied for each of these cross-validation splits, and retrain with the full training set afterwards for that number of epochs. The other hyperparameters are fixed as follows: a batch size of 50, a learning rate of 0.0005, weight decay (L2 regularization) of 10^{-5} and no dropout.

A.3 Multimodal baseline: Concat-tab-text

We use the implementation of the neural-text-tab baseline as described in Rabaey et al. [1]. According to this implementation, the tabular features are transformed into a vector representation. We use a one-hot encoding for the categorical (binary) features, and normalize the `#Days` feature using a StandardScaler. This tabular feature representation (vector of shape 9) is concatenated with the text embedding representation (vector of shape 768) and then fed into the exact same architecture as was used for the text-only baseline (only its input layer is updated from size 768 to size 777). Accordingly, we use the exact same cross-validation strategy and hyperparameters to train the model as described in Section A.2.

Appendix B Consistency node numerical example

To illustrate the functionality of the consistency node, we provide a simple numerical example. We provide ground truth labels and probabilities from a text classifier and BN for four fictional patients in Table B1.

	GT label	$\mathcal{P}(T \mid note)$	$\mathcal{P}(BN \mid tab)$
1	yes	0.1	0.7
2	no	0.1	0.2
3	no	0.1	0.6
4	yes	0.9	0.6

Table B1: Initial symptom probabilities predicted for four samples by the text classifier ($\mathcal{P}(T \mid note)$) and the BN ($\mathcal{P}(BN \mid tab)$). GT label indicates the ground truth label for the symptom in the training set.

Table B2 shows the agreement between the text classifier and BN across these 4 patients. This agreement is calculated through Equation 5 in Section 3.5. For example,

the first row of Table B2 is calculated using the probabilities provided by the text classifier and BN that the symptom is `no` in both cases of the ground truth label. Concretely, the probabilities from the cases where the ground truth is `no` are $(1 - 0.1) \times (1 - 0.2) + (1 - 0.9) \times (1 - 0.6)$ (yielding a weight of $W\{S = \text{no}, BN = \text{no}, T = \text{no}\} = 1.08$) while the probabilities where the ground truth is `yes` are $(1 - 0.1) \times (1 - 0.7) + (1 - 0.9) \times (1 - 0.6)$ (yielding a weight of $W\{S = \text{yes}, BN = \text{no}, T = \text{no}\} = 0.31$). These weights are then normalized by dividing by the sum for the row ($W\{S = \text{no}, BN = \text{no}, T = \text{no}\} + W\{S = \text{yes}, BN = \text{no}, T = \text{no}\} = 1.39$), yielding the probabilities in the first row of Table B2.

T	BN	$\mathcal{P}(C = \text{no} T, BN)$	$\mathcal{P}(C = \text{yes} T, BN)$
no	no	0.78	0.22
no	yes	0.51	0.49
yes	no	0.24	0.76
yes	yes	0.12	0.88

Table B2: Consistency node probabilities $\mathcal{P}(C | T, BN)$ obtained from predictions in Table B1.

Given the consistency node probabilities in Table B2, the final probability for the presence of a symptom, $\mathcal{P}(C = \text{yes} | tab, note)$, can be calculated using the probabilities in `yes` column, following Equation 4 in Section 3.5. For example, given probabilities of 0.1 from the text classifier and 0.8 from the BN, the final probability is calculated as follows:

$$\begin{aligned} \mathcal{P}(C = \text{yes} | tab, note) &= ((1 - 0.1) \times (1 - 0.8) \times 0.22) + ((1 - 0.1) \times 0.8 \times 0.49) \\ &\quad + (0.1 \times (1 - 0.8) \times 0.76) + (0.1 \times 0.8 \times 0.88) \approx 0.48 \end{aligned}$$

Appendix C Mentions label construction

The SimSUM dataset does not include labels indicating if a symptom is mentioned in the clinical note. We created these labels from the information available in SimSUM, augmented by manual annotation. The labels we used are made available on our Github repository, along with the rest of the code, at <https://github.com/AdrickTench/patient-level-IE>.

SimSUM includes two key pieces of information that we used to create the mentions labels: (1) a label indicating if the LLM that generated the clinical note was instructed to mention (or not mention) the symptom, and (2) pre-identified span annotations indicating portions of the clinical note that mention the symptom. In cases where (1) and (2) agree, we automatically generated the corresponding label. Disagreements between (1) and (2) were resolved through manual annotation by one of the authors.

Such disagreements can be attributed to either a failure of the LLM to follow the instructions provided in its prompt, or a failure of the span annotation process. Because of the possibility that both (1) and (2) failed on the same symptom and note

but were automatically accepted as the true label, and the possibility of human error in the manual annotation process, we cannot guarantee that our mentions labels are completely accurate. However, we expect that inaccuracies are exceptional and the labels are correct in the overwhelming majority of cases.

Appendix D Extended results

D.1 Results for all symptoms

We report the average precision (Table D3) and Brier scores (Table D4) of our models for all symptoms, as calculated on \mathcal{X}_{test} over 20 seeds for various training sizes n . We also report the overall mean of each metric over all five symptoms. We find that **V-C-BN-text** performs the best overall, providing the best Brier scores and a statistically significant improvement in average precision. While the virtual-evidence-only model **V-BN-text** sometimes provides slightly better improvements to average precision, it does not yield statistically significant improvements to the Brier score as reliably as **V-C-BN-text**.

D.2 Ground truth SimSUM Bayesian network

The tabular portion of the SimSUM dataset was generated using a fully expert-defined Bayesian network, where not only the relations between the variables were defined by an expert, but also the conditional probabilities. In a real setting, it would be unrealistic to assume that we have access to this true generating process, which is why we learn the distributions from the data in our work, which gives us the **BN-only** model. However, having access to the true ground truth distributions allows us to swap the **BN-only** model for the **GT-only** model (GT indicating ground truth), and use its predictions to build the fusion models **C-GT-text**, **V-GT-text**, and **V-C-GT-text**. We do not change anything about the **text-only** model. Table D5 and Table D6 show the results. We note that these fusion models with access to the ground truth BN **GT-only** often outperform the models using **BN-only** as the Bayesian network, especially for smaller training sizes, as the BN better matches the true data generating process.

D.3 Present vs. mentioned subsets

We report the Brier scores of the **text-only** baseline and our BN + text models on the subsets $\{\text{present, mentioned}; \text{present, not mentioned}; \text{not present, mentioned}; \text{not present, not mentioned}\}$ in Table D7, Table D8, Table D9, and Table D10 respectively.

As mentioned in Section 4.2, **V-C-BN-text** improves significantly over the **text-only** classifiers in the *present, not mentioned* subset for small training sizes, but this breaks down at larger training sizes for **pain**, **nasal**, and **fever**. We attribute this to two factors: (1) the BN does not increase in accuracy as much as the **text-only** classifiers with more training data, leading the consistency node to favor the predictions of the neural classifiers, and (2) the **text-only** classifiers become more confident at higher training sizes, leading their contributions as virtual evidence to be weighed more heavily.

Evidence of (1) can be seen in Table D3, where the performance of the BN does not improve as much as the **text-only** classifiers at higher training sizes. Note this table also helps indicate why **dysp** and **cough** continue to display larger improvements on the *present, not mentioned* than the other symptoms: the BN is better at predicting those symptoms relative to the neural classifiers at higher training sizes.

As evidence of (2), we define a standard confidence measure using normalized entropy [48]: using standard Shannon entropy to determine the uncertainty of our model's predictions:

$$H(\mathbf{p}) = - \sum_{i=1}^K p_i \log p_i \quad (\text{D2})$$

we define confidence as 1 minus the normalized entropy:

$$1 - \frac{H(\mathbf{p})}{\log K} \quad (\text{D3})$$

We report the confidence of the **text-only** classifiers (substituting $\mathcal{P}(T_{s_j} | \text{note})$ for p in Equation D3) in Table D11, which shows that the neural classifiers become more confident at larger training sizes. As noted in Section 4.2, confident virtual evidence can lead to a more confident final prediction, producing a more confidently wrong prediction in the case of faulty virtual evidence. As shown in Table D8, the consistency node can help correct for this weakness of virtual evidence at higher training sizes for the more difficult symptoms **Pain** and **Fever**.

D.4 Illustrative example

As explained in Section 4.2, many of the cases contributing to the superior performance of the **V-C-BN-text** model over the **text-only** model are those where the symptom is present in the patient, but not present in the text. We zoom in on one of these cases to illustrate this point.

Patient 8809 in SimSUM has a common cold, and as a result of this, they suffer from a cough, nasal symptoms, a low fever, and pain. Furthermore, the tabular data record reveals that the patient visited the doctor's office during winter time, and stayed home for 9 days as a result of these symptoms. From the tabular evidence (**Common cold** = yes, **Season** = winter, **#Days** = 9, all other tabular evidence = no), the Bayesian network predicts an 82.9% chance for pain. The clinical note is as follows:

```
---
**History**
The patient presented with a constant cough persisting for the past week. They confirmed experiencing mild fever fluctuations, primarily in the evenings, describing the fever as low-grade. In addition, the patient reported nasal congestion and mild rhinorrhea, which started around the same time as the cough. The patient denied any episodes of dyspnea. They have been taking over-the-counter decongestants with minimal relief.
```

****Physical Examination****

Vital signs: Temperature: 37.6°C, Blood Pressure: 120/80 mmHg, Pulse: 78 bpm, Respiratory Rate: 16 breaths per minute, and SpO2: 98% on room air. The patient appeared alert and in no acute distress. Upon auscultation, lungs were clear bilaterally with normal breath sounds and no wheezing or crackles. Nasal mucosa appeared slightly inflamed and there were no signs of significant throat erythema or exudates. Palpation of the cervical lymph nodes was unremarkable. Cardiac examination revealed regular rate and rhythm with no murmurs, rubs, or gallops. Abdomen was soft and non-tender with no apparent organomegaly.

This note does not mention pain anywhere. By training on similar examples, the neural text classifier has learned that there is a very low chance of pain in this patient (since most notes in the training set that are labeled with pain, do indeed mention pain somewhere in the note). As a result, the text classifier predicts only a 4.9% chance for pain.

By combining the BN’s prediction with the neural text classifier’s prediction using the **V-C-BN-text** model, we arrive at a probability of 40.1% for pain, which is higher than before and closer to the true label.⁹

D.5 Handling shifts in text data distribution

As explained in Section 4.3, we used our original **BN-only** and **text-only** classifiers, as well as the original consistency nodes (which were trained using the original train set \mathcal{X}_{train} with non-manipulated notes) to evaluate performance on the new test set \mathcal{X}_{test}^* with manipulated notes. Tables D12 and D13 show the results for all symptoms. The **V-C-BN-text** model significantly improves over the **text-only** baseline on both metrics for almost all training sizes and symptoms. These manipulated notes are released on our Github repository, along with the rest of the code, at <https://github.com/AdrickTench/patient-level-IE>.

⁹These results are attained for seed 2014 on training size 187.

Table D3: Average precision ($\uparrow\downarrow$) for the predictions of our models over \mathcal{X}_{test} , averaged over 20 seeds for various training sizes n . The best model per training size and per symptom is highlighted in **bold**. The best baseline model for each class is underlined. Cases where a model outperforms the best baseline model significantly are indicated by * ($p < 0.05$ in a one-sided Wilcoxon signed-rank test over 20 seeds).

		Training size n							
		100	187	350	654	1223	2287	4278	8000
dysp	BN-only	0.7625	0.7644	0.7794	0.7937	0.7972	0.7981	0.7981	0.7989
	text-only	<u>0.9246</u>	<u>0.9482</u>	<u>0.9578</u>	<u>0.967</u>	0.9731	0.9798	0.9831	<u>0.9878</u>
	Concat-text-tab	0.9127	0.9398	0.9533	0.9657	<u>0.9737</u>	<u>0.9801</u>	<u>0.9841</u>	0.987
	C-BN-text	*0.9258	0.948	0.957	0.9669	0.9736	0.9785	0.9817	0.9875
	V-BN-text	0.9186	0.9392	0.9472	0.9665	0.9741	0.9806	0.9841	*0.9882
	V-C-BN-text	*0.928	*0.9519	*0.96	*0.9701	*0.9759	*0.9819	*0.9853	*0.9892
	change vs. baseline	+0.35%	+0.36%	+0.22%	+0.31%	+0.22%	+0.18%	+0.12%	+0.13%
cough	BN-only	0.7568	0.7718	0.7844	0.7898	0.7929	0.7946	0.7947	0.7942
	text-only	<u>0.902</u>	<u>0.9452</u>	<u>0.9571</u>	<u>0.9689</u>	0.978	0.9826	0.9863	0.989
	Concat-text-tab	0.8866	0.9393	<u>0.9582</u>	<u>0.9705</u>	<u>0.9794</u>	<u>0.9851</u>	<u>0.9886</u>	<u>0.9908</u>
	C-BN-text	*0.9086	*0.9483	*0.9607	0.9701	0.9772	0.9813	0.9858	0.988
	V-BN-text	0.8988	0.9431	0.9578	*0.9727	*0.9825	*0.9864	*0.9896	*0.9918
	V-C-BN-text	*0.916	*0.9549	*0.9647	*0.9743	*0.9821	0.9857	0.989	*0.9914
	change vs. baseline	+1.4%	+0.97%	+0.65%	+0.38%	+0.3%	+0.13%	+0.1%	+0.1%
pain	BN-only	0.3197	0.3277	0.3409	0.3464	0.35	0.3509	0.3517	0.3515
	text-only	0.6181	<u>0.7252</u>	<u>0.7648</u>	<u>0.804</u>	<u>0.825</u>	0.8377	0.8471	0.8608
	Concat-text-tab	0.5091	0.6537	0.7422	0.7858	0.8183	<u>0.8412</u>	<u>0.8559</u>	0.868
	C-BN-text	0.6135	*0.7271	*0.7723	*0.8124	*0.8323	0.8435	0.8521	0.8653
	V-BN-text	0.5667	0.7027	0.7532	*0.8194	*0.8463	*0.8598	*0.8699	*0.8826
	V-C-BN-text	0.6022	0.7244	0.7673	*0.8146	*0.8375	*0.8511	*0.8606	*0.8738
	change vs. baseline	-0.46%	+0.19%	+0.76%	+1.54%	+2.13%	+1.86%	+1.4%	+1.46%
nasal	BN-only	0.6337	0.6357	0.6341	0.6303	0.634	0.6324	0.6328	0.634
	text-only	<u>0.9512</u>	<u>0.9584</u>	0.9657	0.9706	0.9743	0.9779	0.98	0.9816
	Concat-text-tab	0.9444	0.9579	<u>0.9666</u>	<u>0.973</u>	<u>0.9777</u>	<u>0.9818</u>	<u>0.9847</u>	<u>0.9869</u>
	C-BN-text	*0.9567	*0.9624	0.9686	*0.9769	*0.9799	0.9817	0.9841	0.9847
	V-BN-text	0.9508	*0.9609	0.9664	*0.976	*0.9819	*0.9861	*0.9879	*0.9885
	V-C-BN-text	*0.9547	*0.9628	*0.9699	*0.9757	*0.9798	*0.9836	*0.9858	0.9869
	change vs. baseline	+0.54%	+0.45%	+0.33%	+0.39%	+0.43%	+0.43%	+0.31%	+0.16%
fever	BN-only	0.4792	0.4983	0.5167	0.5309	0.5398	0.5437	0.5465	0.5474
	text-only	0.6905	<u>0.7501</u>	<u>0.7973</u>	<u>0.8646</u>	<u>0.8986</u>	0.9164	0.9315	0.9393
	Concat-text-tab	0.6605	0.7495	0.7939	0.8526	0.8951	<u>0.922</u>	<u>0.9381</u>	<u>0.9501</u>
	C-BN-text	0.66	0.7472	0.7999	*0.8705	*0.9017	0.9199	0.9345	0.9434
	V-BN-text	0.6521	0.7545	*0.8091	*0.8829	*0.9141	*0.931	*0.9475	*0.9562
	V-C-BN-text	0.6644	*0.7595	*0.8107	*0.8802	*0.9101	*0.9267	*0.9421	*0.9515
	change vs. baseline	-2.61%	+0.94%	+1.35%	+1.82%	+1.54%	+0.9%	+0.94%	+0.61%
mean	BN-only	0.5904	0.5996	0.6111	0.6182	0.6228	0.6239	0.6247	0.6252
	text-only	0.8173	0.8654	0.8885	<u>0.915</u>	<u>0.9298</u>	0.9389	0.9456	0.9517
	Concat-text-tab	0.7827	0.848	0.8828	0.9095	0.9289	<u>0.942</u>	0.9503	0.9566
	C-BN-text	0.8129	*0.8666	*0.8917	*0.9194	*0.933	0.941	0.9477	0.9538
	V-BN-text	0.7974	0.8601	0.8867	*0.9235	*0.9398	*0.9488	*0.9558	*0.9615
	V-C-BN-text	0.8131	*0.8707	*0.8945	*0.923	*0.9371	*0.9458	*0.9525	*0.9586
	change vs. baseline	-0.42%	+0.53%	+0.6%	+0.85%	+1.0%	+0.68%	+0.55%	+0.49%

Table D4: Brier scores (\Downarrow) for the predictions of our models over \mathcal{X}_{test} , averaged over 20 seeds for various training sizes n . The best model per training size and per symptom is highlighted in **bold**. The best baseline model for each class is underlined. Cases where a model outperforms the best baseline model significantly are indicated by * ($p < 0.05$ in a one-sided Wilcoxon signed-rank test over 20 seeds).

		Training size n							
		100	187	350	654	1223	2287	4278	8000
dysp	BN-only	0.0861	0.083	0.0797	0.0777	0.077	0.0766	0.0765	0.0763
	text-only	<u>0.0485</u>	<u>0.0365</u>	<u>0.0314</u>	<u>0.0277</u>	<u>0.023</u>	<u>0.019</u>	<u>0.0172</u>	<u>0.0128</u>
	Concat-text-tab	0.0508	0.0396	0.0339	0.0288	0.0251	0.0196	0.0172	0.0145
	C-BN-text	0.0517	0.0367	0.0324	0.0275	0.0231	0.0192	0.0175	0.0132
	V-BN-text	0.0498	0.0375	0.0333	0.0289	0.0249	0.02	0.0181	0.0145
	V-C-BN-text	*0.0468	*0.0345	*0.0301	*0.0255	*0.0218	*0.0177	*0.0156	0.0123
	change vs. baseline	-0.16%	-0.21%	-0.12%	-0.22%	-0.12%	-0.13%	-0.15%	-0.04%
cough	BN-only	0.1267	0.1209	0.1176	0.1169	0.1158	0.1155	0.1155	0.1154
	text-only	<u>0.0882</u>	<u>0.0659</u>	<u>0.0576</u>	<u>0.0494</u>	<u>0.0401</u>	<u>0.0337</u>	<u>0.027</u>	<u>0.0247</u>
	Concat-text-tab	0.0907	0.0659	<u>0.0553</u>	<u>0.0462</u>	<u>0.0398</u>	<u>0.0327</u>	<u>0.0264</u>	<u>0.0237</u>
	C-BN-text	*0.0857	*0.0642	0.0572	0.0487	0.0389	0.0328	0.0268	0.0244
	V-BN-text	*0.0831	*0.0601	*0.0529	*0.044	*0.0349	*0.0296	*0.0242	0.0229
	V-C-BN-text	*0.0779	*0.0575	*0.0503	*0.0417	*0.0327	*0.0274	*0.0228	*0.0203
	change vs. baseline	-1.04%	-0.84%	-0.5%	-0.45%	-0.71%	-0.54%	-0.36%	-0.34%
pain	BN-only	0.1138	0.1119	0.1091	0.1078	0.1074	0.1073	0.1071	0.1072
	text-only	<u>0.0854</u>	<u>0.0711</u>	<u>0.0628</u>	<u>0.0553</u>	<u>0.0491</u>	<u>0.0451</u>	<u>0.0426</u>	<u>0.0383</u>
	Concat-text-tab	0.0942	0.0789	0.0666	0.0623	0.0524	0.0478	0.0436	0.0393
	C-BN-text	0.0915	0.0734	0.0702	0.0604	0.0535	0.0501	0.047	0.0423
	V-BN-text	0.0963	0.0792	0.0717	0.0577	0.0511	0.046	0.0427	0.0385
	V-C-BN-text	0.0866	*0.0704	0.0647	<u>0.0539</u>	<u>0.0483</u>	<u>0.0444</u>	<u>0.0414</u>	<u>0.0377</u>
	change vs. baseline	+0.13%	-0.07%	+0.19%	-0.14%	-0.08%	-0.06%	-0.13%	-0.06%
nasal	BN-only	0.1216	0.1197	0.1185	0.1183	0.1175	0.1177	0.1174	0.1173
	text-only	<u>0.0424</u>	<u>0.0373</u>	<u>0.0315</u>	<u>0.0274</u>	<u>0.0238</u>	<u>0.0221</u>	<u>0.019</u>	0.0181
	Concat-text-tab	0.0513	0.0417	0.035	0.0309	0.0262	0.0224	0.02	<u>0.0177</u>
	C-BN-text	0.0441	*0.037	*0.0311	*0.0267	*0.0235	*0.0218	0.0189	0.018
	V-BN-text	0.045	*0.0344	*0.0289	0.0262	*0.0224	*0.0201	*0.018	*0.0165
	V-C-BN-text	*0.0395	*0.0336	*0.0279	*0.0244	*0.0214	*0.019	*0.0173	*0.0164
	change vs. baseline	-0.3%	-0.37%	-0.36%	-0.3%	-0.24%	-0.31%	-0.17%	-0.13%
fever	BN-only	0.3253	0.3153	0.3083	0.3049	0.3029	0.3016	0.301	0.3008
	text-only	<u>0.256</u>	0.2257	0.201	0.1744	<u>0.1448</u>	0.1256	0.1126	0.0984
	Concat-text-tab	0.2632	<u>0.2208</u>	<u>0.1978</u>	<u>0.1738</u>	0.1449	<u>0.1243</u>	<u>0.1081</u>	<u>0.0962</u>
	C-BN-text	0.2683	0.2253	0.2089	0.1709	0.1441	0.1261	0.1144	0.1038
	V-BN-text	0.2868	0.2327	0.2098	*0.1613	*0.1353	0.1242	0.1067	0.0986
	V-C-BN-text	<u>0.2535</u>	*0.2115	*0.1911	*0.1511	*0.1269	*0.1146	*0.1014	<u>0.0941</u>
	change vs. baseline	-0.25%	-0.93%	-0.67%	-2.27%	-1.79%	-0.96%	-0.68%	-0.22%
mean	BN-only	0.1547	0.1502	0.1466	0.1451	0.1441	0.1437	0.1435	0.1434
	text-only	<u>0.1041</u>	<u>0.0873</u>	<u>0.0768</u>	<u>0.0668</u>	<u>0.0562</u>	<u>0.0491</u>	0.0437	0.0384
	Concat-text-tab	0.11	0.0894	0.0777	0.0684	0.0577	0.0494	0.0431	0.0383
	C-BN-text	0.1082	0.0873	0.08	0.0668	0.0566	0.05	0.0449	0.0404
	V-BN-text	0.1122	0.0888	0.0793	*0.0636	*0.0537	0.048	*0.0419	0.0382
	V-C-BN-text	*0.1009	*0.0815	*0.0728	*0.0593	*0.0502	*0.0446	*0.0397	*0.0362
	change vs. baseline	-0.32%	-0.58%	-0.4%	-0.75%	-0.6%	-0.45%	-0.34%	-0.21%

Table D5: Average precision ($\uparrow\downarrow$) for the predictions of our models using the ground truth Bayesian network (**GT-only**) over \mathcal{X}_{test} , averaged over 20 seeds for various training sizes n . The best model per training size and per symptom is highlighted in **bold**. The best baseline model for each class is underlined. Cases where a model outperforms the best baseline model significantly are indicated by * ($p < 0.05$ in a one-sided Wilcoxon signed-rank test over 20 seeds).

		Training size n							
		100	187	350	654	1223	2287	4278	8000
dysp	GT-only	0.7996	0.7996	0.7996	0.7996	0.7996	0.7996	0.7996	0.7996
	text-only	<u>0.9246</u>	<u>0.9482</u>	<u>0.9578</u>	<u>0.967</u>	0.9731	0.9798	0.9831	<u>0.9878</u>
	Concat-text-tab	0.9127	0.9398	0.9533	0.9657	<u>0.9737</u>	<u>0.9801</u>	<u>0.9841</u>	0.987
	C-GT-text	*0.9292	*0.9499	*0.9584	0.9676	0.974	0.9786	0.982	0.9875
	V-GT-text	0.9283	*0.9514	0.9572	0.9677	0.9742	0.9805	0.984	*0.9881
	V-C-GT-text	*0.9313	*0.9533	*0.9606	*0.97	*0.9757	*0.9817	*0.9851	*0.989
	change vs. baseline	+0.67%	+0.5%	+0.29%	+0.3%	+0.2%	+0.16%	+0.1%	+0.12%
cough	GT-only	0.797	0.797	0.797	0.797	0.797	0.797	0.797	0.797
	text-only	<u>0.902</u>	<u>0.9452</u>	0.9571	0.9689	0.978	0.9826	0.9863	0.989
	Concat-text-tab	0.8866	0.9393	<u>0.9582</u>	<u>0.9705</u>	<u>0.9794</u>	<u>0.9851</u>	<u>0.9886</u>	<u>0.9908</u>
	C-GT-text	*0.9175	*0.9501	*0.9612	0.9704	0.9776	0.9816	0.986	0.9882
	V-GT-text	*0.9279	*0.9575	*0.9658	*0.9754	*0.9831	*0.9868	*0.9898	*0.9919
	V-C-GT-text	*0.9269	*0.9571	*0.9663	*0.9752	*0.9823	*0.9859	0.9891	*0.9914
	change vs. baseline	+2.59%	+1.24%	+0.81%	+0.49%	+0.37%	+0.17%	+0.12%	+0.11%
pain	GT-only	0.3603	0.3603	0.3603	0.3603	0.3603	0.3603	0.3603	0.3603
	text-only	<u>0.6181</u>	<u>0.7252</u>	<u>0.7648</u>	<u>0.804</u>	0.825	0.8377	0.8471	0.8608
	Concat-text-tab	0.5091	0.6537	0.7422	0.7858	0.8183	<u>0.8412</u>	<u>0.8559</u>	0.868
	C-GT-text	*0.6259	*0.7328	*0.7739	*0.8121	*0.8322	0.8431	0.852	0.8651
	V-GT-text	0.6119	*0.7308	0.7628	*0.8215	*0.8473	*0.8601	*0.8699	*0.8825
	V-C-GT-text	*0.6296	*0.7326	*0.7708	*0.8149	*0.8375	*0.8508	*0.8602	*0.8734
	change vs. baseline	+1.15%	+0.75%	+0.92%	+1.76%	+2.23%	+1.89%	+1.4%	+1.45%
nasal	GT-only	0.6477	0.6477	0.6477	0.6477	0.6477	0.6477	0.6477	0.6477
	text-only	<u>0.9512</u>	<u>0.9584</u>	0.9657	0.9706	0.9743	0.9779	0.98	0.9816
	Concat-text-tab	0.9444	0.9579	0.9666	0.973	0.9777	<u>0.9818</u>	<u>0.9847</u>	0.9869
	C-GT-text	*0.9559	*0.9625	0.9689	*0.9771	*0.9799	0.9817	0.9842	0.9847
	V-GT-text	*0.9538	*0.9644	*0.9715	*0.9785	*0.983	*0.9861	*0.9879	*0.9885
	V-C-GT-text	*0.9548	*0.9632	*0.9703	*0.9758	*0.9799	*0.9836	*0.9857	0.987
	change vs. baseline	+0.47%	+0.6%	+0.49%	+0.55%	+0.53%	+0.43%	+0.31%	+0.16%
fever	GT-only	0.5465	0.5465	0.5465	0.5465	0.5465	0.5465	0.5465	0.5465
	text-only	<u>0.6905</u>	<u>0.7501</u>	<u>0.7973</u>	<u>0.8646</u>	<u>0.8986</u>	0.9164	0.9315	0.9393
	Concat-text-tab	0.6605	0.7495	0.7939	0.8526	0.8951	<u>0.922</u>	<u>0.9381</u>	<u>0.9501</u>
	C-GT-text	0.681	0.7536	*0.803	*0.8711	*0.9018	0.9199	0.9344	0.9434
	V-GT-text	*0.7149	*0.7869	*0.8224	*0.8861	*0.9154	*0.9313	*0.9476	*0.9563
	V-C-GT-text	0.6936	*0.7756	*0.8179	*0.8814	*0.9103	*0.9266	*0.942	*0.9515
	change vs. baseline	+2.44%	+3.68%	+2.52%	+2.15%	+1.68%	+0.93%	+0.95%	+0.62%
mean	GT-only	0.6302	0.6302	0.6302	0.6302	0.6302	0.6302	0.6302	0.6302
	text-only	<u>0.8173</u>	<u>0.8654</u>	0.8885	<u>0.915</u>	<u>0.9298</u>	0.9389	0.9456	0.9517
	Concat-text-tab	0.7827	0.848	0.8828	0.9095	0.9289	<u>0.942</u>	<u>0.9503</u>	0.9566
	C-GT-text	*0.8219	*0.8698	*0.8931	*0.9197	*0.9331	0.941	0.9477	0.9538
	V-GT-text	*0.8273	*0.8782	*0.8959	*0.9258	*0.9406	*0.949	*0.9558	*0.9615
	V-C-GT-text	*0.8272	*0.8764	*0.8972	*0.9234	*0.9372	*0.9457	*0.9524	*0.9585
	change vs. baseline	+1.01%	+1.28%	+0.87%	+1.08%	+1.08%	+0.69%	+0.55%	+0.49%

Table D6: Brier scores (\Downarrow) for the predictions of our models using the ground truth Bayesian network (**GT-only**) over \mathcal{X}_{test} , averaged over 20 seeds for various training sizes n . The best model per training size and per symptom is highlighted in **bold**. The best baseline model for each class is underlined. Cases where a model outperforms the best baseline model significantly are indicated by * ($p < 0.05$ in a one-sided Wilcoxon signed-rank test over 20 seeds).

		Training size n							
		100	187	350	654	1223	2287	4278	8000
dysp	GT-only	0.0761	0.0761	0.0761	0.0761	0.0761	0.0761	0.0761	0.0761
	text-only	<u>0.0485</u>	<u>0.0365</u>	<u>0.0314</u>	<u>0.0277</u>	<u>0.023</u>	<u>0.019</u>	<u>0.0172</u>	<u>0.0128</u>
	Concat-text-tab	0.0508	0.0396	0.0339	0.0288	0.0251	0.0196	0.0172	0.0145
	C-GT-text	0.0508	0.0365	0.0322	0.0274	0.023	0.0192	0.0175	0.0132
	V-GT-text	0.0468	*0.0352	0.0324	0.0287	0.0248	0.02	0.0183	0.0148
	V-C-GT-text	*0.0458	*0.0337	*0.0298	*0.0254	*0.0218	*0.0178	*0.0157	0.0124
	change vs. baseline	-0.27%	-0.29%	-0.16%	-0.23%	-0.12%	-0.13%	-0.15%	-0.03%
cough	GT-only	0.1146	0.1146	0.1146	0.1146	0.1146	0.1146	0.1146	0.1146
	text-only	<u>0.0882</u>	<u>0.0659</u>	0.0576	0.0494	0.0401	0.0337	0.027	0.0247
	Concat-text-tab	0.0907	0.0659	<u>0.0553</u>	<u>0.0462</u>	<u>0.0398</u>	<u>0.0327</u>	<u>0.0264</u>	<u>0.0237</u>
	C-GT-text	*0.085	*0.064	0.0571	0.0486	0.0388	0.0328	0.0268	0.0244
	V-GT-text	*0.0727	*0.0555	*0.0496	*0.042	*0.0341	*0.0292	*0.0239	0.0224
	V-C-GT-text	*0.0742	*0.0558	*0.049	*0.0408	*0.0324	*0.0272	*0.0228	*0.0202
	change vs. baseline	-1.55%	-1.04%	-0.62%	-0.54%	-0.74%	-0.55%	-0.37%	-0.35%
pain	GT-only	0.1066	0.1066	0.1066	0.1066	0.1066	0.1066	0.1066	0.1066
	text-only	<u>0.0854</u>	<u>0.0711</u>	<u>0.0628</u>	<u>0.0553</u>	<u>0.0491</u>	<u>0.0451</u>	<u>0.0426</u>	<u>0.0383</u>
	Concat-text-tab	0.0942	0.0789	0.0666	0.0623	0.0524	0.0478	0.0436	0.0393
	C-GT-text	0.0919	0.0736	0.0702	0.0604	0.0536	0.0501	0.047	0.0423
	V-GT-text	0.0952	0.0767	0.0705	0.0573	0.0508	0.0457	0.0427	0.0384
	V-C-GT-text	0.0866	*0.0699	0.0644	0.0538	0.0482	0.0444	0.0414	*0.0377
	change vs. baseline	+0.13%	-0.12%	+0.16%	-0.14%	-0.09%	-0.07%	-0.12%	-0.07%
nasal	GT-only	0.1167	0.1167	0.1167	0.1167	0.1167	0.1167	0.1167	0.1167
	text-only	<u>0.0424</u>	<u>0.0373</u>	<u>0.0315</u>	<u>0.0274</u>	<u>0.0238</u>	<u>0.0221</u>	<u>0.019</u>	0.0181
	Concat-text-tab	0.0513	0.0417	0.035	0.0309	0.0262	0.0224	0.02	<u>0.0177</u>
	C-GT-text	0.0441	*0.037	*0.0312	*0.0268	*0.0235	*0.0218	0.0189	0.018
	V-GT-text	0.0417	*0.0331	*0.0281	*0.0255	*0.0222	*0.0201	*0.0179	*0.0164
	V-C-GT-text	*0.039	*0.0335	*0.0278	*0.0245	*0.0214	*0.019	*0.0173	*0.0164
	change vs. baseline	-0.35%	-0.42%	-0.37%	-0.29%	-0.25%	-0.31%	-0.17%	-0.14%
fever	GT-only	0.3006	0.3006	0.3006	0.3006	0.3006	0.3006	0.3006	0.3006
	text-only	<u>0.256</u>	0.2257	0.201	0.1744	<u>0.1448</u>	0.1256	0.1126	0.0984
	Concat-text-tab	0.2632	<u>0.2208</u>	<u>0.1978</u>	<u>0.1738</u>	0.1449	<u>0.1243</u>	<u>0.1081</u>	<u>0.0962</u>
	C-GT-text	0.268	0.2253	0.2088	0.1709	0.1441	0.1261	0.1144	0.1038
	V-GT-text	0.2766	0.2241	0.2054	*0.1589	*0.1338	0.124	0.1065	0.0985
	V-C-GT-text	0.2503	*0.2092	*0.1897	*0.1505	*0.1266	*0.1148	*0.1015	0.0942
	change vs. baseline	-0.58%	-1.16%	-0.81%	-2.34%	-1.81%	-0.95%	-0.67%	-0.21%
mean	GT-only	0.1429	0.1429	0.1429	0.1429	0.1429	0.1429	0.1429	0.1429
	text-only	<u>0.1041</u>	<u>0.0873</u>	<u>0.0768</u>	<u>0.0668</u>	<u>0.0562</u>	<u>0.0491</u>	0.0437	0.0384
	Concat-text-tab	0.11	0.0894	0.0777	0.0684	0.0577	0.0494	0.0431	0.0383
	C-GT-text	0.108	0.0873	0.0799	0.0668	0.0566	0.05	0.0449	0.0404
	V-GT-text	0.1066	*0.0849	0.0772	*0.0625	*0.0531	*0.0478	*0.0419	0.0381
	V-C-GT-text	*0.0992	*0.0804	*0.0721	*0.059	*0.0501	*0.0446	*0.0397	*0.0362
	change vs. baseline	-0.49%	-0.69%	-0.47%	-0.78%	-0.61%	-0.45%	-0.33%	-0.21%

Table D7: Brier scores (\Downarrow) for our models on the *present, mentioned* subset across various training sizes. The best model per training size and per symptom is highlighted in **bold**. Cases where a model outperforms **text-only** significantly are indicated by * ($p < 0.05$ in a one-sided Wilcoxon signed-rank test over 20 seeds).

		Training size n							
		100	187	350	654	1223	2287	4278	8000
dysp	text-only	0.1174	0.08	0.0719	0.0648	0.0562	0.0369	0.0355	0.0283
	C-BN-text	0.1411	0.0912	0.083	0.0695	0.0592	0.0431	0.0419	0.0309
	V-BN-text	0.1477	0.1031	0.0931	0.0831	0.0719	0.0487	0.0465	0.0364
	V-C-BN-text	0.115	0.0807	0.0742	0.064	0.0567	0.0433	0.04	0.0308
change vs. baseline		-0.24%	+0.07%	+0.23%	-0.08%	+0.05%	+0.62%	+0.45%	+0.25%
cough	text-only	0.1153	0.0659	0.0592	0.0402	0.0283	0.0182	0.017	0.0098
	C-BN-text	0.1259	0.0733	0.0634	0.0473	0.0328	0.0217	0.0183	0.0135
	V-BN-text	0.1228	0.0668	0.0563	* 0.0365	* 0.0243	* 0.0145	* 0.0139	0.0088
	V-C-BN-text	* 0.1031	0.0627	* 0.0521	0.0377	0.025	0.0161	0.0147	0.0098
change vs. baseline		-1.22%	-0.32%	-0.71%	-0.38%	-0.4%	-0.37%	-0.31%	-0.1%
pain	text-only	0.3809	0.2508	0.1941	0.1393	0.1252	0.103	0.0909	0.0706
	C-BN-text	0.4679	0.3097	0.3086	0.2337	0.1912	0.1696	0.1489	0.1162
	V-BN-text	0.5569	0.401	0.3573	0.2387	0.1921	0.15	0.1247	0.0931
	V-C-BN-text	0.4157	0.28	0.2625	0.1915	0.1608	0.1354	0.1147	0.0885
change vs. baseline		+3.48%	+2.91%	+6.84%	+5.22%	+3.56%	+3.24%	+2.38%	+1.79%
nasal	text-only	0.0521	0.0349	0.0205	0.019	0.0124	0.0065	0.0057	0.004
	C-BN-text	0.0681	0.041	0.0281	0.0222	0.0163	0.0112	0.0086	0.007
	V-BN-text	0.0924	0.0501	0.0328	0.0287	0.0178	0.009	0.0067	0.0038
	V-C-BN-text	0.054	0.037	0.0234	0.0189	0.014	0.0083	0.0067	0.0046
change vs. baseline		+0.19%	+0.21%	+0.29%	-0.01%	+0.16%	+0.18%	+0.09%	-0.03%
fever	text-only	0.7589	0.5334	0.4449	0.2503	0.1718	0.155	0.0998	0.0722
	C-BN-text	0.8994	0.6497	0.5861	0.3683	0.2582	0.2037	0.1579	0.1223
	V-BN-text	1.0834	0.7634	0.6551	0.3741	0.2486	0.2152	0.1294	0.0905
	V-C-BN-text	0.8151	0.5772	0.4998	0.2932	0.1965	0.1643	0.1076	0.0774
change vs. baseline		+5.62%	+4.38%	+5.49%	+4.29%	+2.47%	+0.93%	+0.78%	+0.52%
mean	text-only	0.2849	0.193	0.1581	0.1027	0.0788	0.0639	0.0498	0.037
	C-BN-text	0.3405	0.233	0.2139	0.1482	0.1115	0.0898	0.0751	0.058
	V-BN-text	0.4007	0.2769	0.2389	0.1522	0.1109	0.0875	0.0642	0.0465
	V-C-BN-text	0.3006	0.2075	0.1824	0.1211	0.0906	0.0735	0.0567	0.0422
change vs. baseline		+1.57%	+1.45%	+2.43%	+1.83%	+1.18%	+0.96%	+0.69%	+0.52%

Table D8: Brier scores (\mathbb{U}) for our models on the *present, not mentioned* subset across various training sizes. The best model per training size and per symptom is highlighted in **bold**. Cases where a model outperforms **text-only** significantly are indicated by * ($p < 0.05$ in a one-sided Wilcoxon signed-rank test over 20 seeds).

		Training size n							
		100	187	350	654	1223	2287	4278	8000
dysp	text-only	0.8879	0.8893	0.8958	0.9192	0.8972	0.8517	0.8501	0.847
	C-BN-text	*0.7689	*0.7967	*0.7684	*0.804	*0.7996	*0.7884	*0.7733	*0.7704
	V-BN-text	*0.7117	*0.709	*0.7026	*0.7335	*0.7047	*0.6328	*0.6443	*0.6146
	V-C-BN-text	*0.7241	*0.7441	*0.7389	*0.7826	*0.7559	*0.7284	*0.7318	*0.7038
	change vs. baseline	-17.62%	-18.03%	-19.31%	-18.57%	-19.25%	-21.89%	-20.58%	-23.24%
cough	text-only	0.5749	0.642	0.6555	0.644	0.66	0.6452	0.6715	0.6648
	C-BN-text	*0.4496	*0.5267	*0.5064	*0.535	*0.56	*0.561	*0.5787	*0.5909
	V-BN-text	*0.4654	*0.5249	*0.5272	*0.5376	*0.5587	*0.544	*0.5533	*0.5292
	V-C-BN-text	*0.4714	*0.5556	*0.5518	*0.5751	*0.5946	*0.5921	*0.6008	*0.6111
	change vs. baseline	-12.53%	-11.71%	-14.91%	-10.89%	-10.13%	-10.12%	-11.81%	-13.56%
pain	text-only	0.7786	0.8574	0.8387	0.8591	0.8822	0.8734	0.8711	0.8674
	C-BN-text	*0.7251	*0.8035	*0.7825	*0.8102	*0.8236	*0.8236	*0.8206	*0.8178
	V-BN-text	0.8285	0.8962	0.8896	0.9142	0.9304	0.922	0.9188	0.9092
	V-C-BN-text	*0.724	*0.8182	*0.8038	0.8442	*0.863	0.8695	0.8682	0.8679
	change vs. baseline	-5.46%	-5.39%	-5.62%	-4.89%	-5.86%	-4.98%	-5.05%	-4.96%
nasal	text-only	0.8839	0.9129	0.9198	0.9324	0.9298	0.899	0.9087	0.8948
	C-BN-text	*0.7955	*0.8495	*0.8482	*0.8553	*0.8609	*0.8463	*0.8503	*0.8434
	V-BN-text	*0.8381	*0.8655	*0.8625	*0.8741	*0.8785	*0.8297	*0.8262	*0.809
	V-C-BN-text	*0.8217	*0.8654	*0.8654	*0.8763	*0.8915	*0.8813	*0.8843	*0.885
	change vs. baseline	-8.84%	-6.34%	-7.15%	-7.71%	-6.89%	-6.93%	-8.25%	-8.57%
fever	text-only	1.6366	1.5965	1.5683	1.4084	1.39	1.3764	1.306	1.3218
	C-BN-text	*1.4653	*1.4837	*1.4503	1.3786	*1.3556	*1.3157	1.2763	*1.277
	V-BN-text	1.7423	1.7089	1.701	1.5679	1.5602	1.5568	1.5109	1.5409
	V-C-BN-text	*1.511	*1.5258	*1.4959	1.4464	1.4325	1.3999	1.3855	1.4088
	change vs. baseline	-17.14%	-11.28%	-11.8%	-2.98%	-3.44%	-6.07%	-2.96%	-4.48%
mean	text-only	0.9524	0.9796	0.9756	0.9526	0.9518	0.9291	0.9215	0.9191
	C-BN-text	*0.8409	*0.892	*0.8712	*0.8766	*0.8799	*0.867	*0.8598	*0.8599
	V-BN-text	*0.9172	*0.9409	*0.9366	*0.9255	*0.9265	*0.8971	*0.8907	*0.8806
	V-C-BN-text	*0.8504	*0.9018	*0.8912	*0.9049	*0.9075	*0.8942	*0.8941	*0.8953
	change vs. baseline	-11.15%	-8.76%	-10.44%	-7.6%	-7.19%	-6.21%	-6.16%	-5.92%

Table D9: Brier scores ($\downarrow\downarrow$) for our models on the *not present, mentioned* subset across various training sizes. The best model per training size and per symptom is highlighted in **bold**. Cases where a model outperforms **text-only** significantly are indicated by * ($p < 0.05$ in a one-sided Wilcoxon signed-rank test over 20 seeds).

		Training size n							
		100	187	350	654	1223	2287	4278	8000
dysp	text-only	0.0358	0.028	0.0227	0.0188	0.0137	0.0135	0.0105	0.0053
	C-BN-text	*0.0339	*0.0253	*0.0215	0.0176	0.0134	*0.0124	0.0096	0.0057
	V-BN-text	*0.0302	*0.0243	*0.0211	*0.0167	0.0137	0.0134	0.0105	0.0076
	V-C-BN-text	0.0352	*0.0258	*0.0213	0.0166	0.0128	*0.0105	*0.0076	0.005
change vs. baseline		-0.56%	-0.37%	-0.16%	-0.22%	-0.09%	-0.3%	-0.28%	-0.03%
cough	text-only	0.0929	0.0792	0.0656	0.0612	0.0473	0.0393	0.0239	0.0222
	C-BN-text	*0.0819	*0.0725	0.0648	0.0571	0.044	0.0374	0.0253	0.0217
	V-BN-text	*0.0771	*0.0666	*0.0588	*0.0538	*0.0413	*0.0362	0.0241	0.0242
	V-C-BN-text	*0.0803	*0.064	*0.056	*0.0469	*0.0348	*0.0287	*0.0192	*0.0158
change vs. baseline		-1.58%	-1.52%	-0.96%	-1.43%	-1.25%	-1.05%	-0.47%	-0.64%
pain	text-only	0.0293	0.032	0.0305	0.0275	0.0169	0.0127	0.0103	0.0066
	C-BN-text	*0.0246	*0.0253	*0.0208	*0.0184	*0.0134	*0.0107	0.0089	0.007
	V-BN-text	*0.0138	*0.0144	*0.0112	*0.009	*0.0055	*0.0043	*0.0035	*0.0025
	V-C-BN-text	*0.0264	*0.0258	*0.0203	*0.0145	*0.009	*0.0063	*0.0051	*0.0036
change vs. baseline		-1.55%	-1.76%	-1.94%	-1.85%	-1.13%	-0.84%	-0.67%	-0.41%
nasal	text-only	0.0584	0.0587	0.0514	0.0371	0.0299	0.0247	0.0155	0.0141
	C-BN-text	*0.0517	*0.0528	*0.0445	*0.0336	*0.0269	*0.0215	*0.0144	*0.0129
	V-BN-text	*0.0317	*0.0323	*0.0281	*0.0222	*0.0186	*0.0172	*0.0123	*0.011
	V-C-BN-text	*0.0465	*0.043	*0.0354	*0.0268	*0.0197	*0.0144	*0.0099	*0.009
change vs. baseline		-2.66%	-2.63%	-2.33%	-1.49%	-1.13%	-1.03%	-0.56%	-0.52%
fever	text-only	0.0929	0.1232	0.1025	0.1117	0.0691	0.0368	0.0318	0.019
	C-BN-text	*0.0728	*0.0821	*0.0697	*0.0678	*0.0467	*0.0307	*0.0244	0.0183
	V-BN-text	*0.0278	*0.0361	*0.0263	*0.0329	*0.0195	*0.01	*0.0086	*0.0057
	V-C-BN-text	*0.0724	*0.078	*0.0623	*0.0507	*0.0301	*0.0193	*0.0142	*0.0107
change vs. baseline		-6.51%	-8.71%	-7.62%	-7.88%	-4.96%	-2.68%	-2.32%	-1.34%
mean	text-only	0.0619	0.0642	0.0545	0.0513	0.0354	0.0254	0.0184	0.0135
	C-BN-text	*0.053	*0.0516	*0.0443	*0.0389	*0.0289	*0.0225	*0.0165	0.0131
	V-BN-text	*0.0361	*0.0347	*0.0291	*0.0269	*0.0197	*0.0162	*0.0118	*0.0102
	V-C-BN-text	*0.0522	*0.0473	*0.0391	*0.0311	*0.0213	*0.0159	*0.0112	*0.0088
change vs. baseline		-2.57%	-2.95%	-2.55%	-2.43%	-1.57%	-0.95%	-0.72%	-0.47%

Table D10: Brier scores (\Downarrow) for our models on the *not present, not mentioned* subset across various training sizes. The best model per training size and per symptom is highlighted in **bold**. Cases where a model outperforms **text-only** significantly are indicated by * ($p < 0.05$ in a one-sided Wilcoxon signed-rank test over 20 seeds).

		Training size n							
		100	187	350	654	1223	2287	4278	8000
dysp	text-only	0.0107	0.0091	0.0061	0.0049	0.0042	0.0044	0.0047	0.0039
	C-BN-text	0.0111	*0.0082	0.0061	0.0048	0.0042	*0.004	0.0042	0.0039
	V-BN-text	*0.0074	*0.0057	*0.0039	*0.003	*0.0027	*0.0028	*0.0031	*0.0034
	V-C-BN-text	0.0101	*0.0073	*0.0049	*0.0038	*0.0033	*0.0027	*0.0031	*0.0032
	change vs. baseline	-0.33%	-0.34%	-0.22%	-0.19%	-0.14%	-0.17%	-0.16%	-0.07%
cough	text-only	0.0158	0.0085	0.0058	0.0054	0.0057	0.0074	0.0061	0.0093
	C-BN-text	0.018	0.0106	0.0093	0.007	0.0065	0.0076	0.0068	0.0087
	V-BN-text	0.0186	0.0121	0.0098	0.0075	0.0062	0.007	*0.0055	*0.0082
	V-C-BN-text	0.0173	0.0097	0.0078	0.0057	0.005	*0.0053	*0.0045	*0.0058
	change vs. baseline	+0.15%	+0.12%	+0.2%	+0.03%	-0.07%	-0.21%	-0.16%	-0.35%
pain	text-only	0.021	0.0164	0.0147	0.0131	0.0097	0.0098	0.0094	0.0081
	C-BN-text	0.0196	*0.015	*0.013	0.0106	0.0091	0.0089	0.0086	0.0082
	V-BN-text	*0.011	*0.0083	*0.0059	*0.0047	*0.0037	*0.0041	*0.0039	*0.0039
	V-C-BN-text	0.0205	*0.0147	*0.0117	*0.0081	*0.0063	*0.0058	*0.0053	*0.0049
	change vs. baseline	-1.01%	-0.81%	-0.88%	-0.84%	-0.6%	-0.57%	-0.55%	-0.42%
nasal	text-only	0.0112	0.0086	0.0066	0.0044	0.0033	0.0053	0.0028	0.0027
	C-BN-text	0.0118	0.009	0.0068	0.0049	0.0038	0.0051	0.0033	0.0031
	V-BN-text	*0.0083	*0.0068	0.006	0.0044	0.0034	0.0049	0.0038	0.0032
	V-C-BN-text	0.0106	*0.0074	0.0056	0.004	0.0026	*0.0028	*0.0018	*0.0014
	change vs. baseline	-0.29%	-0.18%	-0.1%	-0.04%	-0.07%	-0.25%	-0.1%	-0.13%
fever	text-only	0.0289	0.0293	0.0265	0.0527	0.05	0.0388	0.0427	0.029
	C-BN-text	0.0367	0.0271	0.0278	*0.0356	*0.0359	*0.0328	*0.0331	0.0276
	V-BN-text	*0.0096	*0.0088	*0.0072	*0.0163	*0.017	*0.0122	*0.0121	*0.0076
	V-C-BN-text	0.0311	*0.0223	*0.0213	*0.0249	*0.0246	*0.0211	*0.0186	*0.0138
	change vs. baseline	-1.92%	-2.05%	-1.93%	-3.64%	-3.3%	-2.66%	-3.06%	-2.15%
mean	text-only	0.0175	0.0144	0.0119	0.0161	0.0146	0.0131	0.0131	0.0106
	C-BN-text	0.0194	0.014	0.0126	*0.0126	*0.0119	*0.0117	0.0112	0.0103
	V-BN-text	*0.011	*0.0084	*0.0065	*0.0072	*0.0066	*0.0062	*0.0057	*0.0053
	V-C-BN-text	0.0179	*0.0123	*0.0103	*0.0093	*0.0083	*0.0075	*0.0067	*0.0058
	change vs. baseline	-0.65%	-0.6%	-0.54%	-0.89%	-0.8%	-0.7%	-0.75%	-0.53%

Table D11: The confidence of the **text-only** model, as measured by $1 -$ the normalized Shannon entropy (D3) on \mathcal{X}_{test} over 20 seeds.

		Training size n							
		100	187	350	654	1223	2287	4278	8000
dysp	text-only	0.7987	0.8466	0.8499	0.8794	0.8998	0.9025	0.9022	0.9237
cough	text-only	0.6497	0.7263	0.7333	0.7833	0.8341	0.8637	0.8811	0.8914
pain	text-only	0.6589	0.7571	0.7176	0.7494	0.7833	0.7898	0.8096	0.8279
nasal	text-only	0.8164	0.872	0.876	0.8963	0.903	0.9058	0.9189	0.9243
fever	text-only	0.6781	0.7484	0.7475	0.7855	0.8196	0.8534	0.8585	0.868

Table D12: Average precision ($\uparrow\uparrow$) for the predictions of our models over the test set \mathcal{X}_{test}^* containing **manipulated text notes**. The best model per training size and per symptom is highlighted in **bold**. Cases where a model outperforms **text-only** significantly are indicated by * ($p < 0.05$ in a one-sided Wilcoxon signed-rank test over 20 seeds).

		Training size n							
		100	187	350	654	1223	2287	4278	8000
dysp	text-only	0.8762	0.9068	0.9206	0.9342	0.943	0.9519	0.9552	0.961
	C-BN-text	*0.8891	*0.9122	*0.9245	*0.9366	*0.9455	0.9515	0.9551	0.9616
	V-BN-text	*0.8905	*0.9114	0.9204	*0.94	*0.948	*0.9553	*0.9587	*0.9643
	V-C-BN-text	*0.8915	*0.9172	*0.9278	*0.9404	0.9478	*0.9555	*0.9592	*0.9645
change vs. baseline		+1.53%	+1.04%	+0.72%	+0.62%	+0.5%	+0.36%	+0.41%	+0.35%
cough	text-only	0.825	0.8651	0.881	0.8992	0.9144	0.9251	0.9328	0.9345
	C-BN-text	*0.8515	*0.8847	*0.902	*0.9146	*0.9274	*0.9353	*0.9428	*0.9431
	V-BN-text	*0.8574	*0.8942	*0.9106	*0.9254	*0.938	*0.9451	*0.9519	*0.955
	V-C-BN-text	*0.8586	*0.8928	*0.9071	*0.9193	0.9313	*0.9394	*0.9471	*0.95
change vs. baseline		+3.35%	+2.92%	+2.97%	+2.63%	+2.36%	+2.0%	+1.91%	+2.05%
pain	text-only	0.5357	0.6195	0.6575	0.6946	0.722	0.7349	0.7422	0.7516
	C-BN-text	*0.5467	*0.6262	*0.669	*0.7031	*0.7285	*0.7404	*0.7493	*0.7595
	V-BN-text	0.5176	0.6106	0.6567	*0.714	*0.7473	*0.764	*0.775	*0.7868
	V-C-BN-text	0.538	0.6228	*0.6632	*0.7065	*0.737	*0.7528	*0.7613	*0.7723
change vs. baseline		+1.1%	+0.67%	+1.15%	+1.94%	+2.53%	+2.91%	+3.27%	+3.52%
nasal	text-only	0.8788	0.8869	0.9013	0.9072	0.9117	0.9097	0.9101	0.9034
	C-BN-text	*0.9032	*0.9073	*0.9205	*0.927	*0.9308	*0.9263	*0.9288	*0.9238
	V-BN-text	*0.9057	*0.9116	*0.9229	*0.9307	*0.9369	*0.9371	*0.9388	*0.9337
	V-C-BN-text	*0.8986	*0.9052	*0.919	*0.9238	*0.9288	*0.928	*0.9296	*0.9231
change vs. baseline		+2.69%	+2.47%	+2.16%	+2.35%	+2.52%	+2.74%	+2.87%	+3.03%
fever	text-only	0.6023	0.6524	0.6875	0.7361	0.7714	0.7909	0.8072	0.8161
	C-BN-text	0.5975	*0.6562	*0.6943	*0.7424	*0.7761	*0.7981	*0.8132	*0.8243
	V-BN-text	0.6047	*0.6782	*0.7202	*0.7746	*0.8086	*0.8269	*0.8439	*0.8543
	V-C-BN-text	0.6034	*0.6671	*0.7044	*0.7572	*0.7915	*0.8104	*0.829	*0.839
change vs. baseline		+0.23%	+2.58%	+3.27%	+3.86%	+3.72%	+3.6%	+3.67%	+3.83%
mean	text-only	0.7436	0.7861	0.8096	0.8342	0.8525	0.8625	0.8695	0.8733
	C-BN-text	*0.7576	*0.7973	*0.822	*0.8447	*0.8617	*0.8703	*0.8778	*0.8825
	V-BN-text	*0.7552	*0.8012	*0.8262	*0.857	*0.8758	*0.8857	*0.8937	*0.8988
	V-C-BN-text	*0.758	*0.801	*0.8243	*0.8495	*0.8673	*0.8772	*0.8853	*0.8898
change vs. baseline		+1.44%	+1.51%	+1.66%	+2.27%	+2.32%	+2.32%	+2.42%	+2.55%

Table D13: Brier scores ($\downarrow\downarrow$) for the predictions of our models over the test set \mathcal{X}_{test}^* containing **manipulated text notes**. The best model per training size and per symptom is highlighted in **bold**. Cases where a model outperforms **text-only** significantly are indicated by * ($p < 0.05$ in a one-sided Wilcoxon signed-rank test over 20 seeds).

		Training size n							
		100	187	350	654	1223	2287	4278	8000
dysp	text-only	0.062	0.051	0.0452	0.0419	0.0377	0.0337	0.032	0.028
	C-BN-text	0.063	*0.0497	*0.0443	*0.0401	*0.0365	*0.0326	*0.031	*0.0275
	V-BN-text	0.0606	*0.0492	0.0441	0.0406	0.0373	0.0336	0.0326	0.0305
	V-C-BN-text	*0.058	*0.0469	*0.0419	*0.038	*0.0349	*0.0312	*0.0296	0.0274
	change vs. baseline	-0.4%	-0.41%	-0.34%	-0.39%	-0.28%	-0.25%	-0.24%	-0.05%
cough	text-only	0.137	0.1212	0.1141	0.105	0.0973	0.0889	0.087	0.0847
	C-BN-text	*0.12	*0.1084	*0.0992	*0.094	*0.0872	*0.081	*0.0786	*0.0778
	V-BN-text	*0.1213	*0.1039	*0.0958	*0.0886	*0.0818	*0.0754	*0.0721	*0.0694
	V-C-BN-text	*0.1173	*0.105	*0.0965	*0.091	*0.0838	*0.0783	*0.0756	*0.0743
	change vs. baseline	-1.97%	-1.73%	-1.82%	-1.64%	-1.55%	-1.35%	-1.49%	-1.53%
pain	text-only	0.0955	0.0879	0.0793	0.0753	0.0698	0.0665	0.0657	0.0621
	C-BN-text	0.0982	0.0867	0.082	0.0754	0.0703	0.0679	0.0662	0.0628
	V-BN-text	0.1043	0.0933	0.0869	0.0773	0.0733	0.0694	0.067	0.0639
	V-C-BN-text	0.0949	*0.0853	0.0789	*0.072	0.0683	0.0661	0.0641	0.0618
	change vs. baseline	-0.05%	-0.26%	-0.04%	-0.33%	-0.15%	-0.04%	-0.16%	-0.03%
nasal	text-only	0.0905	0.0847	0.0767	0.0751	0.0708	0.0682	0.0677	0.0688
	C-BN-text	*0.0869	*0.0817	*0.0737	*0.0713	*0.0678	*0.0657	*0.065	*0.0661
	V-BN-text	0.0915	0.0832	0.0761	0.0758	0.0712	0.0679	0.0673	*0.0668
	V-C-BN-text	*0.0834	*0.0798	*0.0727	*0.071	*0.069	0.068	0.0683	0.0694
	change vs. baseline	-0.71%	-0.48%	-0.4%	-0.42%	-0.3%	-0.24%	-0.27%	-0.27%
fever	text-only	0.3071	0.2887	0.2695	0.2613	0.2388	0.2231	0.2134	0.2041
	C-BN-text	0.3029	*0.2743	*0.2612	*0.2401	*0.2222	*0.2102	*0.2012	*0.196
	V-BN-text	0.3233	0.2878	0.2736	*0.2437	*0.2251	0.2171	0.2051	0.2036
	V-C-BN-text	*0.294	*0.2671	*0.2515	*0.2306	*0.2131	*0.2034	*0.1949	*0.1929
	change vs. baseline	-1.31%	-2.16%	-1.8%	-3.07%	-2.57%	-1.97%	-1.85%	-1.12%
mean	text-only	0.1384	0.1267	0.117	0.1117	0.1028	0.0961	0.0932	0.0895
	C-BN-text	*0.1342	*0.1202	*0.1121	*0.1042	*0.0968	*0.0915	*0.0884	*0.0861
	V-BN-text	0.1402	*0.1235	*0.1153	*0.1052	*0.0977	*0.0927	*0.0888	*0.0868
	V-C-BN-text	*0.1295	*0.1168	*0.1083	*0.1005	*0.0938	*0.0894	*0.0865	*0.0852
	change vs. baseline	-0.89%	-0.99%	-0.87%	-1.12%	-0.9%	-0.67%	-0.67%	-0.44%