ID N. AIDR02/18793

UNIVERSITÀ CAMPUS BIO-MEDICO DI ROMA

DEPARTMENT OF ENGINEERING

UNIVERSITÀ DEL PIEMONTE ORIENTALE

DEPARTMENT OF SCIENCE AND TECHNOLOGY

Italian National Ph.D. in Artificial Intelligence
Health and Life Sciences
XXXVIII Cycle

Knowledge-Grounded Machine Learning for Biomedical Domains: Extraction, Injection and Evaluation

Supervisors
Prof. Luigi Portinale
Prof. Stefania Montani

Candidate
Christopher Irwin

January, 2026

Abstract

The adoption of machine learning in biomedical and clinical domains is frequently constrained by structural challenges: data scarcity, large dimensionality, and a misalignment between statistical optimization and established medical practice. While data-driven approaches can learn complex patterns, they may lack the transparency and reasoning capabilities required for high-stakes healthcare environments. This thesis proposes a knowledge grounded framework that bridges this gap by integrating structured domain knowledge into the machine learning (ML) pipeline through three aspects: extraction, injection, and evaluation.

First, to address knowledge injection in sequential data, this work introduces a novel Structural Positional Encoding (SPE) method for Transformer-based process monitoring. Applied to stroke management guidelines, this approach embeds clinical ontologies directly into the model architecture, yielding better performance and adherence to guidelines compared to standard baselines. Second, addressing injection in high-dimensional data, the thesis presents a Graph Representation Learning (GRL) framework for multi-omics microbiome data. By encoding taxonomic knowledge into a graph structure, this method effectively handles feature sparsity and creates an encoder for microbiome data which can then be optimized for downstream tasks.

Finally, addressing the extraction and evaluation gap, the thesis presents a framework that leverages Large Language Models (LLMs) to extract structured clinical reasoning trees from medical literature. This creates a benchmark for assessing whether LLMs reasoning follows clinical knowledge, moving evaluation beyond simple accuracy metrics. Collectively, these contributions demonstrate that explicit knowledge integration acts as a powerful inductive bias, creating systems that are not only statistically robust but also aligned with the logical structures of medical reasoning.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

Machine learning has transformed numerous scientific domains through its capacity to extract patterns from vast datasets, leveraging unprecedented computational resources that align with the strengths of deep learning architectures. However, in clinical and biomedical applications, direct deployment of these methods often proves insufficient, not due to a single limitation, but rather a confluence of persistent structural challenges. This thesis identifies three fundamental bottlenecks that constrain the effective deployment of AI in healthcare:

- **The curse of dimensionality**: a challenge arising from the inherently high-dimensional nature of medical datasets coupled with the difficulty of acquiring large, well-curated data cohorts.
- **The lack of interpretability**: a critical limitation of many current ML systems, particularly deep learning solutions, which often lack the transparency required for clinical trust and accountability.
- **The limited alignment of objectives**: the common discrepancy between purely statistical optimization goals used during model training and the practical, clinical objectives relevant to patient care and decision-making.

Rather than addressing these limitations through purely data-driven solutions, this work proposes a knowledge-grounded approach: systematically extracting, injecting, and evaluating structured medical knowledge within machine learning models.

1.1.1 The Curse of Dimensionality

Low data regime. In many biomedical domains, data is inherently scarce. While large-scale clinical databases exist, high-quality, well-annotated datasets often remain small in sample size (N). This "low N" problem is common in clinical and biomedical research, focused cohort studies, and in studies involving expensive or invasive data collection procedures. Standard models, particularly in deep learning, struggle in this regime; they tend to overfit and fail to generalize from the limited examples, leading to models that are statistically brittle and clinically unreliable.

The High-Dimensionality Challenge. Conversely, many biomedical contexts are characterized by extreme feature dimensionality. Modern multi-omics datasets (e.g. genomics, proteomics, transcriptomics, and metabolomics) often generate tens of thousands of features

(D) from relatively few patients, creating a classic "high D, low N" scenario. This imbalance leads to the "curse of dimensionality," where model performance degrades as the number of features increases relative to sample size. In high-dimensional spaces, data points become increasingly sparse, the average distance between samples grows, making it difficult for learning algorithms to identify meaningful patterns. Consequently, models struggle to distinguish useful signals from spurious correlations arising from random noise. This challenge manifests in several ways: increased risk of overfitting as models memorize noise instead of learning generalizable patterns, exponential growth in computational complexity, greater susceptibility to the multiple testing problem where chance associations appear significant, and fundamental difficulties in discovering robust, replicable biomarkers. The result is a statistical landscape where traditional machine learning approaches often fail to extract reliable clinical insights despite the apparent richness of the data.

1.1.2 Interpretability

A fundamental barrier to clinical adoption arises from the opaque nature of high-performance machine learning models. Deep learning architectures, in particular, learn high-dimensional data representations optimized solely for task performance (e.g., classification accuracy) without explicitly modeling the underlying biological or clinical processes that generate the data. While effective for predictive performance, this optimization strategy produces models with inscrutable decision-making logic. In clinical practice, this lack of transparency is not just inconvenient; it is a critical barrier to adoption. Physicians require transparent reasoning to validate predictions against their clinical expertise, identify potential model failures, satisfy regulatory requirements for medical devices, and maintain accountability in patient care decisions. Furthermore, interpretability supports scientific goals beyond clinical deployment: it enables the discovery of novel biomarkers, facilitates hypothesis generation about disease mechanisms, and allows researchers to assess whether models have learned clinically meaningful patterns or merely exploited dataset artifacts. In sensitive domains like healthcare, where algorithmic decisions directly impact patient well-being, model interpretability becomes a fundamental requirement.

1.1.3 Alignment

Finally, most machine learning systems are optimized for statistical objectives (e.g., maximizing accuracy) that can be disconnected from clinical practice. A model may achieve high predictive accuracy yet remain clinically irrelevant if it recommends infeasible diagnostic tests, contradicts established treatment guidelines, ignores critical patient comorbidities, or

suggests interventions inconsistent with care pathways. The medical domain is rich with structured clinical knowledge: evidence-based protocols, diagnostic reasoning frameworks, and treatment hierarchies that reflect both biological mechanisms and practical constraints. By failing to incorporate this knowledge, standard machine learning approaches often overlook a powerful source of inductive bias that could address the fundamental challenges outlined above. Alignment with medical protocols is not merely a desirable property for deployment; it is a methodological lever that can guide model development, reduce reliance on large training datasets by encoding domain expertise, impose structure to mitigate overfitting in high-dimensional spaces, and enhance interpretability by grounding predictions in clinically recognized reasoning patterns.

1.2 The role of Medical Knowledge in ML systems

The central argument of this thesis is that the challenges of data scarcity, high dimensionality, lack of interpretability, and misalignment with clinical practice can be addressed and mitigated by grounding machine learning models in structured medical knowledge. Rather than relying exclusively on data-driven pattern discovery, we propose explicitly integrating the rich, curated, and validated knowledge already codified by medical experts into the learning pipeline. This approach can guide learning algorithms toward solutions that are more data-efficient, biologically plausible, and clinically actionable. This thesis presents and investigates three fundamental elements of knowledge management in clinical AI: **extraction** of structured knowledge from clinical sources, **injection** of this knowledge into machine learning architectures, and **evaluation** of models using knowledge-grounded benchmarks that assess clinical reasoning rather than mere statistical performance (Figure 1.1).

1.2.1 Extraction

Medical knowledge exists in diverse forms: biological pathways, pharmacological databases, clinical guidelines, and standardized ontologies. Yet substantial clinical expertise, such as textual protocols, practice guidelines, and expert reasoning, remains in unstructured formats and consequently is either treated as such or remains unused. Extraction thus represents a foundational step: developing methods to formalize this knowledge into computable representations such as knowledge graphs, rule sets, or semantic embeddings. This task is of increasing importance, especially given the data scarcity and "hallucination" challenges associated with training large-scale models like LLMs.

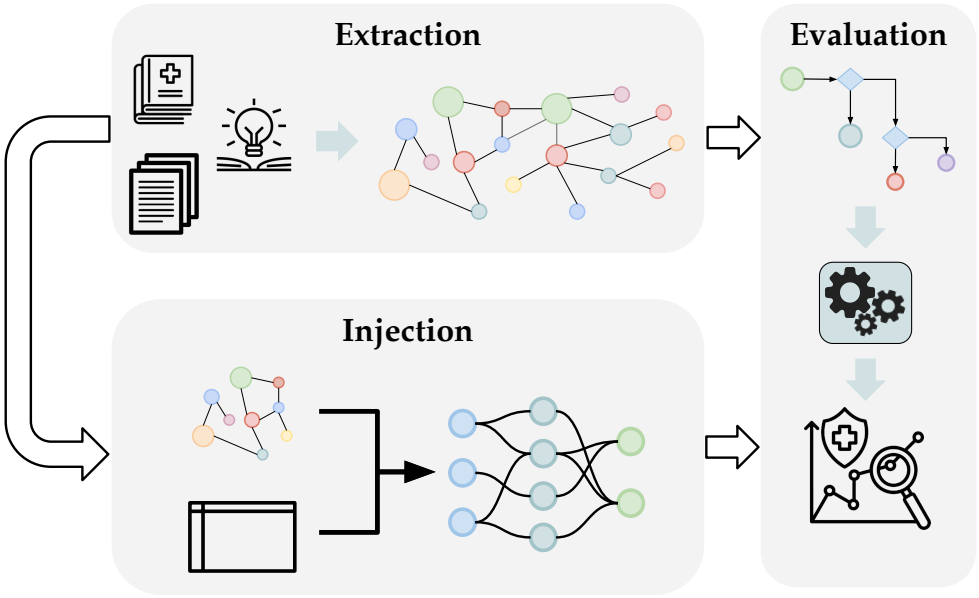

Figure 1.1: A visual summary of the core thesis contributions. The framework integrates domain knowledge through the **Extraction** of clinical reasoning structures (top), the **Injection** of this knowledge into predictive models to act as a structural inductive bias (bottom), and a comprehensive **Evaluation** phase that benchmarks model decision-making against established medical standards.

1.2.2 Injection

Knowledge injection is the process of integrating explicit domain knowledge or prior information into a Machine Learning (ML) model's structure, training, or inference. This work investigates this mechanism from three key perspectives: (i) defining the conceptual and technical requirements for knowledge integration, (ii) classifying different technical injection approaches, and (iii) assessing the measurable impact of this integration on the resulting model's performance, generalization, and interpretability. Injected knowledge serves as a powerful inductive bias: a set of well-founded assumptions that strategically constrains the model's hypothesis space [10]. This steering mechanism guides the learning process toward more plausible and robust solutions, proving especially effective in data-scarce or high-dimensional biomedical domains.

1.2.3 Evaluation

Standard evaluation practices in machine learning focus on predictive metrics (e.g., accuracy, AUC) that measure *what* a model predicts without examining *how* a conclusion is reached. This work argues for evaluation paradigms that address fundamental questions:

- Is the model's reasoning aligned with established medical knowledge?
- Does it effectively leverage injected knowledge?
- Does it bypass this information to exploit spurious correlations?
- Has the model learned clinically meaningful patterns for scientifically sound reasons?

Additionally, the scarcity of annotated medical datasets that address these issues creates a benchmarking gap: current evaluation approaches focus primarily on predictive performance without assessing whether models reason in clinically sound ways. This requires developing metrics that quantify a model's adherence to established medical knowledge and transforming structured clinical knowledge into evaluation benchmarks, thereby moving beyond statistical accuracy to assess clinical utility and reasoning quality.

1.3 Outline and contributions

This work is structured to build upon this framework:

- **Chapter 2** will analyze in-depth the state-of-the-art associated with knowledge integration in ML systems, covering all aspects (extraction, injection, evaluation) highlighted in this introduction. It will also present two case studies using more "data-driven" approaches to data scarcity and high dimensionality, which will serve as a baseline.
- **Chapter 3** and **Chapter 4** will present novel approaches and applications focused on the "injection" component. These chapters will form the core empirical contributions, demonstrating how knowledge injection can be practically implemented and validated in two different, challenging data domains.
- **Chapter 5** will then shift focus to extraction and evaluation components, specifically within the highly relevant context of clinical Large Language Models (LLMs). This chapter will present a novel approach for extracting medical knowledge and, critically, for evaluating the alignment of an LLM.
- **Chapter 6** will summarize the contributions of the thesis, highlighting limitations and further developments on the presented topics.

Chapter 2

Background and Literature Review

This chapter presents the context for knowledge-grounded machine learning in biomedical domains. First, it characterizes the fundamental data challenges in clinical AI: patient cohort scarcity, label scarcity, and high-dimensionality, and surveys data-driven solutions across different stages of the machine learning pipeline. Second, it reviews structured knowledge representations, including ontologies and knowledge graphs, and their integration into modern machine learning architectures. Finally, two case studies demonstrate both the power and limitations of purely data-driven approaches: while these methods can achieve strong performance, they lack mechanisms to explicitly incorporate medical knowledge, creating a "reasoning gap" that motivates the core contributions of this thesis.

2.1 Approaches to Data Scarcity

The approaches presented in the following focus on the four key stages of machine learning development pipeline: data preprocessing, feature engineering, model training, and model evaluation.

2.1.1 Data Preprocessing

When working with medical data, data inefficiency is often a central concern. Researchers frequently work with small patient cohorts while simultaneously, with the advent of personalized medicine, facing numerous features that can describe each patient. Additionally, a critical challenge involves missing or erroneous data, difficulties in data collection campaigns, or other practical constraints often result in datasets with substantial missing values.

Data preprocessing provides mechanisms to address these challenges by actively transforming the dataset itself. Moreover, domain knowledge can often be leveraged to guide and optimize this preprocessing stage. The principal methods for data preprocessing include:

- **Sample augmentation**: As previously noted, working with small datasets is common in clinical settings. Domain-specific rules can be employed to generate synthetic samples and augment the dataset. For instance, oversampling techniques can apply medically informed transformations to create realistic variations of existing patient records.
- **Sample removal**: Sampling errors can introduce outliers or samples that contribute only noise to the dataset. Domain expertise can inform the identification and exclusion of such samples during model construction, ensuring that the training data better reflects clinically meaningful patterns.

- **Sample modification**: Raw sample representations or scales are often suboptimal for machine learning models. For example, numerical laboratory test results are frequently interpreted and utilized by clinicians in a discretized manner, applying clinically established thresholds to convert continuous attributes into categorical ones. The way these samples are discretized represents a direct injection of domain knowledge into the dataset representation.

In all these cases, data processing involves decisions that are intrinsically tied to the domain from which the data originated. Consequently, applying these techniques inevitably incorporates characteristics of domain knowledge into both the dataset and, ultimately, the resulting model.

The case study presented in Section 2.3.1 provides an example of utilizing oversampling techniques to address data scarcity and provides an interpretability analysis describing how these techniques impact the original dataset patterns.

2.1.2 Feature Engineering

Many medical datasets contain raw features that directly report the results of clinical tests and examinations. In such cases, it is common practice to derive higher-level features through mathematical models that reduce dimensionality while improving predictive performance. Domain expertise guides the selection of which transformations are clinically meaningful, for instance, computing composite scores that align with established clinical indices. For example, Canonico et al. [22] develop a formal mathematical model of gait for Parkinson's patients monitored via wrist accelerometers over one year. The authors derive entropy-based features from raw acceleration data as a proxy for gait quality, which they then use both for activity recognition and for longitudinal analysis of motor function decline. Another example is presented by Zaccaria et al. [153], in which, using different clustering techniques, a new prognostic index for lymphoma (MLC) is constructed using a subset of the available features that describe the patients.

2.1.3 Model Training

Influencing the model training phase to accommodate external knowledge is a central theme of this thesis. The literature presents several approaches to achieve this integration. We highlight two primary strategies:

- **Knowledge as regularization**: These approaches introduce additional terms into

the loss function that penalize the model when it learns patterns inconsistent with domain knowledge. For example, monotonicity constraints can enforce that certain features maintain clinically expected relationships with outcomes, or penalty terms can discourage violations of known physiological boundaries [142].

- **Knowledge in model architecture**: This category contains more heterogeneous approaches. It is important to distinguish between methods that construct the model architecture to inherently accommodate the structure of domain knowledge, thereby constraining the model to follow this knowledge throughout training, and those that provide knowledge as a flexible prior. Examples of the first group are Jiang et al. [74], which defines a taxonomy-adaptive neural network to model microbiome-trait associations. Ruiz et al. [118] leverages a Knowledge Graph and a Graph Neural Network to condition the first layer of an MLP for tasks heavily affected by the $N \ll D$ phenomenon. Similarly, Chen and Zou [30] follows a comparable approach for gene-associated tasks but using an LLM as a backbone for generating gene descriptions in natural language. The second approach focuses on providing knowledge as a "prior" that the model may follow rather than enforce. For example, Margeloiu et al. [99] presents a setup similar to Plato [118]; however, in this case, the graphs are built from the feature values of individual samples and may be further conditioned using external knowledge graphs to improve connectivity between features.

  This second approach, which is central to the experiments presented in Chapters 3 and 4, facilitates the model's discovery of known patterns while allowing flexibility to learn novel relationships present only in the empirical data.

2.1.4 Model Evaluation

The evaluation phase of machine learning systems has been addressed across many clinical and biomedical applications, proving highly effective when domain knowledge is incorporated to quantify consistency, interpret predictions, and potentially modify the outputs of the models. The resulting systems can provide users with a measure of the model's adherence to domain knowledge, or even frameworks where predictions can be rejected entirely if they fail to satisfy certain clinical criteria. This knowledge-grounded evaluation approach ensures that models not only achieve high statistical performance but also produce clinically sound and trustworthy outputs.

Recently, these approaches have been widely adopted to overcome the limitations of Large Language Models (LLMs). As autoregressive, probabilistic systems, LLMs generate predictions through a sampling pipeline that can be leveraged to incorporate domain knowl-

edge in the form of rules that potentially reject certain predictions to improve consistency or correctness. For example, Wang and Zhou [144] introduces a score based on LLM logits to select the best generative Chain-of-Thought trace from multiple candidates produced in parallel, resulting in substantial improvements across different benchmarks. Or works like Farquhar et al. [45] which leverage semantic entropy to detect hallucinations in LLMs.

Of course, in medical and clinical applications of LLMs, these kinds of approaches are central. For example, Panagoulias et al. [108] presents a system where an LLM is augmented with explicit "rules" (domain/diagnostic rules) for primary-care diagnostic advice. Or Meng et al. [103] which uses an oracle function that judges whether LLM output satisfies attribute constraints and uses that for fine-tuning.

2.2 Knowledge Representation

The medical domain benefits from decades of systematic knowledge formalization. Comprehensive taxonomies, vocabularies, coding systems, and ontologies have been developed to represent shared medical understanding, facilitating knowledge exchange and computational processing across healthcare systems (ICD-11 [55], UMLS [15], SNOMED-CT [66], Gene Ontology [7]). Complementing these formal structures, Clinical Practice Guidelines provide evidence-based recommendations that standardize clinical decision-making and ensure care quality.

These formalized knowledge structures take various forms depending on their intended purpose.

Hierarchical taxonomies organize medical codes, diseases, procedures, and drugs into parent-child relationships, enabling classification at multiple levels of granularity. **Sequential models** and temporal structures capture the flow of clinical procedures such as the ones explored in Chapter 3, or describe the evolution of biological processes over time. **Ontologies and Knowledge Graphs** (KGs) represent complex relationships among biological entities, clinical guidelines, and medical vocabularies, with resources like PrimeKG [24] demonstrating the power of integrating multi-modal biomedical data [71]. **Rule-based systems and flowcharts** encode clinical reasoning pathways (as formalized in HealthBranches, Chapter 5) or represent diagnostic constraints and relationships (e.g., RadGraph [70] for radiology reports).

While these structures appear diverse in their surface representation, they share a common mathematical foundation: all can be conceptualized as special cases of graphs (see Figure 2.1). This observation provides a unifying framework for how to integrate biomedical

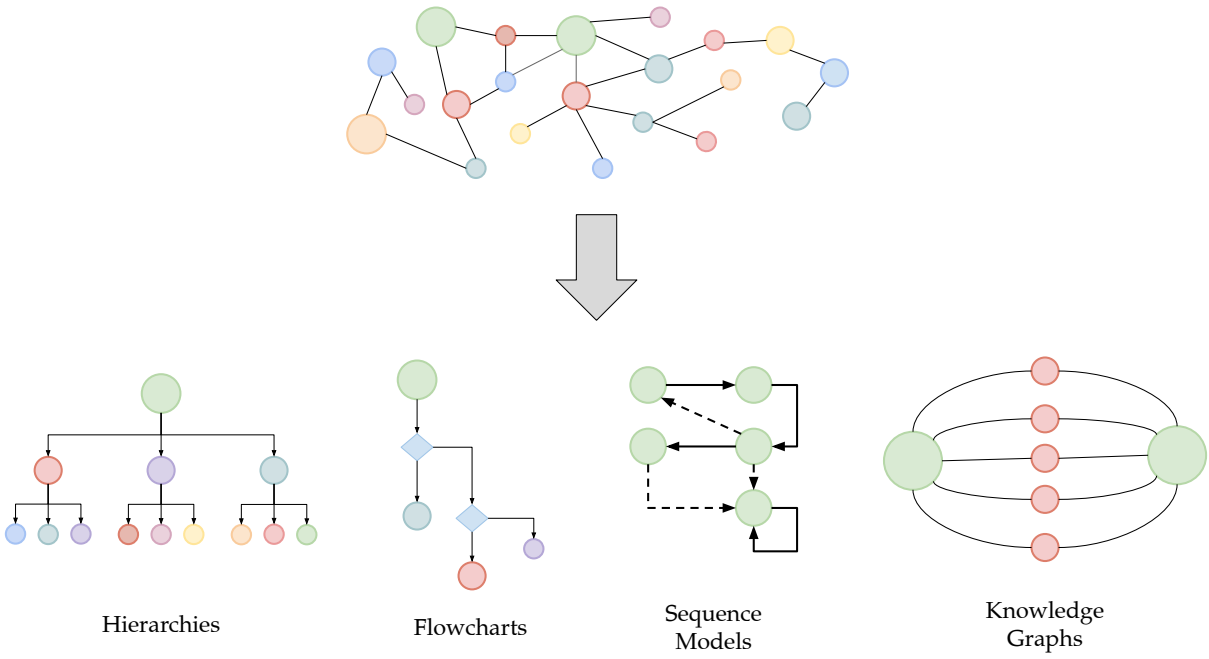

Figure 2.1: Different ways of representing knowledge with graph-based data structures.

knowledge into machine learning systems.

Graphs as a Language for Knowledge Representation. Following Battaglia et al. [10], a graph G is formally defined as a tuple (u, V, E), where u represents global attributes, V is a set of nodes (entities) with attributes, and E is a set of edges (relationships) with attributes. This mathematical abstraction serves as a powerful formal language for encoding domain knowledge, particularly in biomedicine, where relationships are central to understanding.

The expressiveness of graphs for knowledge representation stems from several key properties. First, graphs express a **relational inductive bias**: they naturally capture the inherently relational nature of biomedical systems, where entities interact through complex dependencies, from protein-protein interactions at the molecular level to patient-disease associations in clinical settings. Second, graphs provide **structured, semantic representation**: graphs decompose knowledge into discrete, machine-readable components with explicit semantic labels. A triple ⟨subject, predicate, object⟩ represents a single, verifiable fact, enabling unambiguous interpretation. Third, this structured representation **supports reasoning**: graph algorithms and neural architectures (such as the Graph Neural Networks employed in Chapter 4) can perform inference and derive new knowledge by propagating information through the graph's topology. Finally, graphs offer **flexibility**: diverse knowl-

edge sources from molecular databases to clinical guidelines can be integrated into a unified framework that evolves as new information becomes available.

The graph structure provides an expressive formalism for organizing information in a way that mirrors the relational nature of many real-world domains, making it a practical framework for knowledge representation and reasoning in AI.

2.3 Case Studies

This section presents two case studies that explore strategies for overcoming common dataset limitations in clinical machine learning, specifically the challenges of limited sample sizes and sparse annotations. In both instances, data augmentation and advanced training techniques are employed to maximize the utility of the available data. The first case study addresses a multi-label classification task for Long-COVID detection using a small tabular dataset. By applying approaches such as MLSMOTE to expand the dataset, the study successfully improves predictive performance, subsequently relying on post-hoc interpretability tools (SHAP) to verify that the augmented model still learns medically sound patterns. However, this reliance on retrospective validation raises a critical question: rather than verifying clinical validity post-hoc, can domain knowledge be structurally incorporated directly into the model training or augmentation phases? The second case study further explores this theme in the context of histopathological image classification under similar data scarcity. While current paradigms typically rely on massive foundation models fine-tuned for clinical tasks, such approaches often compromise data control, domain specificity, and interpretability. To address this, the second study demonstrates that training self-supervised models from scratch on small-scale datasets can achieve competitive performance with state-of-the-art models while retaining full control over the training pipeline. Yet, much like the first case, it highlights a fundamental gap: these purely data-driven, self-supervised pipelines still lack a dedicated mechanism to explicitly integrate clinical knowledge. Together, these studies illustrate the efficacy of current augmentation and self-supervision techniques, while exposing the methodological gaps that motivate the need for knowledge-grounded architectures.

2.3.1 Case Study 1: Over-sampling for Long-COVID Sequelae

To illustrate the practical challenges and solutions for applying machine learning to small medical datasets, we present a comprehensive case study on predicting long-COVID sequelae [12]. This study exemplifies how classical data-level and feature-level techniques can address the fundamental statistical problems that arise when working with limited clinical data, particularly in the context of multi-label classification tasks where the complexity is inherently amplified.

Dataset and Problem Characterization

The dataset is based on a long-COVID study conducted at Ospedale Maggiore of Novara, Italy, comprising data from 324 patients hospitalized during the first three waves of the COVID-19 pandemic. Following the clinical definition established in the literature, patients were considered to have long-COVID syndrome if they exhibited at least one symptom persisting beyond 12 weeks from acute infection onset, not attributable to alternative causes. The study focused on a one-year follow-up period to assess the presence of specific sequelae.

The dataset is characterized by: a sample size of 324 patients, with a total of 57 features, consisting of 47 binary and 10 numeric attributes. A label space consisting of 7 long-COVID sequelae, namely symptoms of COVID which are present at follow-up (e.g., arthromyalgia, asthenia, cough).

The features encompass two temporal categories: baseline data capturing demographic characteristics and pre-existing comorbidities (e.g., age, sex, obesity, hypertension, anxiety), and hospitalization data reporting symptoms at acute COVID-19 onset, administered treatments, and clinical outcomes (e.g., fever, cough, duration of hospitalization, oxygen administration, ICU intubation). The predictive task is formulated as follows: given baseline and hospitalization information, predict the presence of specific long-COVID sequelae at one-year follow-up.

Multi-Label Classification Framework

This problem maps to a multi-label classification (MLC) setting, where each patient can simultaneously exhibit multiple sequelae. Formally, let $X = x_1, \ldots, x_n$ be an instance space where each $x_i \in X$ represents a patient described by D features, and $L = \lambda_1, \ldots, \lambda_Q$ be a label space of Q possible labels (sequelae). The objective is to learn a function $h : X \to 2^L$ that predicts the subset of labels $Y_i \subseteq L$ associated with each patient. Multi-label classification introduces several challenges beyond standard single-label problems:

- **Label space sparsity**: The number of observed label combinations (label-sets) is typically much smaller than the theoretical maximum of $2^{|L|}$, creating a highly sparse label-set space. In the presented dataset, only 36 distinct label-sets were observed out of 128 theoretically possible combinations.
- **Label imbalance and inter-label dependencies**: Different sequelae occur with varying frequencies, and their co-occurrence patterns may encode important clinical information. The dataset exhibited a Mean Imbalance Ratio (MeanIR) of 2.303, indicating moderate imbalance across labels.

- **Increased complexity in evaluation**: Unlike single-label tasks, MLC requires evaluation metrics that account for partial correctness, as predictions may be partially accurate when only a subset of labels is correctly identified.

Understanding the dataset's label distribution characteristics is crucial for selecting appropriate methodologies. In particular, it is important to evaluate the SCUMBLE (Score of ConcUrrence among iMBalanced LabEls) metric, which quantifies the extent to which minority (rare) and majority (frequent) labels co-occur within the same data instances. Mathematically, for a single instance i with a set of active labels L_i, it measures the variance of their Imbalance Ratios Per Label ($IRLbl$) using the Atkinson index:

$$\text{SCUMBLE}_i = 1 - \frac{\left(\prod_{l \in L_i} \text{IRLbl}_l\right)^{1/|L_i|}}{\frac{1}{|L_i|}\sum_{l \in L_i} \text{IRLbl}_l}$$

where $IRLbl_l$ represents the ratio of the most frequent label's occurrence to the occurrence of label l in the dataset. The overall SCUMBLE is the average of $SCUMBLE_i$ across all instances. A high overall value indicates that instances with rare labels also frequently contain majority labels, making it difficult to oversample the minority without inadvertently over-representing the majority. In the present dataset, the low value of this metric (0.012) indicates minimal concurrence, confirming that data augmentation techniques targeting minority labels can be safely applied without disproportionately increasing the overall label imbalance.

Two Approaches to overcome data scarcity

Two complementary approaches were investigated to address the small dataset challenge. The first approach targets the augmentation of training samples, while the second aims to reduce feature dimensionality. While both strategies are well-established for data scarcity scenarios, their adaptation to multi-label classification (MLC) in this sensitive clinical domain presents unique challenges.

Data Augmentation via MLSmote. The first approach aims to increase the effective sample size (n) by generating synthetic instances, particularly targeting under-represented label combinations. We employed MLSmote (Multi-Label Synthetic Minority Over-sampling Technique) [25], which extends the classical SMOTE [26] algorithm to the multi-label setting. MLSmote operates through a three-stage process:

1. Minority label identification: Labels are ranked by their Imbalance Ratio per Label (IRLbl), identifying those requiring augmentation.

2. Synthetic instance generation: For each minority instance, k nearest neighbors are identified, and new synthetic instances are created through feature interpolation.

3. Label-set assignment: Labels for synthetic instances are determined using one of three strategies based on the reference instance and its neighbors.

We systematically evaluated six augmentation configurations, varying both the number of neighbours ($k = 3, 5$) and the label-set generation strategy: Intersection (I): Conservative approach where only labels appearing in both the reference sample and all neighbors are assigned to synthetic instances. Ranking (R): Moderate approach using majority voting, assigning labels present in more than half of the reference and neighbor samples. Union (U): Aggressive approach where labels appearing in either the reference sample or any neighbor are assigned.

Each configuration generated 100 synthetic instances (approximately 31% increase), producing six augmented datasets: k3I, k3U, k3R, k5I, k5U, and k5R. Table 1 summarizes the characteristics of these datasets compared to the original.

The augmented datasets exhibited notable differences in their structural properties. The aggressive union-based strategies (k3U, k5U) substantially increased label cardinality (1.75 labels per instance) and the number of distinct label-sets (48 vs. 36 in the original), while reducing overall label imbalance (MeanIR decreased to ~ 1.6). In contrast, the conservative intersection strategy (k3I, k5I) maintained label-set distributions closer to the original while still providing additional training examples.

Dimensionality Reduction. The second approach tackles the problem from the opposite direction by reducing feature dimensionality (D), thereby improving the effective sample-to-feature ratio. We evaluated both supervised and unsupervised feature selection methods to determine whether reducing complexity while preserving predictive information could improve model robustness.

Supervised approach: RELIEF [81] is a feature ranking algorithm that evaluates feature importance based on their ability to distinguish between instances with similar and different class labels. The algorithm computes a score for each feature by examining k nearest neighbors, incrementing the score when feature differences distinguish different classes (a

"miss") and decrementing it when differences occur within the same class (a "hit"). To apply RELIEF in the multi-label setting, we transformed the problem by treating each unique label-set as a distinct class, effectively converting it to a multi-class problem. Using $k = 10$ neighbours, we computed relevance scores for all 57 features. Following the statistical rule of thumb that correlation is significant when exceeding $\sigma = 2/\sqrt{n}$ (where $n = 324$, yielding $\sigma \approx 0.11$), we selected the top 8 features that surpassed this threshold. This aggressive reduction retained only 14% of the original features, creating the Relief dataset.

Unsupervised approach: Principal Component Analysis (PCA) provides an unsupervised alternative by identifying linear combinations of features that capture maximum data variance. After excluding the target sequelae labels, we performed PCA on the 57 input features and retained components explaining 95% of the cumulative variance. This yielded 44 principal components, representing a more moderate 23% reduction in dimensionality, creating the PCA dataset.

The contrasting reduction levels between these approaches—86% reduction for RELIEF versus 23% for PCA—reflect their fundamentally different objectives: RELIEF aggressively selects features correlated with specific outcomes, while PCA preserves overall data structure. This difference proved consequential for model performance, as we will discuss.

Experimental Results and Model Performance

We evaluated a representative set of problem transformation-based MLC methods spanning three categories: binary methods (Binary Relevance, Classifier Chain, Bayesian Classifier Chain with two ordering strategies), multi-class methods (Label Combination, Pruned Set), and ensemble methods (Conditional Dependency Network, RAkEL). All methods employed Random Forest as the base classifier, evaluated on a 5-run of 10-fold cross-validation. The results are presented in Appendix A.1.

Across all metrics, several consistent patterns emerged:

1. Multi-class methods (PS, LC) and classifier chains effectively captured label dependencies and generally provided the best bipartition-based performance;
2. Data augmentation, particularly with intersection and ranking strategies, improved model robustness without introducing artifacts;
3. Aggressive feature reduction through RELIEF consistently degraded performance, suggesting loss of clinically relevant information;

4. PCA offered computational benefits without sacrificing accuracy, though it provided no performance gains over using the full feature set;

5. Union-based augmentation showed mixed results. In general, it seems beneficial for ranking metrics but potentially detrimental for exact prediction tasks.

Remarks on model calibration. The use of data augmentation techniques such as MLSMOTE raises an important consideration regarding model calibration. In clinical contexts, reliable probabilistic outputs are essential for accurate risk assessment. However, by oversampling minority classes, MLSMOTE alters the empirical prior distribution of the training data, which can artificially inflate the predicted probabilities of rare labels. Therefore, before deploying these models in real-world clinical scenarios, the application of post-hoc calibration techniques requires further investigation.

Despite this limitation, an analysis of our dataset metrics suggests that the risk of major probability distortion was inherently limited in this specific setting. In multi-label classification, calibration is most severely degraded when synthetic instances are generated in high-entropy regions where minority and majority labels heavily overlap. To monitor this, we can look at the MeanIR and the SCUMBLE metric. The original dataset exhibited a high MeanIR (2.303) but a notably low SCUMBLE (0.012). Crucially, after applying MLSMOTE, both metrics decreased (see Appendix A.1). This reduction indicates "clean oversampling"; the algorithm successfully populated minority classes without increasing label entanglement or injecting synthetic noise into majority-class feature spaces.

Consequently, while formal calibration remains a necessary step for future clinical deployment, the stability of the SCUMBLE metric provides a strong theoretical indication that the augmentation process did not fundamentally compromise the underlying separability and reliability of the model's probabilistic outputs.

Interpretability Analysis: Validating Model Decisions

While achieving strong predictive performance is essential, a critical aspect of clinical deployment of machine learning models is related to interpretability. Physicians must understand why a model makes specific predictions to trust and appropriately act on its outputs. We conducted a comprehensive interpretability analysis using SHAP (Shapley Additive Explanations) [96], an established method that quantifies each feature's contribution to individual predictions by considering all possible feature combinations.

In particular, we examined whether the model's decision-making aligned with established clinical knowledge and, more importantly for this discussion, what impact the data

augmentation techniques had on the SHAP analysis. Indeed, does synthetic data generation through MLSmote alter the model's decision-making process? If augmentation fundamentally changed which features the model relied upon, it could introduce artifacts or biases absent in real clinical data.

Clinical Insight Extraction. In general, the aggregation of SHAP values across all patients revealed that the most influential features for predicting long-COVID sequelae were: sex, BMI, hospitalization duration, and psychological factors. All of which were, with different levels of certainty, associated with Long-COVID in other works in literature [100, 125, 141]. A more detailed analysis on the features importance is presented in Appendix A.1.

Impact of Data Augmentation on Interpretability. To investigate this, we normalized SHAP values for each dataset independently (0-1 range) and compared the relative importance of the top 15 features from the original dataset model across all augmented dataset models. This showed consistent results: features identified as most important in the original dataset maintained high SHAP values across augmented datasets, particularly for intersection and ranking strategies (k3I, k5I, k5R). Union-based augmentation showed slightly reduced importance for the top three features, likely reflecting the more substantial structural changes these aggressive strategies introduced (see Appendix A.1).

This analysis confirms that appropriate data augmentation, particularly with conservative strategies, preserves the underlying clinical relationships the model learns, providing additional training examples without distorting the fundamental feature-outcome associations.

From Post-Hoc Explanation to Knowledge Integration

The interpretability analysis revealed a critical aspect in current data-driven approaches to clinical AI: while SHAP analysis successfully identified that the model used clinically sensible features (i.e., sex, BMI, hospitalization duration, psychological factors), this validation occurred after model training through post-hoc analysis requiring expert consultation. This reactive approach has several limitations:

1. Resource intensity: Interpreting model decisions requires time-consuming collaboration with clinical experts for each new model or application.

2. Indirect knowledge incorporation: Clinical knowledge influences the model only through its implicit encoding in training data, not through direct integration of established

medical understanding.

3. Limited guarantees: Even when post-hoc analysis reveals clinically sensible patterns, there is no guarantee the model will maintain these relationships when deployed on new populations or edge cases.

4. Knowledge extraction burden: Validating that features like "comorbidity count" and "severity indicators" matter for long-COVID required extracting and formalizing clinical knowledge specifically for this interpretability study.

This case study thus motivates a fundamental question:

If we ultimately need to extract and validate structured clinical knowledge to trust model predictions, why not integrate this knowledge directly into the model training process?

Several arguments support a more active approach to knowledge integration:

- **For consolidated medical knowledge**: When clinical relationships are well established (e.g., the association between obesity and severe COVID-19 outcomes, or the link between initial disease severity and persistent sequelae), directly encoding these relationships could improve model robustness, particularly in low-data regimes. Rather than relying solely on the 324 patients to rediscover these associations, leveraging consolidated results from clinical research could enhance generalization.

- **For interpretability by design**: Knowledge-grounded models could provide inherent interpretability, explaining predictions in terms of known clinical pathways rather than requiring post-hoc analysis. A model that explicitly reasons about "disease severity $\rightarrow$ inflammatory response $\rightarrow$ persistent symptoms" is more transparent than one that implicitly learns these associations.

- **For handling knowledge gaps**: Importantly, knowledge injection should be flexible: if incorporated knowledge proves irrelevant or contradictory for a specific prediction task, the model should be able to down-weight or ignore it. Thus, architectural designs should treat knowledge as soft guidance rather than hard constraints. Indeed, this aspect becomes crucial when distinguishing between consolidated knowledge (e.g., known risk factors) that should inform model decisions and exploratory hypotheses (e.g., novel treatment effects) where blind knowledge injection could introduce unwanted biases.

Critical caveat. Leveraging and incorporating existing knowledge must be balanced against the risk of introducing biases or limiting the model's ability to discover novel patterns. For instance, when investigating the effectiveness of a new treatment for long-COVID, injecting prior beliefs about treatment efficacy could prejudice the analysis. Thus, knowledge integration must be selective, focusing on stable background knowledge rather than the specific phenomena under investigation.

2.3.2 Masked Autoencoders for Small Histopathology Datasets

This case study examines our work "Beyond Labels: A Self-Supervised Framework with Masked Autoencoders and Random Cropping for Breast Cancer Sub-type Classification," [40], which demonstrates how self-supervised learning (SSL) can address annotation and data-related issues in histopathological image analysis. While the previous case study focused more on cohort scarcity in tabular clinical data, this work tackles label scarcity situations where images are available but expert annotations are prohibitively expensive or impossible to obtain at scale.

The Annotation Bottleneck

Medical imaging, particularly histopathology, faces a critical challenge: obtaining expert annotations requires specialized training and significant time. A pathologist must examine tissue samples at cellular resolution to identify cancerous regions and classify tumor subtypes. This creates severe bottlenecks in several scenarios:

- Rare diseases: Few cases exist even in large medical centers, making it impossible to collect thousands of labeled examples.
- Small-scale studies: Research projects with limited patient cohorts cannot afford comprehensive annotation of all available imaging data.
- Resource-limited settings: Smaller institutions lack access to extensive pathology expertise for large-scale annotation efforts.
- Privacy constraints: Data sharing regulations prevent pooling data across institutions to create larger labeled datasets.

These considerations raise questions about the feasibility of building effective classification models with limited labeled data.

Related Approaches

Recent advances in computational pathology have turned to large-scale self-supervised learning as a solution. Foundation models like UNI and HIPT [28, 29] demonstrate the potential of this approach by training on millions of images and achieving state-of-the-art performance across more than 34 pathology tasks. These models also exhibit remarkable generalization, allowing application to new tasks with minimal fine-tuning. However, their use in small-scale studies raises important considerations:

- Limited data control: Researchers have no control over what patterns dominate the model's representations. In specialized studies on specific cancer subtypes, the model may not effectively capture the subtle distinctions most relevant to the task.
- Interpretability challenges: Understanding why a model trained on hundreds of thousands of diverse slides makes specific predictions is difficult, raising issues for clinical adoption and regulatory approval.
- Resource barriers: Training foundation models requires substantial computational infrastructure. Researchers must rely on external organizations to train and release these models, creating dependency and limiting customization.
- Domain specificity: In highly specialized contexts, such as rare tissue types, novel staining protocols, or specific morphological features, representations learned from broad datasets may be insufficient.

These considerations motivated our investigation: **Can self-supervised learning on small, domain-specific datasets achieve competitive performance while maintaining full control over the training pipeline?**

Methods

The Histopathological Masked Autoencoder (HMAE) framework (see Figure 2.2) addresses label scarcity through three key innovations.

1. Masked Autoencoder Architecture. MAE [57] separates representation learning from classification through a two-phase approach: A **self-supervised pre-training phase** in which input images are divided into patches, 75% randomly discarded; the remaining 25% are processed through a Vision Transformer (ViT) encoder [39]. Subsequently, a decoder reconstructs the complete image from the sparse encoded patches and pad tokens. Finally, the reconstructed image is optimized by using a reconstruction error loss. A **supervised classification phase** - after the pretraining phase is completed, the encoder module is frozen and used to extract features from the complete images. A linear probe is then trained on these features using the limited available labels.

The aggressive 75% masking serves dual purposes: it forces the model to learn robust representations capable of reconstructing complex tissue structures from partial information, and it significantly reduces computational cost by processing only 25% of patches, making training approximately more efficient than full-image processing.

2. Strategic Data Augmentation. A fundamental aspect that enables effective SSL on small datasets is the sampling module. Scale-aware extraction is employed by sampling crop dimensions from a distribution based on annotated regions of interest, ensuring the model encounters realistic scale variations of diagnostically relevant features. Selection is then applied by using pixel variance to filter samples, via thresholding, retaining those with sufficient content, and discarding those containing primarily background. This approach ensures that each whole-slide image generates numerous training patches at varying scales and locations, allowing a dataset of hundreds of slides to yield tens of thousands of training examples and provide sufficient data for effective SSL.

3. Computational Efficiency. HMAE's design is practical for resource-constrained settings. By processing only 25% of patches, encoder computation is reduced by approximately 75%, substantially lowering memory and training requirements. Furthermore, the ViT-Small/16 architecture comprises 22 million parameters, orders of magnitude smaller than contemporary foundation models. Complete training runs on a single NVIDIA A40 GPU in 32 hours, demonstrating that effective self-supervised learning does not require large-scale computational infrastructure.

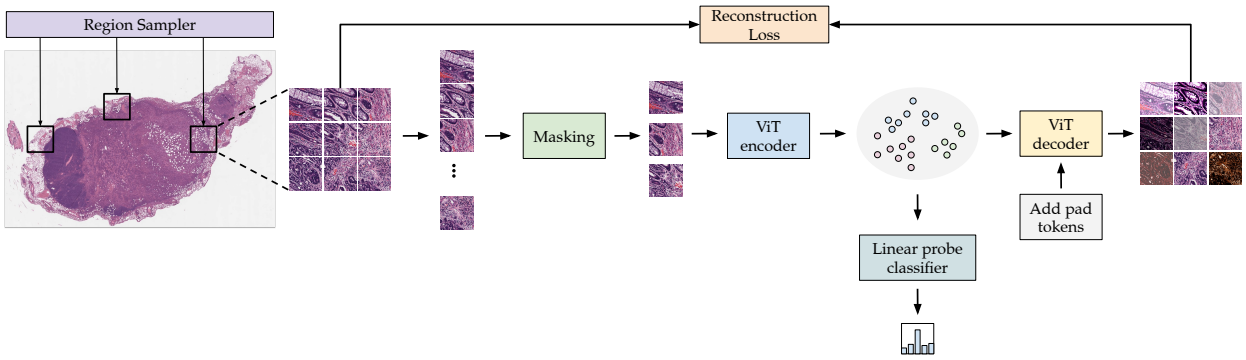

Figure 2.2: The HMAE architecture.

Experimental evaluation

HMAE was evaluated on breast cancer classification using the BRACS dataset (547 whole-slide images with 4,539 annotated regions) and BACH dataset (400 annotated regions) [5, 17]. The experimental evaluation demonstrated that HMAE achieves performance comparable to state-of-the-art methods across multiple tasks. On 3-class cancer classification, statistical analysis revealed no significant difference from top performers (ACMIL) in both AUC and weighted F1-score. For the more challenging 7-class sub-type classification, HMAE achieved a

weighted F1-score of 0.578, with particularly strong performance on normal tissue. Most importantly, the learned representations generalized effectively: when tested on the completely unseen BACH dataset, HMAE achieved performance statistically comparable to specialized methods (weighted F1=0.687 vs. 0.734, p=0.052). For more details, see Appendix A.2.

Discussion

The presented results show that effective SSL does not require large-scale data when appropriately designed for the target domain. HMAE achieves statistically indistinguishable performance from state-of-the-art methods on several tasks despite training on a single small dataset rather than millions of diverse images, using a simple architecture rather than specialized designs, and employing straightforward mean pooling rather than complex attention mechanisms.

Advantages of Full Pipeline Control. SSL can offer substantial benefits in specialized research contexts. Researchers maintain data transparency, understanding precisely which data informed the model's representations and ensuring alignment with study objectives. Sampling strategies can be customized to emphasize scales or tissue regions relevant to specific research questions. The accessible computational requirements enable iterative refinement through experimentation with architectural variations and training strategies. With limited, well-defined training data, understanding the patterns driving model behavior is more feasible than with foundation models trained on diverse, aggregated datasets. This approach is particularly valuable for studies involving rare diseases, specialized tissue types, or resource-constrained settings where foundation model deployment may be impractical.

Limitations

Despite its success, HMAE reveals fundamental limitations of purely data-driven approaches:

- Fine-grained classification challenges: performance on subtle distinctions between intermediate cancer categories suggests reconstruction objectives alone may be insufficient. Pathologists use explicit diagnostic criteria, specific cellular architectures, nuclear features, and tissue organization patterns to distinguish these categories. Current SSL has no mechanism to emphasize diagnostically relevant patterns.

- Dataset balance: random sampling treats all tissue regions equally, but some types are rarer or more diagnostically challenging. More sophisticated sampling guided by structured knowledge about tissue hierarchies could improve representation of difficult cases.

- Lack of explicit medical grounding: the model learns patterns that happen to correlate with diagnostic categories but has no explicit connection to established medical knowledge like ontologies, clinical guidelines, or diagnostic criteria. This limits explainability and provide no guarantees that learned patterns align with medically sound reasoning.

This case study demonstrates that small-scale self-supervised learning provides a viable alternative to foundation models for resource-constrained settings and specialized research contexts. HMAE achieves competitive performance while offering advantages in data control, interpretability, and accessibility. However, it also reveals that purely data-driven approaches lack mechanisms for incorporating structured medical knowledge, limiting their ability to reason about fine-grained distinctions and explain predictions in terms of established medical concepts.

Chapter 3

Knowledge Injection for Process Mining

This chapter presents the first of three core contributions: a methodology for systematically injecting structured knowledge into deep learning architectures. As established in Section 2.1.3, creating clinically grounded machine learning systems requires models that learn not only from empirical observations but also from the formalized knowledge that encodes how those observations are generated. This dual-source learning, balancing data-driven pattern recognition with adherence to established medical reasoning, forms the conceptual foundation of knowledge injection.

We demonstrate this principle through clinical process monitoring, where real-world pathways exhibit substantial variation yet remain fundamentally shaped by evidence-based guidelines. A purely data-driven model may learn spurious patterns that deviate from principled medical reasoning. In contrast, rigid rule-based systems cannot adapt to the inherent complexity of actual clinical practice. Our approach bridges this gap by enabling the model to simultaneously learn from real-world sequential data and the structured ontological knowledge encoded in clinical guidelines.

The Approach. This chapter, derived from "*Structural positional encoding for knowledge integration in transformer-based medical process monitoring and trace classification*" [67], introduces a method for injecting structured clinical knowledge into sequential process data through transformers enhanced with graph-based positional encoding. Rather than relying on post-hoc filtering or rule-based constraints, we embed domain guidelines directly into the model's learning mechanism. The key innovation lies in replacing standard positional encoding with Structural Positional Encoding (SPE), derived from a clinical ontology graph. This ensures that activities with similar clinical purposes (e.g., diagnostic procedures under patient assessment) receive similar positional representations, which allows the transformer to learn semantically meaningful patterns aligned with clinical reasoning. The architecture naturally extends to trace classification by framing it as prediction of a final process outcome token, demonstrating the generality of knowledge injection across distinct process mining tasks.

Experiments on Stroke Management. We validate our approach in stroke management, a domain where predictive process monitoring addresses critical clinical needs. Stroke treatment demands balancing multiple competing objectives, and real patients often present complexities absent from idealized guideline pathways. Our experiments systematically evaluate three key aspects:

- **Performance improvement**: SPE-based consistently outperforms baseline models

in both next activity prediction and trace classification;

- **Robustness to noisy data**: the method demonstrates greater resilience to incomplete traces compared to standard approaches.
- **Clinical alignment**: validated through qualitative expert assessment showing that SPE-based predictions better adhere to established stroke management guidelines.

These results confirm that knowledge injection produces not only statistically superior models but also clinically sound decision support.

Beyond these empirical findings, the modular design of our architecture reveals an important property: the SPE module can be replaced or tested with different ontologies while maintaining the core framework. This flexibility allows practitioners to evaluate how different knowledge structures affect model behavior on the same data, demonstrating that knowledge injection via positional encoding is a principled, generalizable technique rather than a domain-specific solution. This property becomes crucial for the thesis's broader narrative, establishing that structured knowledge can be systematically integrated into deep learning architectures across diverse medical domains and guideline structures.

3.1 Introduction to clinical process mining

The diffusion of advanced medical information systems is progressively enabling the automatic collection of patients' *traces*, i.e., the sequences of activities executed on patients during the care procedures implemented at a hospital organization [115]. Patients' traces are a valuable source of information for several analyses and investigations, within the field of process mining [138]. Among the various available process mining techniques, predictive process monitoring [98, 136] is of particular interest. Predictive process monitoring aims at forecasting relevant information about a running process trace; specifically, it exploits the already logged traces to make predictions about the running trace completion, such as suggesting the next activity to be executed, or estimating the remaining time/cost/resources required to complete itself. Interestingly, such a forecast can also predict an issue before it occurs, so that preventive countermeasures can be taken. Different prediction types can be supported [37], namely:

1. **Outcome-based predictions**: predictions related to predefined categorical or boolean outcome values.

2. **Numeric value predictions**: predictions related to measures of interest taking numeric or continuous values, such as remaining times or costs.

3. **Next activity predictions**: predictions related to sequences of future activities.

In the medical field, predictive process monitoring can support better time and resource allocation; moreover, and most importantly, through next activity predictions, it can support decision-making in non-trivial cases (Figure 3.1). Indeed, despite the availability of clinical guidelines, it is worth noting that guidelines are ideal processes, designed for ideal patients, and meant to be applied in an ideal setting, where all the needed resources are always available [150]. In reality, this is often not the case: local resource constraints may prevent the (timely) execution of the scheduled guidelines activities. Moreover, real patients may be atypical, for instance, due to the presence of comorbidities or rare disease variants. Finally, physicians may lack the necessary background knowledge to correctly interpret and apply guidelines in complex cases. Predictive process monitoring, by suggesting the next activity to be executed, can thus provide helpful advice in these non-trivial situations.

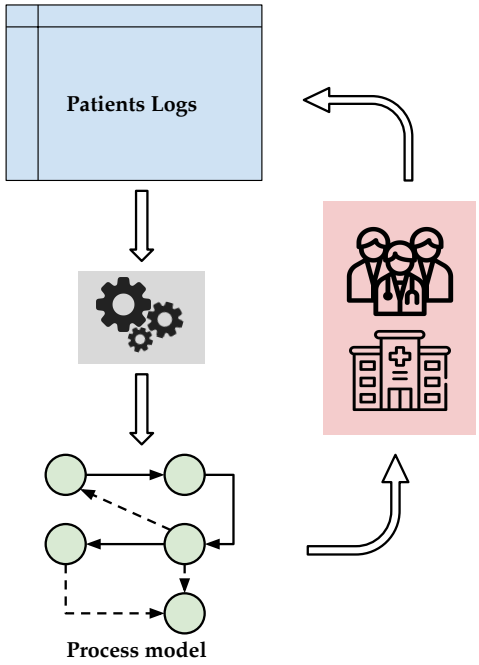

Figure 3.1: Process mining cycle in healthcare. Clinical procedures performed on patients are recorded in event logs. These logs are analyzed to discover and construct a process model. The resulting model is then applied in clinical practice to support decision-making and to evaluate and improve the underlying care processes.

Another very useful process mining technique is trace classification [18], able to categorize traces with respect to specific properties. Interestingly, classification can be seen as a predictive process monitoring task as well, specifically as an example of outcome-based prediction [37], where the categorical outcome is the label of the class. Classification can be used to check if a trace meets given criteria, thus identifying a possible non-compliance with the expected performance, in a process and resource optimization perspective, or, more generally, in a quality assessment perspective. In medical organizations, classification of patient traces allows medical experts and administrators to better understand the actual implementation process, identify bottlenecks or issues, and improve the overall quality of care.

Medicine is peculiar with respect to other settings, since medical processes are typically based on the application of a relevant amount of well-grounded domain knowledge. Process mining is, by definition, based on data (essentially, on traces); nevertheless, recent approaches (see, e.g., [85]) are proposing to integrate medical domain knowledge within various process mining tasks.

Stemming from the considerations discussed above, this work presents a predictive process monitoring approach for next activity prediction in medical process traces; according to the most recent literature advances, the proposed method relies on the exploitation of a *transformer*, a deep learning architecture based on the self-attention mechanism [139], which has already demonstrated its utility in modeling similar tasks, as exemplified in [19]. The core contribution of the work lies in the introduction of a technique that incorporates domain knowledge within the model. This augmentation is carried out through a graph positional encoding method, enhancing the accuracy of our model.

Furthermore, the same transformer architecture is exploited for trace classification. Specifically, it is possible to add the class label as the last trace activity, and then carry out the classification task as a prediction one, resorting to the same architecture. The resulting knowledge-enhanced deep-learning architecture is therefore able to support two particularly significant process mining tasks within a unified framework.

While general, this approach is motivated by a specific application domain, which was selected for the experiments, namely, the one of stroke management. Indeed, in this field, predictive process monitoring can provide a valuable contribution to medical decision-making. To mention a few examples, it can help in the choice of the correct thrombolytic drug, by considering how indications and contraindications of the available medications were balanced in past traces [150], or it can help in deciding whether to confirm the presence of subarachnoid hemorrhage by taking into account pros and cons of a lumbar puncture [16], always referring to already completed traces.

Stroke trace classification, on the other hand, can help quality assessors and medical experts to verify whether different patients are treated according to their characteristics. Specifically, stroke patients can be distinguished between simple and complex ones: complex patients are those affected by rare stroke types or experiencing comorbidities. Different patient types are supposed to follow different guideline recommendations (e.g., complex patients require additional treatments, and/or more frequent diagnostic steps to strictly monitor their evolution over time), thus originating different traces. If the traces belonging to simple versus complex patients are classified into different classes, we can conclude that the hospital is identifying and implementing the correct process type.

3.2 Related work

Predictive process monitoring and classification have been addressed by different types of machine learning techniques, such as Transition Systems [84], Hidden Markov Models [82], and Support Vector Machines [20]. In more recent approaches, however, deep learning architectures are being applied and nowadays represent the state of the art in the field.

Different deep learning architectures have been proposed to deal with the task of next activity prediction. The approach in [102], for instance, uses Autoencoders [62]. Autoencoders can successfully reduce the number of attributes of the input activities; however, they are unable to treat long (temporal) dependencies between activities of a trace. Convolutional Neural Networks (CNNs) [4] have been applied in [101], where sequential data in process traces are treated as a one-dimensional grid.

Due to the sequential nature of traces, however, Recurrent Neural Networks (RNNs) [110] can represent a more natural solution. In fact, RNNs can capture longer dependencies between the activities of a trace, while in a CNN an activity only depends on the k most recent activities, where k is the size of the kernel. Within RNNs, Long-Short Term Memory (LSTM) networks [63] represent a particularly performing approach. Indeed, LSTM can potentially learn the complex dynamics within the temporal ordering of input sequences; therefore, they are well-suited to manage the sequential data of process activity logs. They can also manage long-distance dependencies between activities, since they implement a long-term memory where the information flows from cell to cell with minimal variations, keeping certain aspects constant during the processing of all inputs. The works in Camargo et al. [21], Evermann et al. [44], Tax et al. [130, 131], for instance, rely on LSTMs to predict the next activity of a running case.

Similarly to LSTMs, Gated Recurrent Networks (GRUs) [32] also create paths through time that allow the gradients to flow deeper in the sequence than in basic RNNs; with respect to LSTMs, they have fewer parameters. GRUs are exploited in, e.g., [61]: the approach described in Hinkka et al. [61] is designed to solve any classification problem (i.e., it is not focused just on next activity prediction, as it happens for most of the other already mentioned works).

In Khan et al. [78], instead, a Memory Augmented Neural Network (MANN) is proposed. MANNs allow to learn even longer dependencies; training is more expensive, but MANNs are suitable for very long traces, or when cycles of the same activity may lead LSTMs or GRUs to forget/ignore activities located at the beginning of the trace.

The approaches which most closely relate to one presented here are those in Bukhsh et al. [19], Philipp et al. [112], which rely on a transformer, an architecture that substitutes the recurrence by the attention mechanism [139]. Transformers are becoming the state of the art not only in many Natural Language Processing and time series prediction tasks, but also in the next activity prediction task in predictive process monitoring (see, e.g., [38]). In particular, these works adopt multi-head attention, i.e., they perform the attention operation over different parts of the sequence simultaneously, and, instead of training an encoder-decoder architecture as in [139], they only rely on the decoding part.

Exploiting transformers not only for token prediction, but also for classification (where the class is interpreted as a special token to be predicted), has been adopted in other well-known literature approaches, such as, e.g., in BERT [35].

A few approaches have addressed the idea of incorporating domain knowledge for improving next activity prediction. The work in Di Francescomarino et al. [36] adopts an LSTM architecture to predict all possible suffixes of a process trace (given the prefix), ranks the suffixes based on their likelihood, and then keeps the first suffix that is compliant with domain/background knowledge (expressed by means of Linear Temporal Logic constraints). The work of Donadello et al. [38] is inspired by such a contribution, but moves towards the adoption of a transformer. Moreover, conformance with domain knowledge is tested while the suffix is still being constructed, and not just at the end. Domain knowledge is represented as a Petri Net, and compliance is obtained using token-based replay [13]. Domain knowledge is, however, not integrated within the deep learning architecture, as it happens in our case.

Incorporation of domain knowledge in the form of graphs has been explored in Di Francescomarino et al. [36], Dwivedi et al. [42], Mialon et al. [104], aiming to enrich sequences with geometric information derived from a graph structure. An example of this is the combination of protein structures with the geometric arrangement of molecules. The knowledge-driven approach has also been used in several works involving pre-trained transformers (e.g., BERT) in order to align the model to a specific knowledge domain by means of ontologies and knowledge graphs [92, 121].

As for the specific application domain of stroke management, several machine learning approaches have been adopted (see, e.g., the survey in [34]), demonstrating high accuracy in imaging analysis, stroke sub-types diagnosis, risk stratification, and patient prognosis prediction. To provide some examples, the work in Chilamkurthy et al. [31] has proposed deep learning to diagnose intracranial abnormalities on computer tomography scans; the work in Scrutinio et al. [120], based on the use of a random forest algorithm, has outperformed logis-

tic regression in predicting three-year mortality; linear Support Vector Machine regression has been adopted to predict the outcome of the rehabilitation process in the early stages of stroke [119].

It is worth noting that missing and irregular data in clinical event sequences have been investigated in the literature, with works such as Luo [97] framing irregular time series explicitly as a missing data problem. The present work partially addresses this aspect through the robustness experiments in Section 3.4, demonstrating stable performance under controlled trace incompleteness. Nevertheless, explicit missing data modeling remains a complementary direction worth exploring, particularly in small data regimes where incomplete observations further reduce the effective sample size.

3.3 Methodologies

The following sections will present: (i) the base architecture of the transformer model. (ii) method that enables leveraging domain knowledge using Structural Positional Encoding (SPE), which is based on the Laplacian eigenvalue encoding technique [42]. (iii) How to model the trace classification task directly in the transformer-based architecture.

3.3.1 Transformer-based architecture for process monitoring

The model architecture is based on the transformer decoder (as in [19, 112]). Inputs are defined as a sequence $S = \{a_1, ..., a_i, ..., a_n\}$ (representing a trace) where every element of the sequence represents an activity coming from a set A of possible activities. Figure 3.2 (right) depicts the model structure. The model's architecture and input processing are detailed below.

Embedding layer: it takes S as input and returns a vector of dimension $\mathbb{R}^d$ for each activity a_i. At this point, our samples are in the form $X \in \mathbb{R}^{n \times d}$.

Positional encoding: this layer is responsible for augmenting each embedding representing an activity with information about its position. In particular, two versions of positional encoding are tested. The first version (PE henceforth), described in [139], uses a sine and a cosine function to obtain a representation that respects the order of the activities within

the single sequence. In particular, the positional embeddings are calculated as:

$$\text{PE}(pos, 2i) = sin(pos/10000^{2i/d})$$
$$\text{PE}(pos, 2i+1) = cos(pos/10000^{2i/d})$$

where *pos* represents the position of the embedding in the sequence, i is the i-th component of the embedding, and d is the dimensionality of the embedding.
The second version, which is reported also in Figure 3.2, is Structural Positional Encoding (SPE henceforth, see section 3.3.2), which relies on the Laplacian eigenvector technique [11] to embed knowledge from a graph G (in our case, the graph is an ontology of the activities, described in section 3.3.2).

Positional encoding generates a vector for each activity in a sequence that is added to the original embedding as follows:

$$X = \begin{cases} X + \text{PE}(X) & \text{for PE method} \\ X + \text{SPE}(X, G) & \text{for SPE method} \end{cases}$$

Multi-head attention layer: the use of the self-attention layer is related to the fact that it helps contextualize a running trace, giving the possibility for the model to selectively attend to the previous activities. This component thus also enables the model to find long-range dependencies. To this end, the self-attention layer [139] calculates three representations Q, K, V for every activity in the sequence S. This is done by applying a fully connected layer to the embeddings representing the sequence and subsequently applying the scaled dot-product attention:

$$Q = \Theta^q X$$
$$K = \Theta^k X$$
$$V = \Theta^v X$$
$$H = \text{softmax}\left(\frac{QK^T}{\sqrt{d}}\right) V$$

where $\Theta^q, \Theta^k, \Theta^v$ are learnable matrices and d is a scaling factor set to the number of components of the embeddings.

This operation is carried out in parallel on multiple attention heads in order to learn different types of dependencies. The resulting representations are concatenated to produce the final output.

Note that, since the task is to predict the next token in the sequence, during the self-attention process, we apply a mask so that an activity can only attend to previous ones, and ignore those that follow it. The output is finally added to the input X using a skip connection.

Layer norm: this normalization layer is used to mitigate vanishing or exploding gradients phenomena during the model training phase. The process ensures that the activations are centered around zero mean and unit variance, making the optimization more stable [9]. The current implementation resorts to the *Pre-LN* configuration, where the layer-norm is put inside the residual blocks, which improves the model convergence [149].

Lastly, X is subsequently passed through 2 fully connected layers, where the final layer acts as a decoder, projecting the representations into a dimensionality equivalent to the number of potential activities.

Training: the transformer model is trained in an auto-regressive language modeling manner to predict the next activity given a context (i.e., the previous activities in the sequence). The training process involves the model attempting to predict the next activity given a sequence prefix. In particular, the model is trained on prefixes of varying lengths, ranging from 1 to n (where n is the sequence length). This is indeed a strategy already proposed in the literature for language-modelling architectures [139].

The loss function applied is the cross-entropy loss between the model's predicted probability distribution over all possible activities and the true tokens in the training data.

3.3.2 Domain knowledge injection

This section will first describe the knowledge model (an ontology), and the methodology used to integrate domain knowledge within the transformer-based architecture.

Stroke ontology

This application relies on the availability of an ontology (see Figure 3.3 for an excerpt), defined by medical experts, aiming at grouping together the activities involved in similar diagnostic or treatment goals. Grouping-by-goal is achieved through a taxonomic representation, where activities involved in similar goals are placed in a close relationship (having a shorter path to reach one from the other). According to this viewpoint, two main goals (sons of the root) can be identified: the emergency phase patient care goal and the hospitalization phase patient care goal. Within both phases, three sub-goals are possible: patient management,

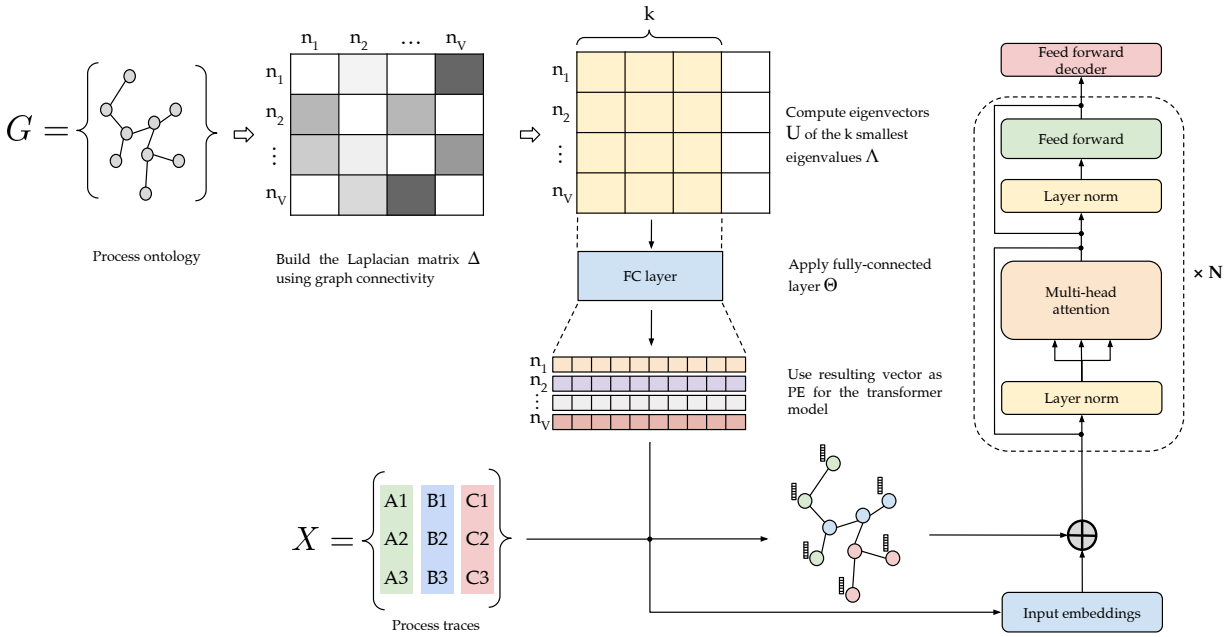

Figure 3.2: The model architecture. The left side shows the Structural Positional Encoding module and how the Laplacian Positional Encoding Embedding is calculated. The right side of the Figure represents one layer of the transformer-decoder module (which can be repeated N times). Based on our experimental dataset, the dimensionality is defined as follows: the process ontology graph consists of $V = 82$ nodes (corresponding to the total unique activities). The input matrix X represents the process traces, which are drawn from a dataset containing 5342 total traces, with an average sequence length of 15 activities per trace.

patient assessment, and therapy. Patient management includes as leaves some administrative ground activities, such as arrival or discharge. Patient assessment includes diagnostic activities, such as Computer Tomography (CT). Therapy includes therapeutic activities, such as pharmacological ones or lifestyle improvement ones. For instance, `ECG_H` (electrocardiogram), `RX_H` (X-rays), and `Angiography_H` (angiography) are three leaves, which represent diagnostic activities, sharing the goal of assessing the patient's condition at the time of hospitalization.

Overall, the taxonomy is composed of 136 classes, organized in a hierarchy of five levels. For instance, `CT_E`, which is a son of therapy in the emergency phase, is, in turn, subdivided into a whole family of specific exams in Figure 3.3.

Knowledge integration method

The integration of domain knowledge within the transformer-based architecture has been accomplished by resorting to Structural Positional Encoding (SPE).

The ontology, encoded as a graph, relies on the Laplacian eigenvector technique [11], which is employed to calculate node embeddings, yielding a vector for each activity (i.e., node) that encodes its position in the graph. This ensures that nodes close to each other possess similar embeddings.

Intuitively, nodes with similar values in a particular eigenvector (e.g., the Fiedler vector) tend to belong to the same cluster or community within the graph. This suggests that they are "close" in terms of graph structure, which could mean either a small graph distance (few edges apart) or being part of the same densely connected subgraph.

During the training phase, these node representations are added to the activity embeddings, before the multi-head attention layer (in a positional encoding manner). This augmentation enriches the representations of the transformer with the relational information inherent in the ontology. This enables the model to learn additional types of interactions among activities.

Laplacian eigenvectors are particularly advantageous in this context due to their ability to preserve the global structure of the graph while simultaneously capturing local neighborhood information, resulting in a smooth embedding space. Moreover, this method does not require a learning process, making it computationally efficient.

The operations in this phase are formalized below, with Figure 3.2 (left) providing a visual overview.

The graph $G = (V, E)$ representing the ontology is structured as follows: V represents the set of nodes, one for every activity in the set A (activity nodes), along with some nodes that represent the "type of activity" (activity-type nodes). E captures the set of edges, connecting activity nodes to activity-type nodes. Activity-type nodes are linked together so that the graph is overall connected. The Laplacian eigenvectors are defined by the factorization of the graph Laplacian matrix:

$$\Delta = I - D^{-\frac{1}{2}} A D^{-\frac{1}{2}} = U^T \Lambda U$$

where I is the identity matrix, $A \in \mathbb{R}^{n \times n}$ assuming $n = |V|$ is the adjacency matrix where $A_{ij} = 1$ if node i and node j are connected by an edge (i.e. the graph connectivity) and $D \in R^{n \times n}$ where $D_{ii} = \sum_{i=1}^{n} A_{ij}$ is a diagonal matrix representing the degree of every node in G; Λ, U denote the eigenvalues and eigenvectors respectively. The embedding for a node i is a vector $u_i \in \mathbb{R}^k$ defined as the k smallest nontrivial eigenvector components of that node [41]. Finally, to incorporate the node embedding $u_i \in \mathbb{R}^k$ into the corresponding activity

embedding $X_i \in \mathbb{R}^d$, a fully-connected layer is employed:

$$X_i = X_i + \Theta u_i$$

where $\Theta \in \mathbb{R}^{k \times d}$ learnable weight matrix that ensures compatibility between the embedding sizes.

3.3.3 Trace classification

Integrating the task of assigning a class to a trace within the transformer architecture is straightforward. The model, trained on a language modelling task, predicts the next activity given all possible prefixes ranging from 1 to n (as detailed in section 3.3.1). To predict the class of a trace, predicts the final token (representing the class) given the entire sequence as a prefix. To implement this, an augmented dataset is constructed by appending an additional token to each trace that represents the corresponding class. This approach allows the model to learn both the sequence of activities and the final class assignment in a unified way.

To steer the model toward the trace classification task, the loss function is tweaked to assign higher weights (calculated empirically) to the class tokens while leaving the weights associated to the activity tokens to 1. In particular, the weighted cross-entropy loss was used:

$$\mathcal{L}_{wCE} = -\sum_{i=0}^{C} w_i \cdot y_i \cdot log(p_i)$$

where C represents the number of classes, w_i represents the weight associated to the class i, y_i represents the true label and finally p_i the predicted probability of class i.

This approach allows the usage of the same model and training procedure for both the next activity prediction task and the trace classification task.

3.4 Experiments

To assess the impact of explicit knowledge integration within the transformer architecture, the experimental phase is structured around the following research questions:

RQ1. How does the integration of Structural Positional Encoding (SPE) impact the accuracy of next activity prediction compared to standard positional encoding and traditional baseline architectures?

Figure 3.3: Excerpt of the domain ontology

RQ2. How robust is the transformer model with SPE when exposed to varying levels of noise and incompleteness within the clinical process traces?

RQ3. Qualitatively, does the incorporation of the SPE module steer the model's predictions to better align with established medical ontologies and clinical guidelines?

RQ4. How effective is the transformer architecture when adapted for other downstream process mining tasks, such as trace classification?

The experiments (as mentioned extensively) focus on stroke management applications. Stroke is a very critical medical condition, characterized by an insufficient blood flow to the brain, which can determine cell death. This can be due to ischemia (lack of glucose and oxygen supply) caused by a thrombosis or embolism, or to a hemorrhage. As a consequence, in the acute phase, the patient's life is threatened; moreover, stroke survivors can experience serious adverse events which can lead to permanent disability. Stroke is the leading cause

of adult disability in the United States and Europe and the second leading cause of death worldwide.

The European Stroke Organisation (ESO) has defined a set of criteria that a hospital must meet to be certified as an ESO Stroke Unit (SU), or as an ESO Stroke Center (SC) [117]. While both SUs and SCs must exhibit a set of features, such as the presence of a dedicated stroke unit ward and of a multiprofessional team, SCs must guarantee the execution of less common/additional diagnostic and therapeutic procedures (e.g., transfemoral cerebral angiography or ventricular drainage), on a 24/7 basis. These procedures enable SCs to care for more complex stroke patients, i.e., those who experience rare stroke types or are affected by various comorbidities, by guaranteeing the execution of the required guideline steps within the correct time window.

The experiments were conducted on a dataset containing 5342 stroke process traces; 905 of such traces belong to complex patients, and 4436 to simple ones. All 905 complex patients were treated in an SC; 1724 simple patients were treated in a SC as well; 2712 simple patients were treated at a SU. All the data come from a research activity conducted with a set of hospitals in Northern Italy, which was approved by ethical committees.

The traces exhibit 15 activities on average, from a set of 82 possible activities (see Table 3.1 for some statistics). Figure 3.4 shows an example trace, where the patient, after suffering from stroke onset, reaches the emergency department (ER_arrival). Here, the patient is assessed through exams such as Computed Tomography (CT) without contrast, Electrocardiogram (ECG), and a hematological test. Neurological examination is then performed in order to determine the most appropriate treatment strategy. Thrombolysis (TPA) is then carried out, followed by a second CT without contrast to evaluate its outcome. An anti-coagulant treatment (Aspirin) is finally started before discharging the patient from the emergency department.

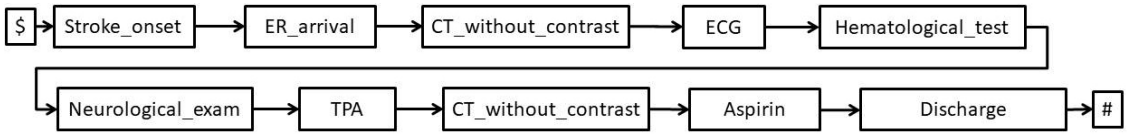

Figure 3.4: Example of a trace. The activities $ and # represent the start and the end of the trace, respectively.

Given that traces have varying lengths (see Figure 3.5), padding was applied to ensure that all sequences had the same number of activities (i.e., the length of the longest sequence) using a special "pad" token at the end of each sequence. Moreover, two special tokens

indicating "start of sequence" and "end of sequence" are appended at the beginning and end of each trace. The ontology associated with the process was parsed in order to obtain a graph containing 110 nodes and 111 edges.

Table 3.1: Datasets traces length statistics measured in number of activities

Traces length statistics	
Mean	15
Standard Deviation	3
Minimum	2
Maximum	25

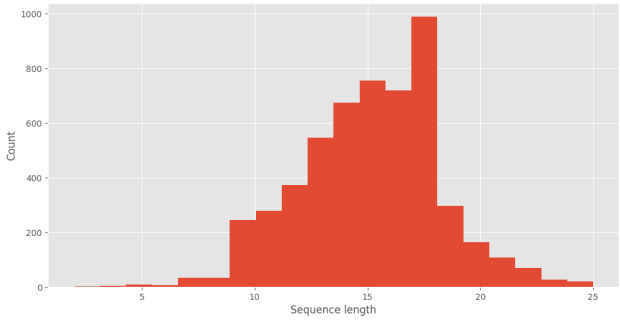

Figure 3.5: Trace length histogram

Notably, during the training phase, examples containing padding tokens or "*start of sequence*" (the "$" activity in Figure 3.4) tokens were excluded from loss calculation. This ensures that the model ignores these examples during the calculation of the gradient, preventing potential degradation of the model's performance. Instead, the "*end of sequence*" (the "#" activity in Figure 3.4) tokens were kept so that the model could learn to predict when a sequence of activities should end.
An 80/10/10 split to the data (training, validation and test) is applied. Hyperparameters were tuned using the Optuna [3] optimization framework. Table 3.2 reports the search space considered and the best parameters found.

3.4.1 Next activity prediction task

The first set of experiments aimed to verify how the model performed in next activity prediction, comparing knowledge injection through SPE, to more classical, state-of-the-art approaches (**RQ1**), namely a baseline model without positional encoding, and the PE model. Moreover, an LSTM model was trained on the same task in order to assess how this classical architecture compares to the transformer model and to determine whether it can also

Table 3.2: Parameter search space used to find optimal hyper-parameters and optimal configuration, by Optuna. Search spaces included between square brackets are discrete sets, while for continuous search spaces, the values are sampled by Optuna space in that range.

Parameter	**Search space**	**Best configuration**
Embedding size	[16, 32, 64, 128, 256]	64
Hidden size	[16, 32, 64, 128, 256]	128
Number of heads	[1, 2, 4, 8]	4
Number of layers	[1, 2, 3, 4, 5]	4
Dropout rate	0.1 - 0.5	0.216375
Optimizer	AdamW	AdamW
Scheduler	StepLR	StepLR
Gamma	0.85 - 0.99	0.989695
Learning rate	1×10^{-2} - 3×10^{-2}	0.002836
Ontology embededing size	[8, 16, 32]	32

benefit from the SPE technique. Such a baseline model is composed as follows: one LSTM layer with a hidden size of 128 and a dropout of 0.39. As for the transformer model, all the hyperparameters were found using Optuna with a similar search space.

For each model configuration, 10 fits with random training-validation-test splits and initialization were performed. The model's performance on the next activity prediction task was assessed using the mean and standard deviation of the accuracy-at-k score, which measures the ability to predict the next token in a sequence given a prefix. In particular, since the model is trained on prefixes of varying lengths, ranging from 1 to n (where n is the sequence length), the results are aggregated using the mean. Additionally, different top-k values were used to evaluate its capability in ranking multiple potential next tokens. This evaluation ensures that the model's prediction and ranking abilities are tested across different scenarios.

The results (see Table 3.3) show that there is a clear benefit in using the SPE method, which substantially improves the model performance across all model sizes. Moreover, the model metrics are stable and saturate with an embedding size of 64, improving only marginally using a 128-sized embedding. This suggests that the model does not tend to overfit.

Another important finding is that the model does not seem to benefit from the classic PE method, which only encodes an ordering of the activities (see Figure 3.7). This observation may be due to the nature of stroke processes, where the next activity depends more on which activities have already been performed rather than their sequential order [68]. As an example, at the beginning of the hospitalization phase, it is necessary to assess the patient's status through ECG, RX, and CT (see Figure 3.3) – independently of their order. Only later is it possible to proceed with the proper pharmacological therapy, based on the collected diagnos-

tic outputs. This also hints at why the model benefits from the SPE technique: activities of a similar type, and aimed at the same goal (in the example, patient assessment, see Figure 3.3), are likely to be executed in close proximity. As a result, the contextual information about an activity's position within the ontology graph describing relations between activities in the process is much more informative than its position within an individual sequence.

Regarding the comparison of the results with the LSTM model (see Table 3.3), it is possible to see that it underperforms with respect to the transformer, particularly in the SPE setting. However, interestingly, it is also clear (see Figure 3.7) that the LSTM benefits from the use of the SPE module.

Table 3.3: Test accuracy-at-k and standard deviation on 10 random initialization and train-val-test splits. Model performance with different positional encoding methods at different embedding sizes are reported

Architecture	**Method**	**Model size**	**Accuracy@1**	**Accuracy@3**	**Accuracy@5**
Transformer	None	16	45.6±0.3	67.1±0.3	75.8±0.3
	None	32	46.9±0.2	69.4±0.3	78.8±0.3
	None	64	47.4±0.4	70.4±0.5	79.8±0.5
	None	128	47.3±0.2	70.6±0.2	80.1±0.3
Transformer	PE	16	45.1±0.4	65.8±0.4	74.7±0.4
	PE	32	46.9±0.2	68.6±0.3	78.4±0.3
	PE	64	47.5±0.1	70.4±0.3	79.9±0.2
	PE	128	47.0±0.2	70.1±0.5	79.8±0.3
Transformer	SPE	16	48.2±0.5	69.7±0.3	77.6±0.4
	SPE	32	52.1±0.4	75.3±0.4	83.4±0.2
	SPE	64	55.1±0.3	79.4±0.6	87.8±0.5
	SPE	128	**56.4±0.5**	**81.5±0.4**	**89.8±0.4**
LSTM	None	128 (best)	46.8±0.2	69.5±0.2	78.8±0.3
	PE	128 (best)	46.7±0.4	69.5±0.4	78.7±0.3
	SPE	128 (best)	52.2±0.3	75.7±0.4	84.1±0.4

To evaluate the approach's robustness to noise and incompleteness in the traces (**RQ2**), which could arise from logging errors in real clinical settings, the performance on two altered datasets was tested. Specifically, by randomly adding noise/incompleteness to 5% and 10% of the traces. This involved modifying the activity sequences by either removing an activity or adding a random new activity.
The results, reported in Table 3.4, highlight that our model loses ∼3% of accuracy on the 5% noisy dataset and ∼8% on the 10% noisy dataset (referring to the 128-sized embedding). Even the standard deviation increases, especially in the mid-sized models, although globally speaking, such a performance decrease (reported by the LSTM model as well) is not very

high. Overall, the approach thus appears to be sufficiently robust to noise/incompleteness.

As a final point, the output of a qualitative evaluation conducted with domain experts is reported through a specific example (**RQ3**). To accomplish this, a query trace is fed to the system, which is tasked with predicting the next activity in the sequence.

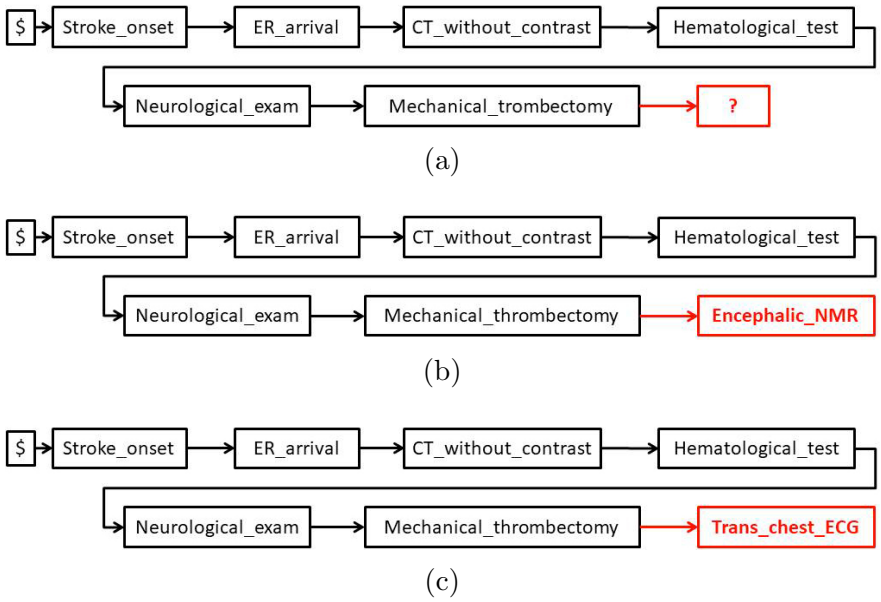

Figure 3.6: (a) Trace given as a query to the transformer. The activity containing the question mark is the one to be predicted. (b) Trace given as a query to the transformer without using domain knowledge, with the predicted next activity colored in red. (c) Trace given as a query to the transformer using SPE, with the predicted next activity colored in red.

The query trace shown in Figure 3.6a describes the therapeutic pathway of a patient eligible for a mechanical thrombectomy, a procedure that allows specialists to remove the thrombus causing the ischemic stroke, through surgery. After stroke onset and emergency (ER) arrival (children of the classes `Stroke_Onset_E` and `ER_arrival_E` classes of the ontology in Figure 3.3, respectively), the first activities follow the standard guideline indications: CT scan (without contrast, in this case), hematological test and neurological evaluation (children of the `Hematological_test_E` and `Exam_E` classes, respectively) are performed for patient assessment. Subsequently, the physician decides to treat the patient through a mechanical thrombectomy (child of the `Interventional_E` class). To support the physician in deciding what should be done after surgery, the system is interrogated to obtain a prediction of the next activity to be taken (the activity which will fill the question mark in Figure 3.6a).

The answers of the transformer are shown in Figures 3.6b and 3.6c, without resorting to

SPE (and thus to domain knowledge) and resorting to it, respectively.

When no domain knowledge is included in the prediction strategy, the transformer suggests performing an encephalic magnetic resonance (NMR), probably to assess the result of the surgical operation. However, the SPREAD stroke clinical guidelines [83] indicate that, after surgery, it is mandatory to monitor the patient's blood pressure (which is related to cardiological functionality) to ensure it remains below 180/105 mm Hg, within the first 24 hours. Even if NMR makes sense, the prediction given as output by the transformer with SPE leads closer to the guideline recommendation. Indeed, trans-chest ECG is predicted in this case, as shown in Figure 3.6c, suggesting that an electro-cardiogram monitoring (which normally includes blood pressure measurement as well) should be performed.

Medical expert assessment confirms the superior accuracy of ECG execution suggestions, as demonstrated by the ontological structure in Figure 3.3. In the ontology, the suggestion provided by the transformer without SPE (`Encephalic_NMR`) is a child of the `NMR_E` class (neuro-imaging tests) which, in turn, is a child of the `Patient_assessment_E` class. Using SPE, we obtain the indication to perform a `Trans_chest_ECG`, located as child of the `ECG_E` class (cardiological monitoring procedures), which is child of the `Patient_assessment_E` class. As a matter of fact, both suggestions are related to a patient assessment procedure, which is expected after brain surgery. However, the suggestion provided by the transformer with SPE (`Trans_chest_ECG`) is actually closer to the guideline recommendation (blood pressure monitoring), since `Trans_chest_ECG` is one of the cardiological monitoring strategies involved in the `ECG_E` group. The procedure suggested without using SPE (`Encephalic_NMR`) is still related to patient assessment, but it is much less specific.

Table 3.4: Test accuracy-at-k and standard deviation on 10 random initialization and train-val-test splits. Model performance with 2 different noisy datasets. The models are trained using the SPE method.

Setup		**5% noise dataset**			**10% noise dataset**		
Architecture	**Model size**	**Accuracy@1**	**Accuracy@3**	**Accuracy@5**	**Accuracy@1**	**Accuracy@3**	**Accuracy@5**
Transformer	16	43.9±0.4	65.1±0.4	74.0±0.4	39.0±0.7	60.2±0.5	69.5±0.5
	32	48.1±0.3	71.5±0.4	80.4±0.4	43.5±2.1	66.3±2.9	75.7±2.9
	64	51.6±0.5	76.6±0.6	84.8±0.5	46.1±2.9	69.7±4.7	79.2±5.1
	128	**53.4±0.6**	**78.3±0.4**	**87.2±0.5**	**48.3±2.3**	**73.9±3.1**	**83.8±3.2**
LSTM	16	37.0±0.4	58.1±0.5	69.0±0.4	33.3±0.6	53.6±0.6	64.7±0.5
	32	42.1±0.1	63.7±0.8	73.0±0.6	36.5±1.2	57.6±1.2	68.1±0.7
	64	46.7±0.5	69.4±0.9	78.0±0.7	42.3±0.6	64.4±0.8	73.4±0.8
	128	49.6±0.4	73.3±0.6	81.9±0.6	45.3±0.9	68.5±1.1	77.8±1.0

3.4.2 Trace classification task

The second set of experiments focuses on evaluating the model's performance in trace classification (**RQ4**). As anticipated, the connection between a trace and its corresponding label

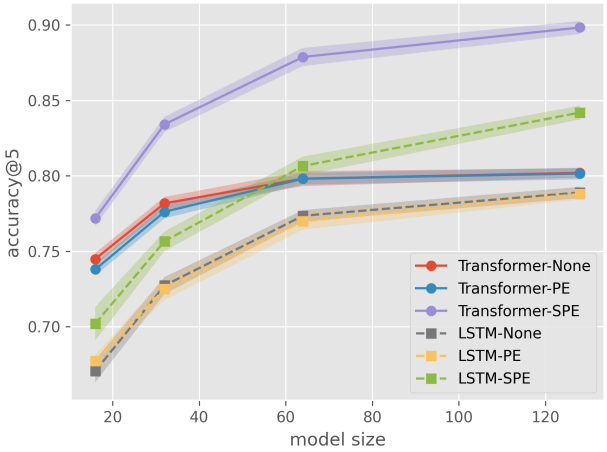

Figure 3.7: Model performance with different embeddings and sizes on testset

is established by appending a class token to the trace when inputted into the transformer model. Then, by increasing the weights in the cross-entropy loss associated with the class token, the model was directed to prioritize the prediction of these tokens (see section 3.3.3).

Two different sets of class labels were assigned to the traces to perform two experiments:

- The first set comprises two classes representing *simple* or *complex* patients, with a highly skewed ratio of 80-20
- The second set of labels is related to the hospital level (SC or SU - explained at the beginning of section 3.4). In this case, the two classes are balanced with a ratio of 53-47.

The transformer model was retrained on the extended datasets, using the optimal hyperparameters found in the previous experiments. Then, the model was evaluated on the prediction of the tokens associated with the classes of interest. To contextualize these results, three baseline methods serve as comparison points: two deep architectures (CNN and LSTM variants) and a random forest ensemble.

In particular, the CNN-based model was already applied in [86] to the same dataset. Such a CNN classifier architecture uses 3 one-dimensional convolution layers, with a kernel size of 3, ReLU activation function, and 16, 32, and 64 filters, respectively. Each convolution layer is followed by a max-pooling layer with a pool size of 2. The output of the last pooling layer is flattened and provided to a layer using the sigmoid activation function to output a final node, representing the predicted class. The LSTM-based model, instead, is composed of two LSTM layers, having tanh activation and 256 and 128 units, respectively. They are

separated by a dropout of 0.6 to reduce overfitting. The output of the last LSTM layer is provided to a fully connected layer, using the sigmoid activation function to output a final node, representing the predicted class.

The obtained results are very encouraging (see Table 3.5); the transformer model demonstrates the ability to surpass all the state-of-the-art baseline models, particularly in the SC versus SU classification experiment. Considering the noisy/incomplete datasets, the classification performance of all the models progressively worsens, as expected, as more noise is injected. However, even in this case, the transformer model maintains an advantage over the competitors.

Table 3.5: Results of the trace classification task. We compare the transformer model with SPE embedding to the CNN-based model, the LSTM-based model, and the random forest model. All the metrics are weighted to account for dataset imbalance

Setup		**Original Dataset**			**5% noise Dataset**			**10% noise Dataset**		
Dataset	**Model**	**Precision**	**Recall**	**F1-score**	**Precision**	**Recall**	**F1-score**	**Precision**	**Recall**	**F1-score**
Simple/complex patients	CNN	0.78	0.78	0.78	0.75	0.75	0.72	0.71	0.68	0.69
	LSTM	0.43	0.65	0.52	0.43	0.65	0.52	0.43	0.65	0.52
	Random Forest	0.75	0.75	0.74	0.72	0.73	0.71	0.70	**0.70**	0.68
	Transformer	**0.83**	**0.83**	**0.83**	**0.77**	**0.78**	**0.77**	**0.80**	0.69	**0.72**
SC/SU hospital	CNN	0.70	0.70	0.70	0.68	0.66	0.66	0.64	0.64	0.64
	LSTM	0.43	0.52	0.36	0.28	0.52	0.36	0.28	0.52	0.36
	Random Forest	0.66	0.66	0.66	0.66	0.66	0.66	0.63	0.63	0.63
	Transformer	**0.72**	**0.72**	**0.72**	**0.69**	**0.70**	**0.69**	**0.68**	**0.65**	**0.65**

3.5 Conclusions

This chapter has presented and validated a knowledge-injection approach for predictive process monitoring and trace classification in medical domains. Through Structural Positional Encoding (SPE), embedding ontological knowledge directly into transformer architectures yields consistent performance improvements over data-driven baselines. Specifically, SPE achieves a 9.1% gain in accuracy-at-1 on next activity prediction and maintains superior performance even under data corruption (10% noise), validating both effectiveness and robustness.

Importantly, qualitative evaluation with domain experts confirms that knowledge-enhanced predictions align more closely with established clinical guidelines than data-only approaches. This finding directly addresses a core limitation of purely data-driven process mining: the inability to ground predictions in formalized medical knowledge. By contrast, the proposed approach ensures that activity suggestions conform to clinical reasoning structures encoded in domain ontologies.

Beyond stroke management, results demonstrate that knowledge injection via SPE benefits diverse architectures (both transformers and LSTMs), suggesting the approach is architecture-agnostic and ontology-agnostic. This modularity indicates applicability across medical domains, provided that formalized knowledge representations are available.

However, this work also reveals important constraints. First, regarding the sequential nature of the data, the current architecture relies on standard sequence ordering rather than explicitly incorporating irregular time-to-event intervals or activity durations. While integrating explicit temporal information, such as through temporal positional encodings or duration-aware tokenization, is a common approach in other predictive monitoring studies, it was not pursued here in order to isolate the evaluation of the SPE module and to mitigate the impact of noisy timestamps present in the dataset. Adapting the method to explicitly handle these temporal dynamics remains a promising direction for future work on datasets with more reliable timing annotations. Furthermore, the stroke domain evaluated here features well-defined sequential processes with relatively modest dimensionality (82 activities, 136 ontology classes). Many medical domains, particularly those involving high-dimensional molecular, genomic, or imaging data, present fundamentally different challenges: the data is not naturally sequential, the feature space is orders of magnitude larger, and domain knowledge must be integrated differently. These constraints motivate the next contribution: extending knowledge-injection mechanisms to high-dimensional multi-omic data, which forms the focus of Chapter 4.

Chapter 4

Knowledge Injection for High-Dimensional Data

While Chapter 3 presented knowledge injection for sequential process data, many biomedical domains present a fundamentally different challenge: high-dimensional static data where traditional feature-driven approaches may fail to capture intricate relationships encoded in domain structure. Microbiome metaomic data exemplifies this problem: datasets typically exhibit extreme dimensionality and sparsity, yet underlying biological relationships are well-defined through taxonomic hierarchies and functional classifications.

This chapter extends the knowledge integration framework to this different data modality through Graph Representation Learning (GRL). Rather than relying only on taxa abundance as input features, the proposed approach directly encodes the structured domain knowledge of microbial taxonomy into a graph structure. Embeddings computed over this taxonomic graph capture not only statistical associations but the evolutionary and functional positioning of genes, species, and genera within the microbiome.

The Graph-Based Encoder. The core of this method lies in constructing a generalizable encoder for microbial taxa networks that operates independently of the downstream classification task. Given metagenomic and metatranscriptomic data, a taxonomic graph is constructed where nodes represent genes, species, and genera, while edges encode known taxonomic relationships. By applying Graph representation learning techniques, it's possible to generate node embeddings that reflect each entity's structural position within the taxonomic network.

Subsequently, a two-level aggregation strategy synthesizes these embeddings into patient-level representations. At the gene level, embeddings are aggregated within taxonomic branches (gene $\rightarrow$ species $\rightarrow$ genus), capturing phylogenetic context. At the patient level, a weighted aggregation based on gene abundance creates a unified, patient-specific microbiome representation. This method is inherently robust to missing data: the model integrates only the information present for each patient, avoiding imputation assumptions.

Experimental Validation on IBD Diagnosis. Experiments on the IBDMDB dataset systematically evaluate the effectiveness of knowledge-injection through graph-based encoding:

1. **Embedding technique comparison**: Laplacian Positional Encoding (LPE) and Node2Vec emerge as the optimal encoding strategies, outperforming standard baselines.

2. **Multi-omics integration**: Incorporating metatranscriptomic data alongside metagenomic data provides complementary biological information. Indeed, the modest but consistent improvement suggests that multiple omics levels encode partially redundant but valuable information.

3. **Feature selection and interpretability**: The method exhibits an optimal operating point around 3500 genes, beyond which additional low-abundance features introduce noise rather than signal. This reveals an important property: despite extreme feature dimensionality, the structured knowledge representation allows effective learning with a constrained subset of features, enhancing interpretability.

Methodological Parallels and Generalization. Notably, Chapter 3 also employs Laplacian eigenvector encoding as the core knowledge-injection mechanism, yet applied to different graph structures: clinical activity ontologies versus microbial taxonomies. In process monitoring, Laplacian eigenvectors encoded activity ontologies; here, they encode microbial taxonomies. In both cases, node proximity in the graph structure, whether activities with shared clinical purpose or taxa with shared evolutionary history, translates to similar learned embeddings. This consistency across distinct domains and data modalities suggests that knowledge injection via structural encoding is a generalizable principle in biomedical machine learning. Building on this principle, the shift toward omics data demonstrates the versatility of this methodological approach across different clinical applications. While transitioning from sequential process logs to multi-omic profiles represents a change in data modality, the underlying mechanism of structural knowledge injection remains the same. Furthermore, this architectural approach provides a clear pathway for scaling to other clinical contexts. Since the model learns an encoder based on a taxonomic structure rather than cohort-specific features, the resulting representations are transferable. This allows the pretrained encoder to be reused in other microbiome studies involving smaller clinical cohorts that would otherwise lack the sample size required to train the architecture from scratch.

4.1 A Network Perspective on Modeling Microbiome Data

Inflammatory Bowel Disease (IBD) is a chronic relapsing inflammatory disease that can be diagnosed using specific microbial signatures that can be identified in the gut microbiome [157]. The gut microbiome, a community of microorganisms residing within the human gut, plays a significant role in health and disease. Advancements in sequencing have yielded vast amounts of metaomic data, detailing the gut microbiome's composition and function. However, effectively analyzing this high-dimensional data remains a challenge [146]. Traditional methods often fail to capture the intricate relationships between the diverse microbial species within the microbiome.

As aforementioned, this work explores the potential of Graph Representation Learning methods (GRL) [54] for analyzing gut microbiome metaomic data. GRL is a set of powerful machine learning models adept at handling graph-structured data, holding promise for unraveling the complex relationships within the microbiome. This work constructs a network capturing the relationships between genes, species, and genera using metagenomic and metatranscriptomic data, built from gene abundance/expression across all patient samples. By leveraging GRL techniques, a latent representations (embeddings) for each entity within the network is learned. These embeddings encode the underlying functional relationships between genes, species, and genera. Finally, patient-specific microbiome representations are generated through aggregation of embeddings weighted by each individual's gene abundance and expression profile. These learned representations serve as input to a binary classifier, discriminating IBD status at the phenotypic level.

The chapter is organized as follows:

- Section 4.2 presents some related work.
- Section 4.3 illustrates the characteristics of the dataset used in the experimental analysis.
- Section 4.4 presents the proposed methodology and learning architecture.
- Section 4.5 is devoted to the obtained experimental results.
- Concluding remarks are finally presented in section 4.6.

4.2 Similar Approaches

A wide range of methods have been recently proposed to analyse microbiome data and use it to obtain predictive models for biomarker discovery and classify patients based on some phenotype [60]. Some of these methods rely on the use of the microbiome features, like the relative abundance or expression of taxa. However, such approaches can be limited by the inherent nature of microbiome datasets: sparsity and high dimensionality (e.g., the presence of many features with a high missing rate). This may lead to overfitting, with models failing to generalize to other datasets [74].

To address these limitations, several works try to leverage the intrinsic phylogenetic or taxonomic information between taxa. For example, PopPhy-CNN [116] adopts a CNN-based model that works on the 2D matrix derived from the phylogenetic tree representation. MIOSTONE [74] uses a gated phylogenetic encoded neural network. TaxoNN [123] leverages an ensemble of CNNs, each specializing in a particular taxonomic phylum.

The proposed method directly builds upon the concept of exploiting taxonomic relationships to create a meaningful representation of the microbiome. This is achieved by employing a GRL-based method to learn a generalized encoder specifically for the network of taxa. This encoder is trained independently from the downstream classification task, which is handled by a separate model.

4.3 Dataset Description

The dataset used in this study is the IBDMDB, described in [94], which focuses on a cohort of more than 1500 patients with the target of IBD diagnosis. The database includes observations from two distinct omics levels: microbiome metagenomics (MGX) and metatranscriptomics (MTX) gene expression levels (expressed in counts per million - CPMs), along with additional metadata related to the patients' health states and the presence or absence of IBD. Complementary taxonomic information concerning species and genera is also included; thus, the complete genomic information consists of the triplet *genes*, *species*, and *genera* (see Figure 4.1).

The dataset exhibits variability in the number of samples and features across omics levels:

- **Metagenomics**: 1,635 samples with 108,433 features.
- **Metatranscriptomics**: 736 samples with 70,711 features.

As already mentioned, patient labels indicate the presence or absence of IBD (binary classification). The dataset includes labels for 1,594 patients and is notably imbalanced, with 83% of samples labeled as negative (absence of IBD) and 17% as positive (presence of IBD).

4.4 Proposed Method

Given the large dimensionality and variable rates of missing data across omics levels, this method leverages graph-based approaches, which are inherently robust to missing samples within a specific omic level and can effectively handle datasets with differing numbers of features. GRL methods naturally represent interactions and connectivity between data points, making them particularly suitable for modeling the complex relationships between microorganisms observed in metaomics data. This capability allows us to capture both within-omic and cross-omic interactions in a unified framework, enhancing predictive power.

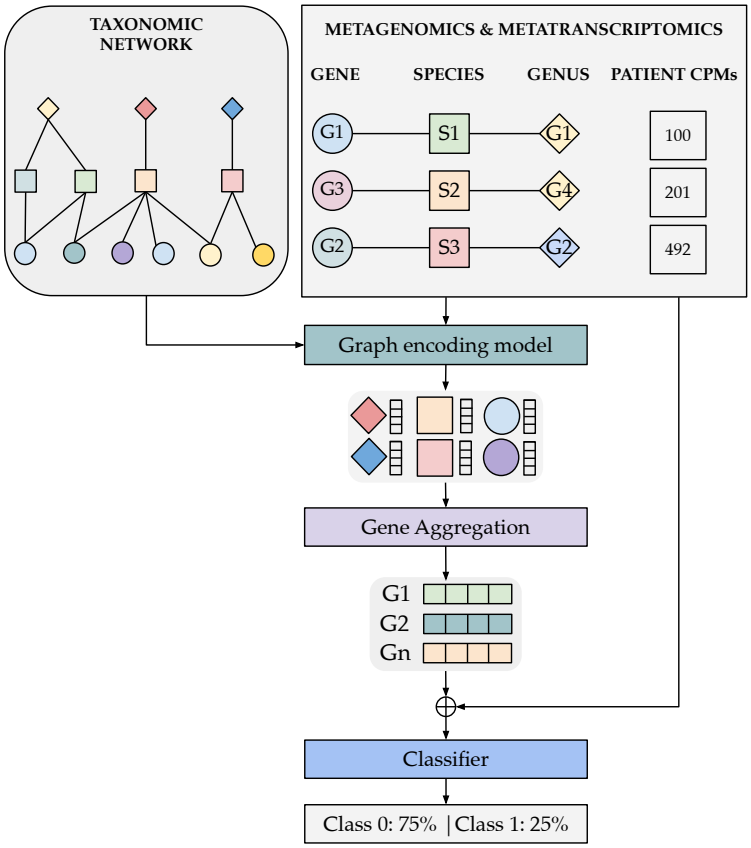

Figure 4.1: Overview of the proposed architecture. The model processes two inputs: a taxonomic network (left) and multi-omic microbiome data (right). Based on our experimental dataset, the multi-omic input comprises 1594 patient samples, characterized by 108,433 metagenomic and 70,711 metatranscriptomic features. A graph encoding module first generates embeddings to capture relationships between microorganisms. Features are then aggregated in two stages: first, along taxonomic branches to create gene/transcript-level representations, and second, across genes/transcripts using patient-specific relative abundances. These final patient embeddings are passed to the classifier for IBD prediction.

The objective is to obtain a meaningful representation of a patient using the metaomic data coming from the analysis of gut microbiome samples. To do this, the idea is to establish relationships among the various microorganisms present in the microbiome, whose presence is determined by the abundance of genes/transcripts that characterize them, using the relational information of species and genera encoded within a graph.

A key strength of this approach lies in its robustness to missing data, either from specific features, such as a particular set of genes, or entire omic levels, such as genomic or transcriptomic data. Robustness arises from the aggregation process, which inherently relies only on the information available for a given patient. Rather than employing explicit mechanisms to encode or impute missing features, the model integrates the representations of the sub-

graphs derived from the data. By ensuring that patient representations are constructed solely from the existing data, it maintains consistency and avoids assumptions about missing information. Figure 4.1 provides a detailed overview of the proposed pipeline and architecture.

The following subsections will describe how to get the representation of a patient starting from the metagenomic level. The same technique can be applied in the same way to other omic levels and seamlessly integrated into the proposed pipeline.

4.4.1 Graph Construction

Given a set of patients with gene abundance levels of gut microbiome microorganisms, and associated taxonomic data formatted as triplets (*gene*, *species*, *genus*), we construct a single, global taxonomic graph $\mathcal{G} = (V, E)$ that is shared across all instances.

- V denotes the set of all distinct nodes, encompassing every gene, species, and genus observed across the entire dataset.
- E represents the set of hierarchical edges, categorized into two types: `(gene, species)` and `(species, genus)`.

Because the underlying ontology is shared, $\mathcal{G}$ acts as a global reference structure rather than being instantiated separately for each microorganism. Once this global taxonomic graph is constructed, each patient P_i is represented as an instantiated subgraph. Specifically, a patient is defined by a subset of active nodes $V_i \subseteq V$, comprising the specific taxonomic lineage for which the patient exhibits a gene abundance level greater than 0.

4.4.2 Graph Representation Learning Module

Given a graph $\mathcal{G}$ described as above, we aim to learn a representation (embedding) of the nodes that reflects their connectivity within the network. To this end, graph representation learning methods are employed, specifically: Graph Laplacian Eigenvector Positional Encoding, Random Walk Positional Encoding, and Node2Vec.

Here, a description of each method is given.

Graph Laplacian Eigenvector Positional Encoding (LPE). This technique relies on the factorization of the graph's Laplacian matrix. The embedding for a node is defined by the k-smallest, non-trivial eigenvectors associated with that node [43]. These eigenvectors

capture the structural properties of the graph, such as community structure and global connectivity. The LPE method is particularly advantageous in capturing global structural information about the graph, making it useful for understanding the relationships between distant nodes in the network. A formal description of how it is calculated and implemented is available in Section 3.3.2.

Random Walk Positional Encoding (RWPE). This approach, as described in Dwivedi et al. [43], utilizes the random walk matrix to generate node embeddings. A random walk is a stochastic process that involves a sequence of steps through the graph, where at each step, the next node is chosen randomly based on the adjacency structure of the graph. The RWPE method computes the node embeddings by analyzing the transitions observed during a random walk.
Specifically, given the aforementioned graph $\mathcal{G} = (V, E)$ the adjacency matrix is defined as $A \in \mathbb{R}^{n \times n}$ where $n = |V|$ and $A_{ij} = 1$ if node i and node j are connected by an edge in E. The degree matrix of $\mathcal{G}$ is defined as $D \in R^{n \times n}$ where $D_{ii} = \sum_i^n = A_{ij}$. The RWPE is defined as a matrix $\text{RWPE} \in \mathbb{R}^{n \times k}$ where k is a hyperparameter representing the length of the random walk where:

$$\text{RWPE}_i = [\text{RW}_{ii}, \text{RW}_{ii}^2, \ldots, \text{RW}_{ii}^k]$$

Where $\text{RW} = AD^{-1}$ is the random walk operator. So the i component of the positional encoding embedding of a node j represents the probability of terminating on the node j during a random walk of the length i.

Node2Vec (N2V). This method, presented in Grover and Leskovec [51], is a semi-supervised algorithm that leverages random walks on the graph to learn meaningful features for the nodes. The primary idea behind Node2Vec is to perform biased random walks on the graph and then use the skip-gram model (originally popularized in natural language processing) to learn embeddings that reflect the node's role in the graph.
In the following, an implementation-focused definition of the algorithm will be proposed, which will result in the same embedding objective of the paper up to negative-sampling approximation.

Node2Vec operates by learning a function $f : V \to \mathbb{R}^d$ which minimizes a stochastic binary classification objective over positive and negative random-walk samples.

The neighborhood sampling operates as follows. For each node $u \in V$ the algorithm generates:

- **Positive random walks** $\mathrm{rw}_{pos}(u) = (u, v_1, \ldots, v_L)$ sampled via a biased second-order random walk controlled by parameters p and q which control how the steps of the random walk are sampled. After moving from node t to v, the next step to $x \in \mathcal{N}(v)$ is chosen with probability:

$$P(x|t,v) \propto \alpha_{pq}(t,x), \quad \alpha_{pq}(t,x) = \begin{cases} 1/p & \text{if } d_{tx} = 0, \\ 1 & \text{if } d_{tx} = 1, \\ 1/q & \text{if } d_{tx} = 2, \end{cases}$$

  where d_{tx} is the shortest-path distance between t and x.

- **Negative random walks** $rw_{neg}(u)$ formed by replacing the context nodes v_i in the walk with nodes sampled uniformly at random from V.

Each walk of length L is sliced into overlapping context windows of size k, defining training pairs (u, v_i).

The **training objective** is defined as: for each positive pair (u, v) and an equal number of negative pairs (u, v^-), the embeddings are optimized via the logistic loss

$$\mathcal{L} = -\mathbb{E}_{(u,v)\sim\mathrm{pos}}\big[\log \sigma(f(u)^\top f(v))\big] - \mathbb{E}_{(u,v^-)\sim\mathrm{neg}}\big[\log(1 - \sigma(f(u)^\top f(v^-)))\big]$$

where $\sigma(\cdot)$ is the sigmoid function.

By then optimizing the node embeddings via stochastic gradient descent, the algorithm encourages nodes that co-occur within biased random walks (positive samples) to have similar embeddings, while pushing apart randomly paired nodes (negative samples). The parameters p and q bias the random walks toward breadth-first or depth-first exploration, interpolating between homophily-based and structural-equivalence embeddings.

4.4.3 Aggregation Function

Once the embeddings for all the nodes in the graph have been computed, a two-tier aggregation methodology is employed, which aims at synthesizing node embeddings at both the gene and patient levels. At the gene level, the embedding is derived from the taxonomic subgraph corresponding to the species and genus of the gene. Specifically, the gene's embedding is calculated as the mean of the embeddings of all nodes within its taxonomic branch (i.e., enzyme, species, and genus), ensuring that the gene's representation reflects both its

intrinsic properties and its evolutionary context. The aggregation step aims at capturing key phylogenetic relationships that are essential for understanding gene function in a broader biological network.

At the patient level, the aggregation process focuses on converting the set of expressed genes into a single, unified embedding representing the patient. A hyperparameter k is introduced to determine the number of the most highly expressed genes to be included in the aggregation. The aggregation is weighted according to the abundance levels of these k selected genes, such that more abundant genes have a greater influence on the final patient representation. To further refine this process, a softmax function is applied to the gene abundance counts, normalizing the abundance levels to mitigate any potential biases introduced by extreme values. This dual-level aggregation framework allows for the creation of patient-specific embeddings that are both biologically meaningful and computationally robust; the impact and the role of the hyper-parameter k is investigated in Section 4.5.3.

Finally, the classification task using patient representations (embedding) as input features has been implemented through a Support Vector Machine (SVM) classifier using a Radial Basis Function (RBF) kernel.

4.4.4 Omic Levels Integration

Previous sections have outlined the methodology for transforming gene abundance data, obtained from a metagenomic (MGX) analysis, of a patient's microbiome into a single patient representation. However, incorporating metatranscriptomic (MTX) features is straightforward. It is sufficient to include genes identified at the transcript level during the construction of the taxonomic graph, and these additional nodes are then considered when generating the patient representation.

4.5 Experimental results

To evaluate the performance of the proposed method, experiments were conducted using the IDMDB dataset described in Section 4.3. These experiments investigated the method's effectiveness in predicting IBD presence or absence under various parameter settings. The following key questions are addressed:

RQ1. Node Embedding Comparison: which node embedding technique (LPE, RWPE, N2V) yields the best performance?

RQ2. Impact of Multi-Omics Integration: how does the model perform with respect to the chosen baseline, and how does the model's performance vary when using different omic levels as input (metagenomics only versus metagenomics combined with metatranscriptomics)?

RQ3. Gene Selection Strategy: how does the number of genes considered during patient representation generation impact the final model performance?

4.5.1 Node Embedding Techniques Evaluation

To assess the performance of different node embedding techniques (LPE, RWPE, N2V), an 80/10/10 train-validation-test split was employed. For each technique, model hyperparameters are optimized using the Optuna framework [3], as shown in Table 4.1.

Following hyperparameter tuning, each optimal model configuration is evaluated over independent runs using 10-fold randomized train-validation-test splits and random initialization. The proposed method is compared against four baseline models: Logistic Regression, Random Forests, Support Vector Machine (RBF kernel), and Multi-Layer Perceptron. To enable baseline models to leverage the MGX+MTX setting, MTX features are concatenated to MGX features, with missing values set to zero.

Model evaluation employs F1-score and ROC-AUC (Receiver Operating Characteristic Area Under the Curve) metrics with macro averaging. F1-score is selected due to dataset imbalance, as it provides a balanced measure between precision and recall. ROC-AUC is chosen for its threshold-independent assessment of discriminative ability between classes, which is particularly relevant for medical applications where different classification thresholds may be preferred.

Results concerning **RQ1** can be read from Table 4.2, by looking at the three bottom rows. It is clear from such results that the best choices are LPE and N2V, which perform very similarly on both datasets, while RWPE significantly underperforms compared other methods.

4.5.2 Impact of Omic Levels

The quantitative influence on performance given by each omic level (**RQ2**) was obtained by training two distinct models. The first used only data from the metagenomics level, while the second incorporated data from both metagenomics and metatrascriptomics levels.

Table 4.1: Parameter search space used to find optimal hyperparameters for every dataset and model configuration. Search spaces included between square brackets are discrete sets, while for continuous search spaces, values are sampled by Optuna within that range.

Parameter	**Search Space**	**Best configuration – MGX**			**Best configuration – MGX+MTX**		
		LPE	**RWPE**	**N2V**	**LPE**	**RWPE**	**N2V**
LPE – Embedding size	[16, 32, 64, 128]	64	–	–	64	–	–
RWPE – Embedding size	[16, 32, 64, 128]	–	64	–	–	64	–
N2V specific							
Embedding size	[16, 32, 64, 128]	–	–	64	–	–	64
Walk length	5–20	–	–	20	–	–	20
Context size	5–20	–	–	18	–	–	16
Walks per node	1–10	–	–	8	–	–	5
Negative samples	1–10	–	–	1	–	–	1
SVM specific							
Complexity (C)	1–3	2.86	2.83	2.13	1.26	1.29	1.12
Kernel type	["rbf", "poly", "linear"]	rbf	rbf	rbf	rbf	rbf	rbf
Shrinking	[True, False]	True	True	True	True	True	False
Tolerance	1×10^{-10}–1×10^{-5}	8.71×10^{-8}	2.66×10^{-8}	2.34×10^{-7}	1.61×10^{-9}	2.53×10^{-8}	2.10×10^{-10}
Use softmax	[True, False]	True	True	False	True	True	False
Standardize data	[True, False]	False	True	False	False	False	True

Importantly, the evaluation for RQ2 employed the same experimental setup established for RQ1 (detailed in the previous section). The corresponding results for both models are presented in Table 4.2. From the results, it is possible to notice that the proposed GRL approach clearly outperforms selected baseline methods; moreover, the model trained using only MGX data slightly underperforms with respect to the model that also includes MTX data.

Table 4.2: Results obtained by the models using different methods for encoding the taxonomic graph (95% confidence interval).

	MGX only		**MGX + MTX**	
Model	**F1 score**	**ROC AUC**	**F1 score**	**ROC AUC**
Logistic Regression	62.69 ± 4.93	80.28 ± 2.34	62.43 ± 5.84	80.55 ± 3.35
Random Forest	80.44 ± 4.48	85.83 ± 3.80	$\underline{80.21 \pm 4.79}$	$\underline{85.52 \pm 4.05}$
SVM	66.34 ± 4.73	79.55 ± 3.36	66.12 ± 4.26	79.11 ± 3.44
MLP	65.32 ± 6.22	79.49 ± 3.87	60.50 ± 4.73	76.72 ± 2.78
LPE	$\mathbf{82.81 \pm 1.81}$	$\mathbf{86.91 \pm 2.11}$	$\mathbf{82.36 \pm 2.52}$	$\mathbf{87.18 \pm 2.72}$
RWPE	67.61 ± 3.30	72.73 ± 2.87	64.78 ± 2.37	69.38 ± 2.33
N2V	$\underline{81.04 \pm 2.26}$	$\underline{86.65 \pm 1.95}$	78.80 ± 3.28	84.73 ± 2.76

4.5.3 Gene Selection Strategy

Regarding **RQ3**, focus is placed on the model trained using LPE encoding for the taxonomic graph. Patient representations are obtained by training the model with varying numbers k of genes as features, specifically the top-k most abundant/expressed genes. Results are compared between the two datasets (MGX-only and MGX+MTX). As shown in Figure 4.2,

model ROC-AUC initially increases with the number of genes, peaks at a certain point, and subsequently decreases slightly before stabilizing. This pattern likely reflects the importance of highly abundant/expressed genes for class separation, while genes with lower abundance/expression levels contribute less discriminative information.

Interestingly, despite a small difference, the optimal number of genes differs between the datasets. The MGX-only model achieves its peak ROC-AUC with 3568 genes, while the MGX+MTX model peaks at 3366 genes. This might suggest that the MTX data provides additional useful information, resulting in a smaller number of necessary genes.

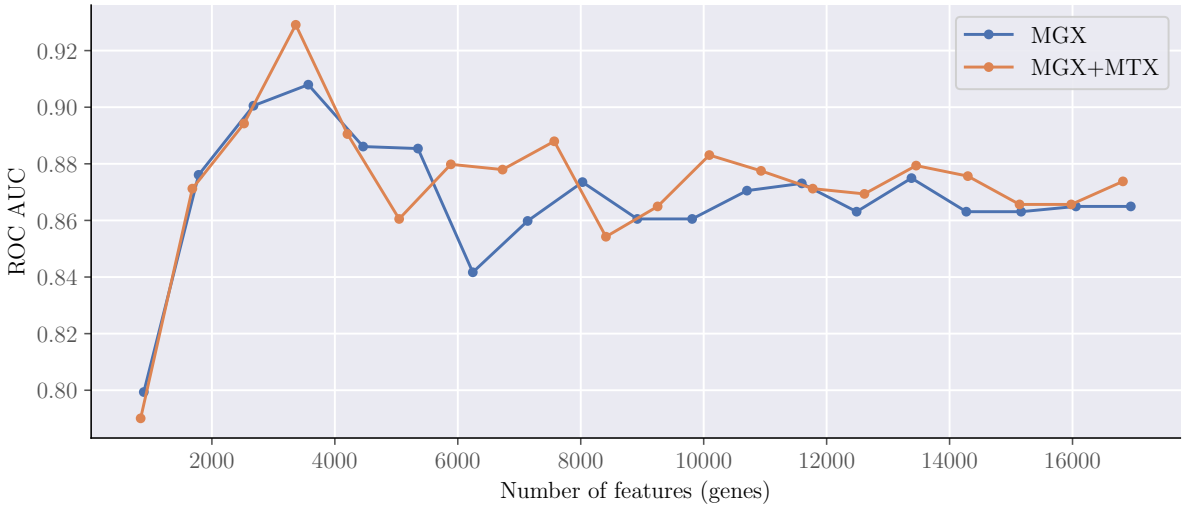

Figure 4.2: The plot compares the ROC AUC achieved by the model trained on two different datasets (MGX and MGX+MTX) by varying the number k of selected (i.e., most expressed) genes. In this case, the graph encoder used the LPE technique for the node embedding representation.

4.6 Remarks and next steps

This chapter extended the knowledge-injection framework to high-dimensional static biomedical data through graph representation learning applied to microbiome metagenomics and transcriptomics. By embedding taxonomic relationships into a graph-based encoder, the approach enables structured integration of biological domain knowledge—phylogenetic hierarchies and functional relationships among microbial taxa—directly into the learning mechanism. This addresses a fundamental difference from Chapter 3: while process monitoring involves naturally sequential, moderate-dimensional data, microbiome analysis requires managing high-dimensional feature spaces where domain structure is non-sequential and ontologically defined.

The resulting microbiome encoder demonstrated strong performance on phenotype classification tasks, validating that knowledge-injection principles extend beyond sequential process data to static, high-dimensional biomedical domains.

Future Directions. This work represents a first step toward modeling microbiome data through graph-based architectures that systematically leverage taxonomic structure. Several promising extensions emerge from this foundation. First, the current encoder could be generalized by training on a substantially larger portion of the gene taxonomy, including organisms not present in the IBD dataset. This would enable the model to learn more refined embeddings that capture nuanced functional relationships, including interactions between microorganisms absent from the training data. Such a generalized encoder could then be tested across different patient cohorts and disease phenotypes, assessing whether learned representations transfer to new clinical contexts. Concurrently, the graph structure itself could be enriched by incorporating additional taxonomic hierarchy levels, providing finer-grained biological context. Scaling graph representation learning methods to handle graphs with orders of magnitude more nodes and edges presents a significant technical challenge, but also creates opportunities to experiment with more sophisticated GNN architectures capable of capturing complex multi-scale biological relationships. A longer-term objective involves integrating heterogeneous biological knowledge by incorporating databases such as Gene Ontology, as well as expanding the graph to encompass exposome-related data. Capturing environmental exposures, dietary habits, and lifestyle factors as interconnected nodes would transform the current taxonomy-focused graph into a highly comprehensive knowledge structure. This expansion is particularly promising for modeling complex chronic diseases, where the interplay between the patient's microbiome and their exposome plays a fundamental role in disease progression. Furthermore, such heterogeneous graphs could leverage large language models to encode textual information as node features, enabling the framework to ground predictions not only in taxonomic and environmental relationships but also in broader biomedical knowledge.

While these directions promise to improve knowledge-grounded approaches, Chapters 3 and 4 share a fundamental assumption: that structured domain knowledge already exists in a usable form. In both cases, we injected pre-existing ontologies into model architectures. Yet clinical knowledge is not always formalized or readily available in structured formats suitable for machine learning. Guidelines exist as natural language documents, and established reasoning patterns are embedded in clinical practice rather than explicitly codified. This raises two critical questions: how do we systematically extract and formalize clinical knowledge

from existing sources, and how do we evaluate whether AI models reason in ways congruent with that knowledge? These questions motivate the final contribution. Chapter 5 presents a framework that addresses both the extraction and evaluation gaps: a methodology for mining structured clinical reasoning pathways from guideline documents and using these pathways as benchmarks to assess whether models demonstrate clinically sound reasoning aligned with established medical practice.

Chapter 5

A Framework for Extracting and Evaluating Clinical Reasoning

The previous chapters have explored methods for injecting structured biomedical knowledge into machine learning models—whether through structural positional encodings for process modeling (Chapter 3) or graph-based representations for high-dimensional multi-omics data (Chapter 4). In both cases, the knowledge itself was treated as given: ontologies, knowledge graphs, and taxonomic structures were expected to be available in a well-defined, machine-readable format. The central challenge addressed was architectural: how to effectively integrate this knowledge into neural network representations to enhance model performance and interpretability.

However, structured knowledge is not always present in real-world scenarios. First, medical knowledge may be represented in unstructured or semi-structured formats like clinical guidelines, medical textbooks, and expert decision protocols that are rich in domain expertise but not immediately usable. Second, even when it exists, conventional evaluation methodologies for clinical AI systems rarely leverage this knowledge to assess whether models truly reason in accordance with established medical principles.

This chapter addresses both challenges by presenting HealthBranches [33], a comprehensive framework that spans the entire lifecycle of knowledge management: from the extraction of clinical reasoning pathways to their use in evaluating model behavior. Unlike the injection-focused approaches of Chapters 3 and 4, HealthBranches tackles the complementary issues of where knowledge comes from and how to verify that models actually use it.

Knowledge Extraction. HealthBranches' first contribution is a semi-automated pipeline that extracts graph-structured decision pathways from medical reference materials, such as clinical textbooks that contain algorithmic frameworks for patient management. This process transforms clinical protocols into graphs, specifically decision trees, where each node represents a diagnostic or therapeutic step, and edges capture the logical flow between them.

The extraction phase addresses a fundamental gap in knowledge-grounded AI: the need for principled methods to convert knowledge from its format (prose, flowcharts, clinical algorithms) into representations suitable for model training and evaluation. By enumerating root-to-leaf paths through these decision graphs, HealthBranches generates realistic clinical scenarios based on validated clinical reasoning chains. This approach enables the creation of benchmark datasets for underrepresented clinical situations without requiring extensive manual curation or access to sensitive patient data.

Importantly, the extracted knowledge structures serve dual purposes: they provide a foundation for generating synthetic yet clinically realistic cases and establish the ground-truth

reasoning paths against which model outputs can be evaluated.

Structured Evaluation. The second contribution of HealthBranches is its use of extracted paths to enable reasoning-aware evaluation of large language models in medical domains. The resulting dataset contains 4,063 question-answer pairs across 17 healthcare domains, with each example explicitly linked to its underlying clinical decision pathway.

Most existing question-answering datasets assess models solely on final answer correctness, lacking mechanisms to verify whether the reasoning process aligns with clinical guidelines. Quiz-based evaluations, in particular, can be vulnerable to models that reach correct answers through inappropriate reasoning paths.

HealthBranches addresses this limitation by incorporating the complete reasoning path for each question-answer pair, allowing evaluation of models' multi-step inference capabilities. In the open-ended question format, this structured knowledge is leveraged through an LLM-as-a-judge evaluation framework that assesses not just semantic similarity to ground-truth answers, but also adherence to the underlying clinical reasoning pathway.

5.1 LLM-Based Medical Question-Answering: Capabilities and Limitations

Large Language Models (LLMs) have significantly advanced Natural Language Processing (NLP) tasks such as translation, summarization, information retrieval, and question-answering (Q&A) [2, 106, 107]. In healthcare, LLM-based Q&A systems provide information on diseases, prevention, and medical services [14].

Deploying LLMs in clinical contexts remains challenging due to accuracy limitations, biases, outdated knowledge, and hallucinations, where plausible but incorrect outputs can misinform care and endanger patients [6, 23, 124, 148]. To address these risks, strategies such as fine-tuning [114], prompt engineering [91], and particularly Retrieval-Augmented Generation (RAG) [87] have been explored. RAG enhances factuality by grounding answers in external knowledge sources [46, 53, 64, 69] and has shown benefits in medical Q&A for improving reasoning and personalization [47, 76, 145].

Recent work highlights the role of Knowledge Graphs (KGs) in structuring domain-specific knowledge to reduce noise and support interpretable reasoning [72, 80, 113, 147]. However, existing benchmarks rarely integrate structured reasoning paths with Q&A tasks, limiting

the evaluation of models on real-world diagnostic complexity.

HealthBranches is a reasoning-focused medical Q&A benchmark that pairs realistic patient scenarios with explicit decision-path chains, enabling evaluation of stepwise clinical inference rather than mere factual recall. Built via a semi-automated pipeline and refined with LLM and expert review, HealthBranches comprises 4,063 cases across 17 domains and supports both open-ended and multiple-choice modalities. Throughout the experiments, several LLMs in zero-shot and RAG settings are evaluated and consistent gains are observed when models are given the reasoning path, underscoring the value of path-aware evaluation for trustworthy, interpretable medical Q&A.

5.2 The current state of clinical Q&A benchmarks

Medical Q&A benchmarks vary widely in format and scope. Closed-ended multiple-choice datasets such as HeadQA, MedQA, MeDiaQA, and PubMedQA focus on fixed-answer formats for dialogue or biomedical text analysis [75, 77, 128, 140], while open-ended datasets like MedCalc-BENCH and PathVQA emphasize free-form responses but require manual verification [58, 79]. Recently, multi-task and multi-modal benchmarks have integrated textual and visual inputs to test reasoning under zero- or few-shot conditions [59, 159], and query-based datasets have emerged for structured data such as EHRs [109]. Other efforts include curated biomedical resources like UltraMedical, which aggregates manual and synthetic datasets for LLM fine-tuning [155], and SM3-Text-to-Query, the first multi-model benchmark for SNOMED-CT–based queries [126].

As summarized in Table 5.1, most existing datasets either assess knowledge recall or offer limited explanation structures. MedQA and PubMedQA test factual knowledge without reasoning paths, while MedCalc-BENCH provides explanations but focuses on quantitative calculations. Similarly, ACI-Bench targets ambient clinical intelligence by benchmarking AI-assisted note generation from doctor–patient dialogues, emphasizing narrative documentation rather than stepwise decision-making [152]. In contrast, HealthBranches uniquely combines multiple-choice and open-ended formats with explicit reasoning paths from clinical decision graphs, allowing for the evaluation of non-computational, interpretable, and step-by-step reasoning essential for trustworthy medical LLMs [152].

As presented in Table 5.1, benchmarks that include richer supervision signals (e.g., calculus, images, graphs, and explicit reasoning paths like ours) are typically smaller than simple QA corpora (e.g., MedQA) due to the substantially higher curation effort.

Dataset (Year)	# questions	Type of questions	Knowledge	Qual. Reasoning	No-Comput.	Explanation
HeadQA (2019) [140]	6,765	Multiple choice	✓	✓	✓	✗
MEDCALC-BENCH (2024) [79]	1,000	Open-ended	✓	✓	✗	✓
MeDiaQA (2021) [128]	23,048	Multiple choice	✓	✗	✓	✗
MedQA (2020) [75]	61,097	Multiple choice	✓	✓	✓	✗
MedXpertQA (2025) [159]	4,460	Multi-modal questions	✓	✓	✓	✗
MIMIC-SPARQL (2020) [109]	10,000	Table- and graph-based queries	✓	✗	✗	✗
PathVQA (2020) [58]	32,799	Open-ended	✓	✗	✓	✗
PubMedQA (2019) [77]	273,500	Multiple choice	✓	✓	✓	✗
HealthBranches (2025)	4,063	Multiple choice & open-ended	✓	✓	✓	✓

Table 5.1: (✓= present, ✗= absent.) Summary of existing medical Q&A datasets and comparison with **HealthBranches**, considering four qualitative dimensions: (1) **Knowledge**: whether the dataset tests knowledge to a particular domain; (2) **Qualitative Reasoning**: whether the dataset tests logical or conceptual reasoning rather than quantitative computation; (3) **No Comput.**: whether the dataset does not require quantitative calculations, formulas, or numeric estimation; (4) **Explanation**: whether the dataset includes a step-by-step justification or reasoning trace. While these criteria are inherently qualitative, we follow prior precedent in MedCalc-BENCH [79].

5.3 Proposed Methodology

The proposed methodology aims to enhance the reliability of LLMs in specialized medical domains by grounding Q&A generation in structured knowledge sources to ensure factual consistency and interpretability. As shown in Figure 5.1, the pipeline consists of three main stages: (1) parsing medical decision pathways, (2) generating Q&A pairs from extracted reasoning paths, and (3) refining outputs through automated and human-in-the-loop validation.

The following section presents the properties of the dataset and the evaluation setup used to benchmark LLM performance.

5.3.1 Dataset

The proposed semi-automated pipeline builds medical Q&A datasets from structured knowledge sources, addressing challenges like data scarcity and privacy concerns. The method utilizes medical reference materials (e.g., textbooks) with textual descriptions of patient treatment protocols and decision graphs that formalize clinical decision-making processes. By sampling realistic decision paths, the pipeline generates plausible patient scenarios that provide a foundation for meaningful Q&A based on authoritative medical knowledge.

The resulting dataset and pipeline can be used in many different scenarios:

- Creating educational resources for medical students.
- Evaluating the proficiency of current LLMs in medical reasoning tasks.

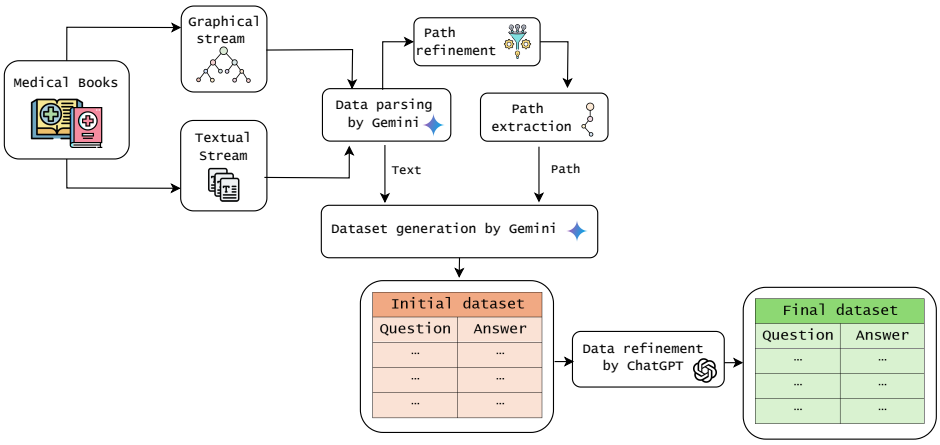

Figure 5.1: A step-by-step workflow for building a Q&A dataset. *Parsing the knowledge source*: textual and graphical streams are extracted from medical books and parsed. *Path extraction and refinement*: graph-based representations are processed to extract and refine relevant semantic paths. *Q&A generation*: an LLM is prompted with textual and path information to generate initial question–answer pairs. *Q&A refinement*: the dataset is further refined using another LLM to ensure consistency and quality.

- Supporting the development of clinically grounded decision-support systems.

Figure 5.1 provides an overview of the pipeline, with implementation details in Appendix A.3. Figure 5.2 shows an example where a clinical path on dyspnea is transformed into a realistic scenario and multiple-choice question, with the correct answer derived directly from the structured reasoning, demonstrating the pipeline's effectiveness.

Parsing of the Knowledge Source. The knowledge source consists of medical textbooks [48, 56, 105] divided into sections, each detailing the management of a patient with a specific symptom or condition through a clinically validated algorithmic framework. These frameworks are typically graph-based representations (often decision trees), where each node corresponds to a diagnostic or therapeutic step (see Figure 5.2). This structure encapsulates the clinical reasoning and domain knowledge for one of the 17 covered domains, while remaining applicable to other medical areas, ensuring broader generalizability.

Each section provides two complementary information streams: the Textual stream, a narrative of clinical rationale and procedural steps, and the Graphical stream, a decision tree organizing the reasoning process. Both are automatically extracted using Gemini-flash 2.0 [132], producing plain-text files for content and graph files encoding the decision structure.

Notably, even if in this context the main source of structured knowledge is represented by these textbooks, the same pipeline can be applied in a straightforward manner to any

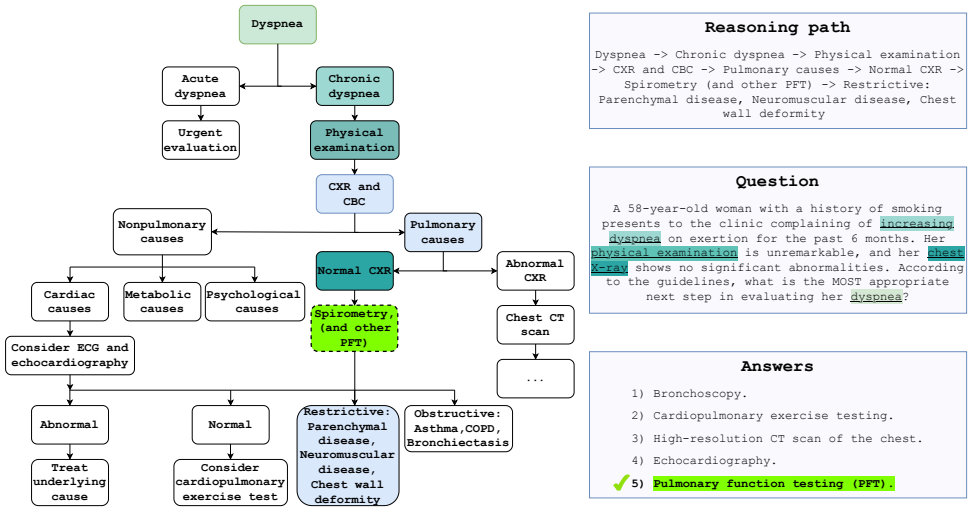

Figure 5.2: Detailed example of Q&A generation process, starting from extracting a reasoning path. The left side shows an example graph outlining treatment steps for dyspnea, while the right side displays the extracted reasoning path and the resulting Q&A pair.

resource that entails structured knowledge.

Path Extraction and Refinement. From each graph, we enumerate root-to-leaf traversals, retaining up to two distinct paths per leaf to balance coverage and dataset size. Each path is then refined with Gemini-flash 2.0 (Appendix A.3), which normalizes terminology and removes formatting artifacts while preserving clinical semantics.

Q&A Generation. The dataset's Q&A content was generated by prompting Gemini-flash 2.0 with the text for a specific condition and its reasoning path. First, Gemini produced an open-ended answer based on the patient scenario, resulting in a detailed response. This answer was then used as the correct option in a multiple-choice format. Gemini also generated four distractors, which were clinically plausible within the scenario but invalid according to the reasoning path, making them more challenging than random alternatives. Each reasoning path is linked to a Q&A pair and distractors, enabling benchmarking in both open-ended and multiple-choice formats.

Q&A Refinement. The generation process may produce flawed examples, such as unanswerable questions, inconsistent answers, or details misaligned with clinical guidelines. To examine the impact of model scale on question difficulty and robustness, Llama 3.1 (405B) [50] was incorporated in the refinement phase, as it exceeds the capacity of the benchmarked models and helps identify intrinsically flawed items.

A two-stage refinement procedure was implemented: first, we collected questions misanswered by both Llama 3.3 (70B) and Llama 3.1; from this subset, we retained only those that at least half of the benchmark models answered incorrectly, resulting in 1,203 items, individually proofread by five reviewers. Specifically, each item was then submitted to GPT-4o with web search and reasoning enabled, refined according to a defined protocol, and subsequently reviewed by a human evaluator. The review protocol includes a standardized prompt and a Q&A template for submitting the item to be corrected to GPT. The results were compared to determine whether to modify the question, adjust the correct option, or discard the item if fundamentally flawed.
The motivation for combining ChatGPT, Gemini, and Llama 3.1 (405B) throughout the pipeline (generation, flawed-path identification, and refinement) was to reduce systematic biases from relying on a single model, thereby enhancing robustness and generalization.

5.3.2 Evaluation

The evaluation strategies for publicly available LLMs were applied to both the quiz-based and open-ended versions of the dataset. Models were tested in two settings:

- **Topline**: receiving (a) only the reasoning path, (b) only the textual description, or (c) both as context, to isolate the contribution of structured reasoning.
- **Benchmark**: answering with the standard zero-shot Q&A prompt (question only), with and without RAG context.

The topline setting aims at measuring how strongly models rely on the reasoning path and, in the open-answer case, verifies whether the reasoning is sound rather than just the final answer.

Models. The benchmark study on the HealthBranches dataset included ten open, decoder-only LLMs that are freely available and deployable via the Ollama platform: Mistral 7B [73], Llama 2 7B [137], Llama 3.1 8B [49], Gemma 7B [133], Gemma 2 9B [134], Gemma 3 4B [135], Qwen 2.5 7B [151], DeepSeek-R1 8B [52], Phi-4 14B [1], and Mistral NeMo 12B. To ensure a consistent evaluation framework, all models were evaluated in chat mode under zero-shot settings without chain-of-thought prompting.
Three larger models were also evaluated via the Together AI platform: Llama 3.3 70B and Llama 3.1 405B [49], the latter used only during the question control phase. These were evaluated under the same zero-shot chat configuration.

RAG Setup. In a standard RAG configuration, once a model is prompted with a query, a dedicated retrieval module embeds the context and fetches the most relevant information from an external knowledge base. In our experimental setting, this knowledge base comprises the textbooks from which the clinical reasoning paths were extracted. While implementing GraphRAG approaches, which leverage entire knowledge graphs within the retrieval pipeline, was structurally possible, we opted for standard RAG. Our primary objective was to evaluate the models' ability to perform multi-step reasoning by actively constructing these logical paths from unstructured text. Providing the information via GraphRAG would have pre-encoded the structural relationships of the decision pathways, thereby bypassing the exact reasoning capabilities we aimed to assess (note: this would resemble the topline setting). Specifically, the retrieval pipeline was implemented using the mxbai-embed-large model, configured with a chunk size of 500 and a chunk overlap of 150 tokens.

Exact Match. In the quiz evaluation, models were instructed via the system prompt to respond with a single letter (A–E). To handle variations (e.g., "The correct answer is A"), a regular expression extracts the letter from each response. Accuracy was then computed as the proportion of extracted letters matching the ground-truth key.

LLM-as-a-judge Score. Ensuring the reliability and coherence of responses is crucial in medical Q&A. To achieve this, an LLM-based Judge (Gemini-flash 2.0) employing in-context learning was used for systematic, data-driven evaluation. Using the ground-truth answer and reasoning path, the judge scores each response on a 0–10 scale based on predefined criteria. The evaluation leverages the G-Eval metric [93], which operates in two phases: first, the judge is introduced to the task and evaluation criteria to generate assessment steps; second, these steps are combined with the criteria and task description to produce the final score. The same judge was used throughout all evaluations. Implementation details and prompts are provided in Appendix A.3.

Semantic Similarity Score. The models were further evaluated using a semantic similarity metric that compares the ground-truth answer with the open response generated by each model. For this, BGE-M3 model [27] was used, one of the most advanced publicly available text embedding models. BGE-M3 can produce various embedding types (dense, sparse, ColBERT-style) and integrate them effectively, allowing for comparison between semantic scores from dense embeddings and lexical scores from sparse ones. This metric serves two purposes: (i) validating the reliability of the LLM-as-a-judge scores and (ii) providing insight into the strategies models use to answer open-ended Q&A.

5.4 Experiments and Findings

The following sections present the main findings from pipeline development and model evaluation. Our experiments focus on answering the following research questions:

RQ1. How do the different dataset construction steps impact the quality and complexity of the generated medical questions?

RQ2. How do different pretrained LLMs perform on the new benchmark across the established evaluation metrics?

RQ3. Does providing structured contextual knowledge yield better model performance than providing unstructured text?

5.4.1 Dataset Characteristics

After filtering, the final HealthBranches dataset contains 4,063 question–answer pairs across various medical specialties (Table 5.2), including cardiology, neurology, infectious diseases, and hematology/oncology. This diversity ensures that the dataset reflects both common and specialized domains encountered in clinical practice. Categories with a similar number of sub-conditions (e.g., *Hematology/Oncology* vs. *Rheumatology*) differ in question count due to variations in the complexity of their clinical reasoning graphs. More complex domains produce a wider variety of reasoning paths and, consequently, more questions, while simpler categories yield fewer. A useful proxy for this complexity is the number of leaf nodes in the reasoning graph for each category, as each leaf node can contribute up to two sampled reasoning paths (to avoid skewing overrepresented classes). Upsampling of these categories was deliberately omitted to preserve semantic diversity and minimize redundancy.

As highlighted in Table 5.1, *HealthBranches* uniquely combines multiple-choice and open-ended formats with explicit reasoning paths derived from decision graphs. Each question is aligned with a clear, structured reasoning path, similar to a clinical decision tree, that integrates domain-specific knowledge checks with qualitative reasoning steps and interpretable explanations. This design makes *HealthBranches* well-suited for training and evaluating trustworthy, explanation-aware medical LLMs.

5.4.2 Experiments on Dataset Creation

Q&A validation. As discussed in Section 5.3.1, the automatic generation process may introduce flawed or imprecise examples. Figure 5.3 shows accuracy gains after applying

Table 5.2: Distribution of questions and number of leaf nodes across clinical categories in the HealthBranches dataset.

Category	**Questions**	**Conditions**	**Leafs**
Hematology/Oncology	531	22	147
General Medicine	457	15	228
Infectious Diseases	441	15	241
Neurology	373	19	258
Women's Health	350	18	152
Gastrointestinal	338	21	239
Nephrology	279	17	138
Cardiology	278	18	153
Rheumatology	217	22	197
Behavioral Medicine	190	8	79
Endocrinology	183	16	117
Pharmacology	146	8	68
Pulmonary Disease	125	12	97
Emergency Medicine	54	7	36
Urology	45	5	30
Dermatology	35	5	25
Ocular	21	4	22

the refinement protocol, with significant improvements in both zero-shot and RAG settings, particularly for larger models (**RQ1**).

Expert Evaluations. An initial validation of the dataset with practising physicians and senior medical students was conducted, using a structured protocol. Each item, comprising the question, multiple-choice answers, and reasoning, was evaluated through written feedback and a structured scoring system. Specifically, each subcomponent (question formulation, answer accuracy, and reasoning quality) received a score ranging from 0 to 5, yielding a maximum total score of 15 per item. Based on the collected data, a total of 145 reviews were submitted by 22 reviewers, yielding high average scores across all dimensions: 4.68 for question accuracy, 4.68 for answer accuracy, and 4.61 for reasoning path quality. The study results show that physicians gave more conservative ratings, while students scored higher. Only 19 questions received a 3 or below in any category, and just 6 fell below 9/15 overall. These results suggest that the pipeline produces clinically reliable Q&A content, though broader expert review could further enhance consistency. A comprehensive description of the validation protocol, along with further details on the final outcomes, is provided in Appendix A.3.

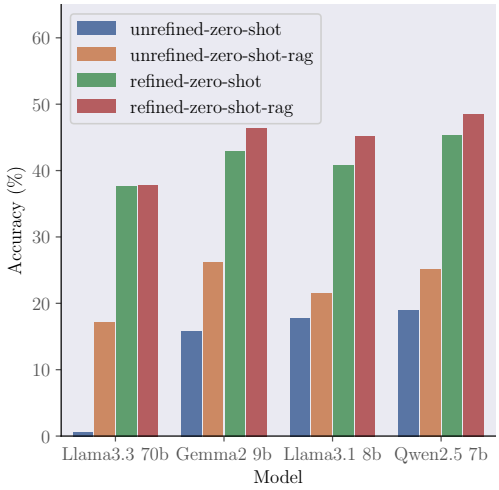

Figure 5.3: Improved accuracy across models after the Q&A refinement process.

Table 5.3: Model performance by modality. Each modality (Zero-shot, RAG, etc.) includes Accuracy, Judge score, and Similarity. Scores in **bold** denote the best-performing model (rank 1), while those underlined indicate the second-best performance (rank 2) for the corresponding metric.

Model	Zero-Shot			Zero-Shot RAG			Topline Path			Topline Text			Topline All		
	Acc	Judge	Sim	Acc	Judge	Sim	Acc	Judge	Sim	Acc	Judge	Sim	Acc	Judge	Sim
Llama2 7b	19.71	5.14	0.314	34.36	4.89	0.317	28.25	8.15	0.385	40.09	4.82	0.320	55.45	6.68	0.360
Mistral 7b	51.93	6.46	0.339	61.14	6.57	0.346	87.69	9.00	0.437	62.86	6.59	0.356	82.40	8.21	0.420
Gemma 7b	57.27	5.24	0.305	62.22	5.73	0.325	89.37	8.67	0.407	61.90	6.08	0.340	80.56	8.23	0.407
Deepseek-r1 8b	61.33	5.46	0.321	63.97	6.02	0.337	88.48	8.41	0.418	60.94	6.16	0.334	76.15	7.43	0.377
Gemma3 4b	62.79	6.14	0.318	62.79	6.20	0.320	89.74	8.91	0.391	61.92	6.06	0.330	79.30	8.18	0.383
Llama3.1 8b	67.22	5.45	0.321	69.87	6.25	0.340	88.01	8.48	0.411	64.73	6.17	0.342	78.05	8.27	0.413
Nemo 12b	67.29	6.00	0.311	68.23	6.49	0.331	92.35	8.69	0.412	70.17	6.67	0.340	88.43	8.32	0.404
Qwen2.5 7b	68.55	6.20	0.335	70.02	6.59	**0.352**	92.84	9.00	**0.456**	71.55	6.73	0.361	89.66	8.55	0.437
Gemma2 9b	69.16	6.36	0.334	71.65	6.53	0.347	92.69	9.07	0.443	71.03	6.89	**0.369**	87.96	8.68	**0.451**
Phi4 14b	73.20	**7.07**	0.340	74.18	**7.32**	0.349	**92.99**	**9.20**	0.419	68.30	**7.28**	0.354	80.04	**8.84**	0.408
Llama3.3 70b	**75.83**	6.53	**0.346**	**75.44**	6.55	0.345	92.67	8.80	0.439	**76.57**	6.91	0.366	**93.45**	8.82	0.446

5.4.3 Performance on the Generated Dataset

Models were evaluated under various topline settings to assess the impact of reasoning paths (**RQ2**). Figure 5.4-top compares the results on the dataset using the quiz modality, while Figure 5.4-middle/bottom does the same for the open-ended answer setting (numerical performances are available in Table 5.3). The findings indicate that models significantly benefit from explicit reasoning paths; however, they do not gain from textual descriptions alone, demonstrating their inability to reliably extract reasoning steps from text (**RQ3**). The introduction of RAG information yielded only marginal improvements over zero-shot performance, particularly for larger models. Notably, smaller models such as Llama2 exhibited greater enhancements, suggesting that contextual information is already embedded in newer, larger models. Results from open-ended questions align with those from the quiz setting,

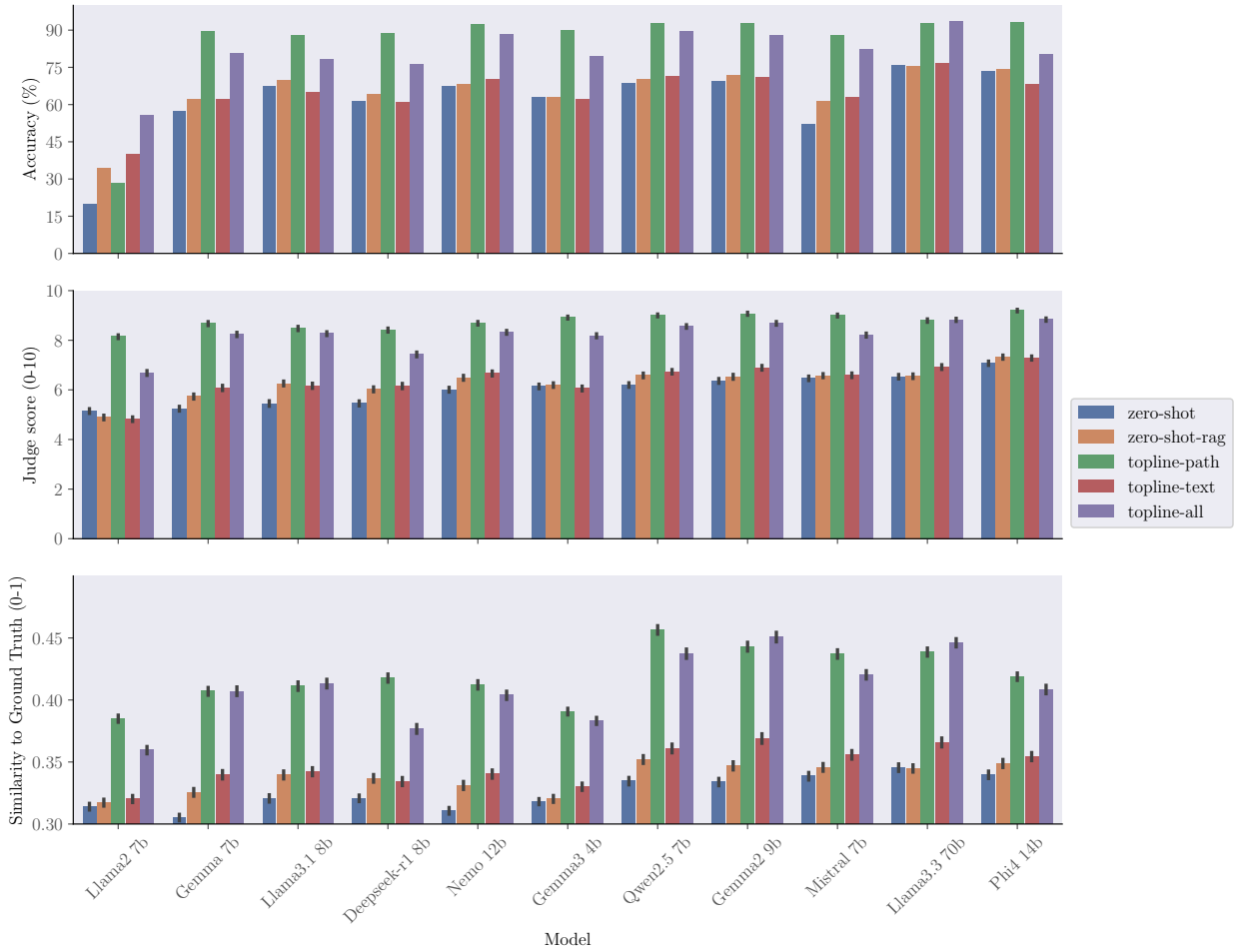

Figure 5.4: Model performance on HealthBranches. Top: quiz accuracy across settings. Middle: G-eval scores for open-ended answers. Bottom: cosine similarity between predictions and ground truth. The bottom plot's y-axis is truncated for visual clarity, as cosine similarity measures relative semantic alignment rather than absolute distance from zero. Error bars show 95% confidence intervals across all plots.

reinforcing the use of LLM-as-a-judge as a dependable quality metric. The semantic similarity metric presented in Figure 5.4, derived from both dense and sparse embeddings with BGE-M3, further substantiates this by incorporating both semantic and lexical similarity.

Comparing Modalities. As observed in Table 5.3, utilizing both "text and path" (Topline All setup) yields performance that is equal to or worse than using the "path" alone (Topline Path). To verify whether this behavior was driven by specific question types, we conducted a per-category breakdown, which is detailed in Appendix A.3. This analysis confirms that the underperformance of the combined modalities is consistent across almost all categories.

While this result is somewhat counterintuitive, as one might assume more context im-

proves performance, it aligns with established findings on how LLMs process extended contexts [90]. The "path" format provides highly structured, logically dense information. In contrast, adding "text" introduces unstructured, verbose data that conveys the same underlying facts but significantly extends the overall context window. Because LLM capabilities are known to degrade as context length increases, often losing track of relevant information, the concise "path only" representation maintains a much higher signal-to-noise ratio, ultimately proving to be the more robust input.

5.5 Review, Challenges & Next Steps

This chapter presented HealthBranches, a framework for extracting structured clinical knowledge from medical literature and leveraging it to evaluate large language models in medical question-answering tasks. The framework addresses two challenges: the lack of methods for converting semi-structured medical guidelines into machine-readable formats and the limitations of evaluation metrics that assess only answer correctness without verifying alignment with clinical reasoning.

Through a semi-automated pipeline, HealthBranches extracts decision pathways from medical textbooks, transforming narrative protocols into graph-structured representations. By systematically sampling paths through these decision graphs, the framework generates synthetic yet clinically grounded patient scenarios without requiring access to real patient data. This approach produces over 4000 question-answer pairs across 17 healthcare domains, each explicitly linked to its underlying reasoning pathway and validated through expert review.

Experimental evaluation across 11 LLMs revealed that models benefit from explicit reasoning paths but struggle to extract implicit reasoning from unstructured text alone. The framework's LLM-as-a-judge evaluation, which utilizes structured reasoning paths to assess open-ended responses, demonstrated strong alignment with human expert assessments, providing a more nuanced measure of clinical competence than accuracy metrics alone.

Limitations. The proposed framework, while demonstrating significant potential, faces several inherent constraints. First, the extraction of structured knowledge from clinical guidelines remains a non-trivial task that requires careful validation to ensure fidelity to the source material. Second, the LLM-as-a-judge evaluation procedure, though practical, may introduce biases into the assessment process. While this could be mitigated through ensemble judging approaches or multiple evaluation passes, such improvements come at increased computational cost. Third, our evaluation is currently based entirely on general-

purpose LLMs rather than domain-specific medical LLMs. While this choice was intentional to establish a baseline and avoid potential data leakage from models already fine-tuned on clinical corpora, it leaves the specific behavior of medical LLMs on our dataset unexplored. Finally, the current implementation uses a limited set of clinical sources, which constrains the generalizability of the extracted reasoning pathways. Expanding the corpus of guidelines would increase the framework's coverage but also amplify the aforementioned extraction challenges.

Impact and Real-World Deployment Translating this framework from benchmark evaluation to utility requires a shift toward a human-in-the-loop paradigm. Future clinical applications cannot treat LLM-generated responses as a static output; instead, they must dynamically adapt to the user's specific medical expertise. Furthermore, this framework holds potential in educational scenarios, where explicit decision pathways can serve as interactive learning tools that model expert clinical reasoning for medical students.

Future Directions. Despite these limitations, the framework opens several promising directions for advancing clinical ML systems. The ability to generate knowledge-grounded synthetic examples from authoritative sources, without requiring access to sensitive patient data, presents immediate opportunities for educational applications and for modeling rare conditions where real-world data is inherently limited.

The emergence of "reasoning" LLMs, which explicitly expose their intermediate thinking processes, presents an opportunity to deepen our understanding of their inner-workings. Evaluating these models on the HealthBranches benchmark could reveal whether their internal reasoning steps align with clinically validated pathways during the "thinking" phase, offering insight into model decision-making processes.

Finally, the extracted structured pathways provide a foundation for targeted model improvement. An immediate next step involves fine-tuning pretrained LLMs on the knowledge-grounded examples generated by the framework. At the same time, these pathways could inform the development of constrained generation techniques or serve as verification mechanisms during model inference.

Chapter 6

Conclusions

6.1 Summary

This thesis addresses a fundamental challenge in the application of artificial intelligence to biomedical and clinical domains: while modern machine learning and deep learning approaches have remarkable statistical power, they often fail to meet the rigorous demands of healthcare applications. Through the work presented in previous chapters, we have explored how this failure stems not from the algorithms themselves, but from a disconnect between data-driven learning and the rich, structured knowledge that defines medical practice.

The limitations of AI in healthcare manifest across three dimensions. Data scarcity presents itself through the "small data, big features" problem, a characteristic of clinical datasets where patient cohorts are inherently limited, labels can be costly to obtain, and feature spaces are vast (Chapter 2). Limited interpretability hinders clinicians' understanding and trust in model decisions, a non-negotiable requirement in high-stakes medical settings where every prediction must be justifiable. Misalignment with clinical practice occurs when models optimize for statistical objectives that diverge from established medical reasoning, guidelines, and domain expertise.

While these challenges are recognized across machine learning, they are particularly prominent in biomedical contexts. A model that achieves high accuracy but reasons in clinically unsound ways, or one that cannot incorporate established pathophysiological relationships, is fundamentally unsuitable for healthcare deployment.

This thesis argues that these limitations can be traced back to a specific gap: the disconnect between the statistical patterns learned from data and the structured knowledge and reasoning processes that characterize the domains of application. Medical practice is not built solely on empirical observation but on accumulated knowledge, formalized in ontologies, encoded in clinical guidelines, and structured in our understanding of biological systems. Yet standard machine learning pipelines remain largely indifferent to this knowledge, treating medical prediction tasks as generic pattern recognition problems.

To bridge this knowledge-reasoning gap, this thesis proposes a framework based on three fundamental operations that connect machine learning pipelines to structured domain knowledge:

Knowledge Extraction addresses the challenge that while clinical and biomedical domains are rich in structured knowledge, often representable as graphs, many critical sources remain in unstructured formats, inaccessible to deep learning models. Chapter 5 demon-

strated that modern large language models can systematically extract knowledge from clinical guidelines and transform it into formal, graph-structured representations. The HealthBranches methodology showed that this process is not merely a reformatting exercise but a principled approach to capture the logical structure of clinical reasoning pathways, creating knowledge artifacts that connect human expertise and machine learning systems.

Knowledge Injection addresses how to leverage structured knowledge within neural architectures. Rather than treating knowledge as external constraints or post-hoc corrections, Chapters 3 and 4 demonstrated methods for embedding domain knowledge directly into model architectures, enabling models to reason with biomedical knowledge. In Chapter 3, we showed how structural positional encodings allow Transformers to incorporate medical ontologies for clinical process monitoring, fundamentally changing how the model processes temporal event sequences. Chapter 4 extended this principle to high-dimensional data, demonstrating how graph representation learning can transform gene ontology structures into learned encodings that guide multi-omics analysis for disease diagnosis.

Knowledge-Based Evaluation acknowledges that assessing model quality requires moving beyond aggregate performance metrics to examine whether models reason in clinically coherent ways. Chapter 5 demonstrated how structured knowledge extracted from clinical guidelines can be repurposed as an evaluation framework. HealthBranches not only formalizes clinical reasoning pathways but also provides a benchmark for assessing whether model predictions align with established clinical logic. This shift from pure predictive accuracy to reasoning alignment represents a new prospective of how we validate machine learning systems for healthcare.

Together, these three components form a complete workflow for knowledge-grounded machine learning. Extraction makes domain knowledge accessible; injection makes it actionable within learning algorithms; evaluation ensures that learned models respect the logical structure of clinical reasoning. Each component addresses a distinct gap, yet they are mutually reinforcing: better extraction enables more effective injection; principled injection creates models more suitable for knowledge-based evaluation; and evaluation insights, in turn, inform what knowledge should be extracted and how it should be injected.

6.2 Implications

In light of the reported work and experiments, the contributions of this thesis extend beyond specific architectures or extraction techniques; they hint at a fundamental shift in the perspective of the machine learning project lifecycle in biomedical and clinical domains.

Traditionally, the workflow for developing clinical ML is data-centric. It begins with data exploration, handling missing values, and analyzing correlations before proceeding to model selection. The framework developed in this thesis advocates for a knowledge-first approach.

Before selecting an architecture, this perspective suggests that practitioners identify the structured knowledge associated with the specific cohort, clinical task, or biological phenomenon being modeled. In particular:

- **What defines the problem?** Is the underlying logic characterized by hierarchical taxonomies (e.g., Gene Ontologies), temporal procedural flows (e.g., Clinical Guidelines), or causal networks?
- **Where does this logic reside?** Is it available in formalized databases, or must it be extracted from unstructured literature?
- **How should it be deployed?** Does this knowledge serve as the architectural scaffold for the model (Injection), or as the ground truth for validating the model's logic (Evaluation)?

Explicit inductive biases. This perspective also offers a clarifying lens on consolidated machine learning methodologies. Techniques such as manual feature engineering, specific data augmentation strategies, or domain-informed feature selection can be seen as manual attempts to inject domain knowledge into a learning system. When a researcher creates a specific feature or selects a specific augmentation, they are introducing an implicit inductive bias based on their understanding of the domain.

The framework presented in this thesis does not discard these methods but rather formalizes the intent behind them. It moves the injection of knowledge from a manual, heuristic, and often opaque process to an explicit, architectural, and systematic process. By making these biases explicit and structurally integrated, we make the model's reliance on domain knowledge transparent, tunable, and verifiable.

Practical Impact

The practical implications of the proposed framework become concrete when considered against the clinical and research scenarios explored in this thesis. In the context of stroke process monitoring (Chapter 3), embedding clinical guidelines as structural positional encodings enables a model that does not merely predict the next activity in a care pathway, but does so in a manner that is aligned with protocol-defined temporal flows, making its outputs more interpretable by clinicians and suitable for real-time audit of guideline adherence. In multi-omics disease diagnosis (Chapter 4), grounding learned representations in Gene Ontology structures produces a knowledge-grounded encoder whose inductive biases reflect established biological relationships. Moreover, the approach directly tackles the problems due to small cohorts of patients by using a model, since the encoder can be reused with only small additional training. Finally, the HealthBranches framework (Chapter 5) addresses a need for the responsible deployment of large language models in clinical settings: by evaluating model outputs against structured clinical reasoning pathways rather than answer correctness alone, it provides a principled mechanism for assessing reasoning alignment, a prerequisite for any system operating in high-stakes medical scenarios.

6.2.1 Future Directions

While this thesis establishes a core framework, several lines of research remain open for future investigation.

End-to-End validation. The experiments presented demonstrated the efficacy of these components in isolation or coupled pairs. However, a compelling avenue for future work is an end-to-end application of this framework. Such a study would be designed with a *knowledge-first* mindset. Starting with the formalization of the relevant medical ontology, followed by the design of a scaffolded architecture, and concluding with reasoning-based validation.

Scaling injection to LLMs and large ontologies. As discussed in the conclusions of individual chapters, technical challenges remain regarding scale. While this work successfully encoded specific sub-graphs, scaling these injection mechanisms to encompass massive ontologies remains an open challenge. Furthermore, exploring methods to inject structured knowledge back into the training dynamics of Large Language Models (LLMs) represents a promising frontier to mitigate hallucination and enforce clinical logic in generative AI.

6.2.2 Limitations

Despite the promise of this framework, it is necessary to acknowledge its constraints, particularly regarding the dependence on the domain knowledge itself.

The knowledge availability bottleneck. The primary limitation of the proposed pipeline is its reliance on high-quality, structured domain knowledge. In frontier areas of medicine, such as rare disease diagnosis or novel drug discovery, the "ground truth" mechanism may be unknown or debated. In such "low-knowledge" scenarios, enforcing a structural scaffold based on incomplete theories could bias the model, blinding it to novel patterns in the data.

Constraint vs. discovery. There is an inherent tension between constraining a model to follow known rules and allowing it to discover new ones. By enforcing alignment with current clinical guidelines, we risk creating systems that are merely as good as current practice, potentially stifling the discovery of new biological markers.

However, this limitation can be reframed as a diagnostic feature. If a data-driven model persistently fails to align with an injected knowledge graph, or if its performance degrades when constrained by "known biology," this discrepancy is a signal. It suggests that the data contains evidence contradicting our formalized understanding. Thus, while the framework may be limited in modeling poorly understood phenomena, it may hold potential as a tool for discrepancy analysis, helping researchers identify exactly where current medical knowledge fails to explain empirical patient data.

Bibliography

[1] Marah Abdin, Jyoti Aneja, Harkirat Behl, Sébastien Bubeck, Ronen Eldan, Suriya Gunasekar, Michael Harrison, Russell J Hewett, Mojan Javaheripi, Piero Kauffmann, et al. Phi-4 technical report. *arXiv preprint arXiv:2412.08905*, 2024.

[2] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[3] Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2019.

[4] M. Z. Alom, T. M. Taha, C. Yakopcic, S. Westberg, P. Sidike, M. S. Nasrin, M. Hasan, B. C. Van Essen, A. A. S. Awwal, and V. K. Asari. A state-of-the-art survey on deep learning theory and architectures. *Electronics*, 8(3), 2019.

[5] Guilherme Aresta, Teresa Araújo, Scotty Kwok, Sai Saketh Chennamsetty, Mohammed Safwan, Varghese Alex, Bahram Marami, Marcel Prastawa, Monica Chan, Michael Donovan, et al. Bach: Grand challenge on breast cancer histology images. *Medical image analysis*, 56:122–139, 2019.

[6] Muhammad Arslan, Hussam Ghanem, Saba Munawar, and Christophe Cruz. A survey on rag with llms. *Procedia Computer Science*, 246:3781–3790, 2024.

[7] Michael Ashburner, Catherine A Ball, Judith A Blake, David Botstein, Heather Butler, J Michael Cherry, Allan P Davis, Kara Dolinski, Selina S Dwight, Janan T Eppig, et al. Gene ontology: tool for the unification of biology. *Nature genetics*, 25(1):25–29, 2000.

[8] Bulut Aygüneş, Selim Aksoy, Ramazan Gökberk Cinbiş, Kemal Kösemehmetoğlu, Sevgen Önder, and Ayşegül Üner. Graph convolutional networks for region of interest

classification in breast histopathology. In *Medical Imaging 2020: Digital Pathology*, volume 11320, pages 134–141. SPIE, 2020.

[9] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

[10] Peter W Battaglia, Jessica B Hamrick, Victor Bapst, Alvaro Sanchez-Gonzalez, Vinicius Zambaldi, Mateusz Malinowski, Andrea Tacchetti, David Raposo, Adam Santoro, Ryan Faulkner, et al. Relational inductive biases, deep learning, and graph networks. *arXiv preprint arXiv:1806.01261*, 2018.

[11] Mikhail Belkin and Partha Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural computation*, 15(6):1373–1396, 2003.

[12] Mattia Bellan, Annalisa Chiocchetti, Marco Dossena, Christopher Irwin, Luca Piovesan, and Luigi Portinale. Predicting long-covid sequelae: A multi-label classification approach. *Intelligenza Artificiale*, page 17248035251317937, 2025.

[13] Alessandro Berti and Wil M. P. van der Aalst. Reviving token-based replay: Increasing speed while improving diagnostics. In Wil M. P. van der Aalst, Robin Bergenthum, and Josep Carmona, editors, *Proceedings of the International Workshop on Algorithms & Theories for the Analysis of Event Data 2019 Satellite event of the conferences: 40th International Conference on Application and Theory of Petri Nets and Concurrency Petri Nets 2019 and 19th International Conference on Application of Concurrency to System Design ACSD 2019, ATAED@Petri Nets/ACSD 2019, Aachen, Germany, June 25, 2019*, volume 2371 of *CEUR Workshop Proceedings*, pages 87–103. CEUR-WS.org, 2019.

[14] Som S Biswas. Role of chat gpt in public health. *Annals of biomedical engineering*, 51 (5):868–869, 2023.

[15] Olivier Bodenreider. The unified medical language system (umls): integrating biomedical terminology. *Nucleic acids research*, 32(suppl_1):D267–D270, 2004.

[16] A. Bottrighi, L. Canensi, G. Leonardi, S. Montani, and P. Terenziani. Trace retrieval for business process operational support. *Expert Syst. Appl.*, 55:212–221, 2016.

[17] Nadia Brancati, Anna Maria Anniciello, Pushpak Pati, Daniel Riccio, Giosuè Scognamiglio, Guillaume Jaume, Giuseppe De Pietro, Maurizio Di Bonito, Antonio Foncubierta, Gerardo Botti, et al. Bracs: A dataset for breast carcinoma subtyping in h&e histology images. *Database*, 2022:baac093, 2022.

[18] D. Breuker, M. Matzner, P. Delfmann, and J. Becker. Comprehensible predictive models for business processes. *MIS Quarterly*, 40:1009–1034, 2016.

[19] Zaharah A. Bukhsh, Aaqib Saeed, and Remco M. Dijkman. Processtransformer: Predictive business process monitoring with transformer network, 2021.

[20] C. Cabanillas, C. DiCiccio, J. Mendling, and A. Baumgrass. Predictive task monitoring for business processes. In S. Wasim Sadiq, P. Soffer, and H. Völzer, editors, *Business Process Management - 12th International Conference, BPM 2014, Haifa, Israel, September 7-11, 2014. Proceedings*, volume 8659 of *Lecture Notes in Computer Science*, pages 424–432. Springer, 2014. doi: 10.1007/978-3-319-10172-9_31. URL https://doi.org/10.1007/978-3-319-10172-9_31.

[21] M. Camargo, M. Dumas, and O. González Rojas. Learning accurate LSTM models of business processes. In T. T. Hildebrandt, B. F. van Dongen, M. Röglinger, and J. Mendling, editors, *Business Process Management - 17th International Conference, BPM 2019, Vienna, Austria, September 1-6, 2019, Proceedings*, volume 11675 of *Lecture Notes in Computer Science*, pages 286–302. Springer, 2019.

[22] Massimo Canonico, Francesco Desimoni, Alberto Ferrero, Pietro Antonio Grassi, Christopher Irwin, Daiana Campani, Alberto Dal Molin, Massimiliano Panella, and Luca Magistrelli. Gait monitoring and analysis: A mathematical approach. *Sensors*, 23(18):7743, 2023.

[23] Nicholas Carlini, Florian Tramer, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Ulfar Erlingsson, et al. Extracting training data from large language models. In *30th USENIX Security Symposium (USENIX Security 21)*, pages 2633–2650, 2021.

[24] Payal Chandak, Kexin Huang, and Marinka Zitnik. Building a knowledge graph to enable precision medicine. *Scientific Data*, 10(1):67, 2023. URL https://doi.org/10.1038/s41597-023-01960-3.

[25] Francisco Charte, Antonio J Rivera, María J Del Jesus, and Francisco Herrera. Mlsmote: Approaching imbalanced multilabel learning through synthetic instance generation. *Knowledge-Based Systems*, 89:385–397, 2015.

[26] Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16:321–357, 2002.

[27] Jianlv Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. Bge m3-embedding: Multi-lingual, multi-functionality, multi-granularity text embeddings through self-knowledge distillation. *arXiv preprint arXiv:2402.03216*, 2024.

[28] Richard J. Chen, Chengkuan Chen, Yicong Li, Tiffany Y. Chen, Andrew D. Trister, Rahul G. Krishnan, and Faisal Mahmood. Scaling vision transformers to gigapixel images via hierarchical self-supervised learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16144–16155, June 2022.

[29] Richard J Chen, Tong Ding, Ming Y Lu, Drew FK Williamson, Guillaume Jaume, Andrew H Song, Bowen Chen, Andrew Zhang, Daniel Shao, Muhammad Shaban, et al. Towards a general-purpose foundation model for computational pathology. *Nature medicine*, 30(3):850–862, 2024.

[30] Yiqun Chen and James Zou. Genept: a simple but effective foundation model for genes and cells built from chatgpt. *bioRxiv*, pages 2023–10, 2024.

[31] S. Chilamkurthy, R. Ghosh, S. Tanamala, M. Biviji M, N.G. Campeau, V.K. Venugopal, V. Mahajan, P. Rao, and P. Warier. Deep learning algorithms for detection of critical findings in head ct scans: a retrospective study. *Lancet*, 392:2388–2396, 2018.

[32] K. Cho, B. van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation, 2014.

[33] Cristian Cosentino, Annamaria Defilippo, Marco Dossena, Christopher Irwin, Sara Joubbi, and Pietro Liò. Healthbranches: Synthesizing clinically-grounded question answering datasets via decision pathways. *arXiv preprint arXiv:2508.07308*, 2025.

[34] M. Daidone, S. Ferrantelli S, and A. Tuttolomondo. Machine learning applications in stroke medicine: advancements, challenges, and future prospectives. *Neural Regen Res*, 19(4):769–773, 2024.

[35] Jacob Devlin. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[36] Chiara Di Francescomarino, Chiara Ghidini, Fabrizio Maria Maggi, Giulio Petrucci, and Anton Yeshchenko. An eye into the future: leveraging a-priori knowledge in predictive business process monitoring. In *Business Process Management: 15th Interna-*

tional Conference, BPM 2017, Barcelona, Spain, September 10–15, 2017, Proceedings 15, pages 252–268. Springer, 2017.

[37] Chiara DiFrancescomarino and Chiara Ghidini. Predictive process monitoring. In Wil M. P. van der Aalst and Josep Carmona, editors, *Process Mining Handbook*, volume 448 of *Lecture Notes in Business Information Processing*, pages 320–346. 2022. doi: 10.1007/978-3-031-08848-3_10.

[38] Ivan Donadello, Jonghyeon Ko, Fabrizio Maria Maggi, Jan Mendling, Francesco Riva, and Matthias Weidlich. Knowledge-driven modulation of neural networks with attention mechanism for next activity prediction, 2023.

[39] Alexey Dosovitskiy. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[40] Marco Dossena, Christopher Irwin, Annalisa Chiocchetti, and Luigi Portinale. Beyond labels: A self-supervised framework with masked autoencoders and random cropping for breast cancer sub-type classification. In *Proceedings of the AAAI Symposium Series*, volume 5, pages 10–17, 2025.

[41] Vijay Prakash Dwivedi and Xavier Bresson. A generalization of transformer networks to graphs. *AAAI Workshop on Deep Learning on Graphs: Methods and Applications*, 2021.

[42] Vijay Prakash Dwivedi, Chaitanya K Joshi, Anh Tuan Luu, Thomas Laurent, Yoshua Bengio, and Xavier Bresson. Benchmarking graph neural networks. *arXiv preprint arXiv:2003.00982*, 2020.

[43] Vijay Prakash Dwivedi, Anh Tuan Luu, Thomas Laurent, Yoshua Bengio, and Xavier Bresson. Graph neural networks with learnable structural and positional representations. *arXiv preprint arXiv:2110.07875*, 2021.

[44] J. Evermann, J.R. Rehse, and P. Fettke. Predicting process behaviour using deep learning. *Decision Support Systems*, 100:129–140, 2017.

[45] Sebastian Farquhar, Jannik Kossen, Lorenz Kuhn, and Yarin Gal. Detecting hallucinations in large language models using semantic entropy. *Nature*, 630(8017):625–630, 2024.

[46] Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, and Haofen Wang. Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2312.10997*, 2023.

[47] Aidan Gilson, Xuguang Ai, Thilaka Arunachalam, Ziyou Chen, Ki Xiong Cheong, Amisha Dave, Cameron Duic, Mercy Kibe, Annette Kaminaka, Minali Prasad, et al. Enhancing large language models with domain-specific retrieval augment generation: A case study on long-form consumer health question answering in ophthalmology. *arXiv preprint arXiv:2409.13902*, 2024.

[48] Alexander Goldfarb-Rumyantzev. *Critical Care Medicine: An Algorithmic Approach E-Book*. Elsevier Health Sciences, 2021.

[49] Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

[50] Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

[51] Aditya Grover and Jure Leskovec. NODE2VEC: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, page 855–864, New York, NY, USA, 2016. Association for Computing Machinery. ISBN 9781450342322. doi: 10.1145/2939672.2939754. URL https://doi.org/10.1145/2939672.2939754.

[52] Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

[53] Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Mingwei Chang. Retrieval augmented language model pre-training. In *International conference on machine learning*, pages 3929–3938. PMLR, 2020.

[54] W.L. Hamilton. *Graph Representation Learning*. Synthesis Lectures on Artificial Intelligence and Machine Learning. Springer, 2020.

[55] James E Harrison, Stefanie Weber, Robert Jakob, and Christopher G Chute. Icd-11: an international classification of diseases for the twenty-first century. *BMC medical informatics and decision making*, 21(Suppl 6):206, 2021.

[56] Mark Harrison and Ala Mohammed. *Algorithms for emergency medicine*. Oxford University Press, 2023.

[57] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16000–16009, 2022.

[58] Xuehai He, Yichen Zhang, Luntian Mou, Eric Xing, and Pengtao Xie. Pathvqa: 30000+ questions for medical visual question answering. *arXiv preprint arXiv:2003.10286*, 2020.

[59] Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*, 2020.

[60] Ricardo Hernández Medina, Svetlana Kutuzova, Knud Nor Nielsen, Joachim Johansen, Lars Hestbjerg Hansen, Mads Nielsen, and Simon Rasmussen. Machine learning and deep learning applications in microbiome research. *ISME Communications*, 2(1):98, 10 2022. ISSN 2730-6151. doi: 10.1038/s43705-022-00182-9. URL https://doi.org/10.1038/s43705-022-00182-9.

[61] M. Hinkka, T. Lehto, K. Heljanko, and A. Jung. Classifying process instances using recurrent neural networks. In F. Daniel, Q. Z. Sheng, and H. Motahari, editors, *Business Process Management Workshops - BPM 2018 International Workshops, Sydney, NSW, Australia, September 9-14, 2018, Revised Papers*, volume 342 of *Lecture Notes in Business Information Processing*, pages 313–324. Springer, 2018.

[62] G. E. Hinton and R. R. Salakhutdinov. Reducing the dimensionality ofdata with neural networks. *Science*, 313:504–507, 2006.

[63] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Computation*, 9 (8):1735–1780, 1997.

[64] Siqing Huo, Negar Arabzadeh, and Charles LA Clarke. Retrieving supporting evidence for llms generated answers. *arXiv preprint arXiv:2306.13781*, 2023.

[65] Maximilian Ilse, Jakub Tomczak, and Max Welling. Attention-based deep multiple instance learning. In *International conference on machine learning*, pages 2127–2136. PMLR, 2018.

[66] SNOMED International. Snomed international – snomed ct. https://www.snomed.org/, 2025. Accessed: 2025-11-17.

[67] Christopher Irwin, Marco Dossena, Giorgio Leonardi, and Stefania Montani. Structural positional encoding for knowledge integration in transformer-based medical process monitoring and trace classification. *Progress in Artificial Intelligence*, pages 1–13, 2024.

[68] Italian-Stroke-Association. Current guidelines, 2023. URL https://isa-aii.com/linee-guida/linee-guida-attuali/.

[69] Gautier Izacard, Patrick Lewis, Maria Lomeli, Lucas Hosseini, Fabio Petroni, Timo Schick, Jane Dwivedi-Yu, Armand Joulin, Sebastian Riedel, and Edouard Grave. Atlas: Few-shot learning with retrieval augmented language models. *Journal of Machine Learning Research*, 24(251):1–43, 2023.

[70] Saahil Jain, Ashwin Agrawal, Adriel Saporta, Steven QH Truong, Du Nguyen Duong, Tan Bui, Pierre Chambon, Yuhao Zhang, Matthew P Lungren, Andrew Y Ng, et al. Radgraph: Extracting clinical entities and relations from radiology reports. *arXiv preprint arXiv:2106.14463*, 2021.

[71] Shaoxiong Ji, Shirui Pan, Erik Cambria, Pekka Marttinen, and Philip S Yu. A survey on knowledge graphs: Representation, acquisition, and applications. *IEEE transactions on neural networks and learning systems*, 33(2):494–514, 2021.

[72] Mingyi Jia, Junwen Duan, Yan Song, and Jianxin Wang. medikal: Integrating knowledge graphs as assistants of llms for enhanced clinical diagnosis on emrs. *arXiv preprint arXiv:2406.14326*, 2024.

[73] Fengqing Jiang. Identifying and mitigating vulnerabilities in llm-integrated applications. Master's thesis, University of Washington, 2024.

[74] Yifan Jiang, Matthew Aton, Qiyun Zhu, and Yang Young Lu. Miostone: Modeling microbiome-trait associations with taxonomy-adaptive neural networks. *bioRxiv*, 2024. doi: 10.1101/2023.11.04.565596. URL https://www.biorxiv.org/content/early/2024/02/04/2023.11.04.565596.

[75] Di Jin, Eileen Pan, Nassim Oufattole, Wei-Hung Weng, Hanyi Fang, and Peter Szolovits. What disease does this patient have? a large-scale open domain question answering dataset from medical exams. *Applied Sciences*, 11(14):6421, 2021.

[76] Mingyu Jin, Qinkai Yu, Dong Shu, Chong Zhang, Lizhou Fan, Wenyue Hua, Suiyuan Zhu, Yanda Meng, Zhenting Wang, Mengnan Du, et al. Health-llm: Personalized retrieval-augmented disease prediction system. *arXiv preprint arXiv:2402.00746*, 2024.

[77] Qiao Jin, Bhuwan Dhingra, Zhengping Liu, William Cohen, and Xinghua Lu. Pubmedqa: A dataset for biomedical research question answering. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2567–2577, 2019.

[78] M. A. Khan, H. Le, K. Do, T. Tran, A. Ghose, K. H. Dam, and R. Sindhgatta. Memory-augmented neural networks for predictive process analytics. *CoRR*, abs/1802.00938, 2018. URL http://arxiv.org/abs/1802.00938.

[79] Nikhil Khandekar, Qiao Jin, Guangzhi Xiong, Soren Dunn, Serina Applebaum, Zain Anwar, Maame Sarfo-Gyamfi, Conrad Safranek, Abid Anwar, Andrew Zhang, et al. Medcalc-bench: Evaluating large language models for medical calculations. *Advances in Neural Information Processing Systems*, 37:84730–84745, 2024.

[80] Ksenia Kharitonova, David Pérez-Fernández, Javier Gutiérrez-Hernando, Asier Gutiérrez-Fandiño, Zoraida Callejas, and David Griol. Leveraging retrieval-augmented generation for reliable medical question answering using large language models. In *International Conference on Hybrid Artificial Intelligence Systems*, pages 141–153. Springer, 2024.

[81] Kenji Kira and Larry A Rendell. A practical approach to feature selection. In *Machine learning proceedings 1992*, pages 249–256. Elsevier, 1992.

[82] G.T. Lakshmanan, D. Shamsi, Y. N. Doganata, M. Unuvar, and R. Khalaf. Markov prediction model for data-driven semi-structured business processes. *Knowl. Inf. Syst.*, 42 (1):97–126, 2015.

[83] G. Lanza, C. Setacci, S. Ricci, P. Castelli, A. Cremonesi, J. Lanza, C. Novali, C. Pratesi, P. Santalucia, F. Speziale, A. Zaninelli, and G.F. Gensini. An update of the italian stroke organization-stroke prevention awareness diffusion group guidelines on carotid endarterectomy and stenting: A personalized medicine approach. *International journal of stroke : official journal of the International Stroke Society*, 12(5):560–567, 2017. doi: https://doi.org/10.1177/1747493017694395.

[84] M. Le, B. Gabrys, and D. Nauck. A hybrid model for business process event prediction. In M. Bramer and M. Petridis, editors, *Research and Development in Intelligent Systems XXIX, Incorporating Applications and Innovations in Intelligent Systems XX:*

Proceedings of AI-2012, The Thirty-second SGAI International Conference on Innovative Techniques and Applications of Artificial Intelligence, Cambridge, England, UK, December 11-13, 2012, pages 179–192. Springer, 2012.

[85] G. Leonardi, M. Striani, S. Quaglini, A. Cavallini, and S. Montani. Leveraging semantic labels for multi-level abstraction in medical process mining and trace comparison. *J. Biomed. Informatics*, 83:10–24, 2018.

[86] G. Leonardi, S. Montani, and M. Striani. Explainable process trace classification: An application to stroke. *J. Biomed. Informatics*, 126:103981, 2022.

[87] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474, 2020.

[88] Bin Li, Yin Li, and Kevin W Eliceiri. Dual-stream multiple instance learning network for whole slide image classification with self-supervised contrastive learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 14318–14328, 2021.

[89] Tiancheng Lin, Zhimiao Yu, Hongyu Hu, Yi Xu, and Chang-Wen Chen. Interventional bag multi-instance learning on whole-slide pathological images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19830–19839, 2023.

[90] Nelson F Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. Lost in the middle: How language models use long contexts. *Transactions of the association for computational linguistics*, 12:157–173, 2024.

[91] Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Computing Surveys*, 55(9):1–35, 2023.

[92] Weijie Liu, Peng Zhou, Zhe Zhao, Zhiruo Wang, Qi Ju, Haotang Deng, and Ping Wang. K-bert: Enabling language representation with knowledge graph. page 2901 – 2908, 2020. Cited by: 308.

[93] Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. G-eval: NLG evaluation using gpt-4 with better human alignment. In Houda Bouamor,

Juan Pino, and Kalika Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 2511–2522, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.153. URL https://aclanthology.org/2023.emnlp-main.153/.

[94] Jason Lloyd-Price, Cesar Arze, Ashwin N Ananthakrishnan, Melanie Schirmer, Julian Avila-Pacheco, Tiffany W Poon, Elizabeth Andrews, Nadim J Ajami, Kevin S Bonham, Colin J Brislawn, et al. Multi-omics of the gut microbial ecosystem in inflammatory bowel diseases. *Nature*, 569(7758):655–662, 2019.

[95] Ming Y Lu, Drew FK Williamson, Tiffany Y Chen, Richard J Chen, Matteo Barbieri, and Faisal Mahmood. Data-efficient and weakly supervised computational pathology on whole-slide images. *Nature biomedical engineering*, 5(6):555–570, 2021.

[96] Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. *Advances in neural information processing systems*, 30, 2017.

[97] Yuan Luo. Evaluating the state of the art in missing data imputation for clinical data. *Briefings in Bioinformatics*, 23(1):bbab489, 12 2021. ISSN 1477-4054. doi: 10.1093/bib/bbab489. URL https://doi.org/10.1093/bib/bbab489.

[98] F.M. Maggi, C. DiFrancescomarino, M. Dumas, and C. Ghidini. Predictive monitoring of business processes. In M. Jarke, J. Mylopoulos, C. Quix, C. Rolland, Y. Manolopoulos, H. Mouratidis, and J. Horkoff, editors, *Advanced Information Systems Engineering - 26th International Conference, CAiSE 2014, Thessaloniki, Greece, June 16-20, 2014. Proceedings*, volume 8484 of *Lecture Notes in Computer Science*, pages 457–472. Springer, 2014. doi: 10.1007/978-3-319-07881-6_31. URL https://doi.org/10.1007/978-3-319-07881-6_31.

[99] Andrei Margeloiu, Nikola Simidjievski, Pietro Lio, and Mateja Jamnik. Gcondnet: A novel method for improving neural networks on small high-dimensional tabular data. *arXiv preprint arXiv:2211.06302*, 2022.

[100] Lourdes Mateu, Cristian Tebe, Cora Loste, José Ramón Santos, Gemma Lladós, Cristina López, Sergio España-Cueto, Ruth Toledo, Marta Font, Anna Chamorro, et al. Determinants of the onset and prognosis of the post-covid-19 condition: a 2-year prospective observational cohort study. *The Lancet Regional Health–Europe*, 33, 2023.

[101] N. Di Mauro, A. Appice, and T. M. A. Basile. Activity prediction of business process instances with inception CNN models. In M. Alviano, G. Greco, and F. Scarcello,

editors, *AI*IA 2019 - Advances in Artificial Intelligence - XVIIIth International Conference of the Italian Association for Artificial Intelligence, Rende, Italy, November 19-22, 2019, Proceedings*, volume 11946 of *Lecture Notes in Computer Science*, pages 348–361. Springer, 2019.

[102] N. Mehdiyev, J. Evermann, and P. Fettke. A multi-stage deep learning approach for business process event prediction. In P. Loucopoulos, Y. Manolopoulos, O. Pastor, B. Theodoulidis, and J. Zdravkovic, editors, *19th IEEE Conference on Business Informatics, CBI 2017, Thessaloniki, Greece, July 24-27, 2017, Volume 1: Conference Papers*, pages 119–128. IEEE Computer Society, 2017.

[103] Tao Meng, Ninareh Mehrabi, Palash Goyal, Anil Ramakrishna, Aram Galstyan, Richard Zemel, Kai-Wei Chang, Rahul Gupta, and Charith Peris. Attribute controlled fine-tuning for large language models: A case study on detoxification. *arXiv preprint arXiv:2410.05559*, 2024.

[104] Grégoire Mialon, Dexiong Chen, Margot Selosse, and Julien Mairal. Graphit: Encoding graph structure in transformers. *arXiv preprint arXiv:2106.05667*, 2021.

[105] Stuart B Mushlin and Harry L Greene. *Decision making in medicine: an algorithmic approach*. Elsevier Health Sciences, 2009.

[106] Humza Naveed, Asad Ullah Khan, Shi Qiu, Muhammad Saqib, Saeed Anwar, Muhammad Usman, Naveed Akhtar, Nick Barnes, and Ajmal Mian. A comprehensive overview of large language models. *arXiv preprint arXiv:2307.06435*, 2023.

[107] Shirui Pan, Linhao Luo, Yufei Wang, Chen Chen, Jiapu Wang, and Xindong Wu. Unifying large language models and knowledge graphs: A roadmap. *arXiv preprint arXiv:2306.08302*, 2023.

[108] Dimitrios P Panagoulias, Maria Virvou, and George A Tsihrintzis. Augmenting large language models with rules for enhanced domain-specific interactions: The case of medical diagnosis. *Electronics*, 13(2):320, 2024.

[109] Junwoo Park, Youngwoo Cho, Haneol Lee, Jaegul Choo, and Edward Choi. Knowledge graph-based question answering with electronic health records. In *Machine Learning for Healthcare Conference*, pages 36–53. PMLR, 2021.

[110] R. Pascanu, Ç. Gülçehre, K. Cho, and Y. Bengio. How to construct deep recurrent neural networks. In Y. Bengio and Y. LeCun, editors, *2nd International Conference*

on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings, 2014.

[111] Pushpak Pati, Guillaume Jaume, Lauren Alisha Fernandes, Antonio Foncubierta-Rodríguez, Florinda Feroce, Anna Maria Anniciello, Giosue Scognamiglio, Nadia Brancati, Daniel Riccio, Maurizio Di Bonito, et al. Hact-net: A hierarchical cell-to-tissue graph neural network for histopathological image classification. In *Uncertainty for Safe Utilization of Machine Learning in Medical Imaging, and Graphs in Biomedical Image Analysis: Second International Workshop, UNSURE 2020, and Third International Workshop, GRAIL 2020, Held in Conjunction with MICCAI 2020, Lima, Peru, October 8, 2020, Proceedings 2*, pages 208–219. Springer, 2020.

[112] P. Philipp, R. Jacob, S. Robert, and J. Beyerer. Predictive analysis of business processes using neural networks with attention mechanism. In *2020 International Conference on Artificial Intelligence in Information and Communication, ICAIIC 2020, Fukuoka, Japan, February 19-21, 2020*, pages 225–230. IEEE, 2020.

[113] Tyler Thomas Procko and Omar Ochoa. Graph retrieval-augmented generation for large language models: A survey. In *2024 Conference on AI, Science, Engineering, and Technology (AIxSET)*, pages 166–169. IEEE, 2024.

[114] Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36, 2024.

[115] M. Reichert and B. Weber. *Enabling Flexibility in Process-Aware Information Systems - Challenges, Methods, Technologies*. 2012. ISBN 978-3-642-30408-8. doi: 10.1007/978-3-642-30409-5. URL https://doi.org/10.1007/978-3-642-30409-5.

[116] Derek Reiman, Ahmed A. Metwally, Jun Sun, and Yang Dai. Popphy-cnn: A phylogenetic tree embedded architecture for convolutional neural networks to predict host phenotype from metagenomic data. *IEEE Journal of Biomedical and Health Informatics*, 24(10):2993–3001, 2020. doi: 10.1109/JBHI.2020.2993761.

[117] E.B Ringelstein, A. Chamorro, M. Kaste, P. Langhorne, D. Leys, P. Lyrer, V. Thijs, L. Thomassen, and D. Toni. European stroke organisation recommendations to establish a stroke unit and stroke center. *Stroke*, 44(3):828–840, 2013.

[118] Camilo Ruiz, Hongyu Ren, Kexin Huang, and Jure Leskovec. Enabling tabular

deep learning when $d >> n$ with an auxiliary knowledge graph. *arXiv preprint arXiv:2306.04766*, 2023.

[119] P. Sale, G. Ferriero, L. Ciabattoni, A.M. Cortese, F. Ferracuti, L. Romeo, F. Piccione, and S. Masiero. Predicting motor and cognitive improvement through machine learning algorithm in human subject that underwent a rehabilitation treatment in the early stage of stroke. *J Stroke Cerebrovasc Dis*, 27:2962–2972, 2018.

[120] D. Scrutinio, C. Ricciardi, L. Donisi, E. Losavio, P. Battista, P. Guida, M. Cesarelli, G. Pagano, and G. Daddio. Machine learning to predict mortality after rehabilitation among patients with severe stroke. *Sci Rep*, 10:20127, 2020.

[121] Junyuan Shang, Tengfei Ma, Cao Xiao, and Jimeng Sun. Pre-training of graph augmented transformers for medication recommendation. *CoRR*, abs/1906.00346, 2019. URL http://arxiv.org/abs/1906.00346.

[122] Zhuchen Shao, Hao Bian, Yang Chen, Yifeng Wang, Jian Zhang, Xiangyang Ji, et al. Transmil: Transformer based correlated multiple instance learning for whole slide image classification. *Advances in neural information processing systems*, 34:2136–2147, 2021.

[123] Divya Sharma, Andrew D Paterson, and Wei Xu. TaxoNN: ensemble of neural networks on stratified microbiome data for disease prediction. *Bioinformatics*, 36(17):4544–4550, 05 2020. ISSN 1367-4803. doi: 10.1093/bioinformatics/btaa542. URL https://doi.org/10.1093/bioinformatics/btaa542.

[124] Xiaoming Shi, Zeming Liu, Li Du, Yuxuan Wang, Hongru Wang, Yuhang Guo, Tong Ruan, Jie Xu, and Shaoting Zhang. Medical dialogue: A survey of categories, methods, evaluation and challenges. *arXiv preprint arXiv:2405.10630*, 2024.

[125] Julio Silva, Takehiro Takahashi, Jamie Wood, Peiwen Lu, Alexandra Tabachnikova, Jeff R Gehlhausen, Kerrie Greene, Bornali Bhattacharjee, Valter Silva Monteiro, Carolina Lucas, et al. Sex differences in symptomatology and immune profiles of long covid. *MedRxiv*, pages 2024–02, 2024.

[126] Sithursan Sivasubramaniam, Cedric E Osei-Akoto, Yi Zhang, Kurt Stockinger, and Jonathan Fürst. Sm3-text-to-query: Synthetic multi-model medical text-to-query benchmark. *Advances in Neural Information Processing Systems*, 37:88627–88663, 2024.

[127] Thomas Stegmüller, Behzad Bozorgtabar, Antoine Spahr, and Jean-Philippe Thiran. Scorenet: Learning non-uniform attention and augmentation for transformer-based histopathological image classification. In *Proceedings of the IEEE/CVF winter Conference on applications of computer vision*, pages 6170–6179, 2023.

[128] Huqun Suri, Qi Zhang, Wenhua Huo, Yan Liu, and Chunsheng Guan. Mediaqa: A question answering dataset on medical dialogues. *arXiv preprint arXiv:2108.08074*, 2021.

[129] Wenhao Tang, Sheng Huang, Xiaoxian Zhang, Fengtao Zhou, Yi Zhang, and Bo Liu. Multiple instance learning framework with masked hard instance mining for whole slide image classification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4078–4087, 2023.

[130] N. Tax, I. Teinemaa, and S. J. van Zelst. An interdisciplinary comparison of sequence modeling methods for next-element prediction. *CoRR*, abs/1811.00062, 2018.

[131] Niek Tax, Ilya Verenich, Marcello La Rosa, and Marlon Dumas. Predictive business process monitoring with LSTM neural networks. In *Proceedings of the 29th International Conference on Advanced Information Systems Engineering (CAiSE)*, pages 477–492. Springer, 2017.

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

[133] Gemma Team, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, et al. Gemma: Open models based on gemini research and technology. *arXiv preprint arXiv:2403.08295*, 2024.

[134] Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, et al. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*, 2024.

[135] Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, et al. Gemma 3 technical report. *arXiv preprint arXiv:2503.19786*, 2025.

[136] I. Teinemaa, M. Dumas, M. LaRosa, and F.M. Maggi. Outcome-oriented predictive process monitoring: Review and benchmark. *ACM Trans. Knowl. Discov. Data*, 13(2): 17:1–17:57, 2019. doi: 10.1145/3301300. URL https://doi.org/10.1145/3301300.

[137] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

[138] W. van der Aalst. *Process Mining. Data Science in Action*. 2016.

[139] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need, 2017.

[140] David Vilares and Carlos Gómez-Rodríguez. Head-qa: A healthcare dataset for complex reasoning. *arXiv preprint arXiv:1906.04701*, 2019.

[141] Luigi Vimercati, Luigi De Maria, Marco Quarato, Antonio Caputi, Loreto Gesualdo, Giovanni Migliore, Domenica Cavone, Stefania Sponselli, Antonella Pipoli, Francesco Inchingolo, et al. Association between long covid and overweight/obesity. *Journal of Clinical Medicine*, 10(18):4143, 2021.

[142] Laura Von Rueden, Sebastian Mayer, Katharina Beckh, Bogdan Georgiev, Sven Giesselbach, Raoul Heese, Birgit Kirsch, Julius Pfrommer, Annika Pick, Rajkumar Ramamurthy, et al. Informed machine learning–a taxonomy and survey of integrating prior knowledge into learning systems. *IEEE Transactions on Knowledge and Data Engineering*, 35(1):614–633, 2021.

[143] Xiyue Wang, Sen Yang, Jun Zhang, Minghui Wang, Jing Zhang, Junzhou Huang, Wei Yang, and Xiao Han. Transpath: Transformer-based self-supervised learning for histopathological image classification. In *Medical Image Computing and Computer Assisted Intervention–MICCAI 2021: 24th International Conference, Strasbourg, France, September 27–October 1, 2021, Proceedings, Part VIII 24*, pages 186–195. Springer, 2021.

[144] Xuezhi Wang and Denny Zhou. Chain-of-thought reasoning without prompting. *Advances in Neural Information Processing Systems*, 37:66383–66409, 2024.

[145] Ziyu Wang, Hao Li, Di Huang, and Amir M Rahmani. Healthq: Unveiling questioning capabilities of llm chains in healthcare conversations. *arXiv preprint arXiv:2409.19487*, 2024.

[146] Jael Sanyanda Wekesa and Michael Kimwele. A review of multi-omics data integration through deep learning approaches for disease diagnosis, prognosis, and treatment. *Frontiers in Genetics*, 14, 2023. ISSN 1664-8021. doi: 10.3389/fgene.2023.1199087. URL https://www.frontiersin.org/journals/genetics/articles/10.3389/fgene.2023.1199087.

[147] Junde Wu, Jiayuan Zhu, Yunli Qi, Jingkun Chen, Min Xu, Filippo Menolascina, and Vicente Grau. Medical graph rag: Towards safe medical large language model via graph retrieval-augmented generation. *arXiv preprint arXiv:2408.04187*, 2024.

[148] Guangzhi Xiong, Qiao Jin, Zhiyong Lu, and Aidong Zhang. Benchmarking retrieval-augmented generation for medicine. *arXiv preprint arXiv:2402.13178*, 2024.

[149] Ruibin Xiong, Yunchang Yang, Di He, Kai Zheng, Shuxin Zheng, Chen Xing, Huishuai Zhang, Yanyan Lan, Liwei Wang, and Tieyan Liu. On layer normalization in the transformer architecture. In *International Conference on Machine Learning*, pages 10524–10533. PMLR, 2020.

[150] H. Xu, J. Pang, X. Yang, M. Li, and D. Zhao. Using predictive process monitoring to assist thrombolytic therapy decision-making for ischemic stroke patients. *BMC Medical Informatics Decis. Mak.*, 20-S(3):120, 2020. doi: 10.1186/s12911-020-1111-6. URL https://doi.org/10.1186/s12911-020-1111-6.

[151] An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, et al. Qwen2. 5 technical report. *arXiv preprint arXiv:2412.15115*, 2024.

[152] Wen-wai Yim, Yujuan Fu, Asma Ben Abacha, Neal Snider, Thomas Lin, and Meliha Yetisgen. Aci-bench: a novel ambient clinical intelligence dataset for benchmarking automatic visit note generation. *Scientific data*, 10(1):586, 2023.

[153] Gian Maria Zaccaria, Simone Ferrero, Eva Hoster, Roberto Passera, Andrea Evangelista, Elisa Genuardi, Daniela Drandi, Marco Ghislieri, Daniela Barbero, Ilaria Del Giudice, et al. A clinical prognostic model based on machine learning from the fondazione italiana linfomi (fil) mcl0208 phase iii trial. *Cancers*, 14(1):188, 2021.

[154] Hongrun Zhang, Yanda Meng, Yitian Zhao, Yihong Qiao, Xiaoyun Yang, Sarah E Coupland, and Yalin Zheng. Dtfd-mil: Double-tier feature distillation multiple instance learning for histopathology whole slide image classification. In *Proceedings of*

the IEEE/CVF conference on computer vision and pattern recognition, pages 18802–18812, 2022.

[155] Kaiyan Zhang, Sihang Zeng, Ermo Hua, Ning Ding, Zhang-Ren Chen, Zhiyuan Ma, Haoxin Li, Ganqu Cui, Biqing Qi, Xuekai Zhu, et al. Ultramedical: Building specialized generalists in biomedicine. *Advances in Neural Information Processing Systems*, 37: 26045–26081, 2024.

[156] Yunlong Zhang, Honglin Li, Yunxuan Sun, Sunyi Zheng, Chenglu Zhu, and Lin Yang. Attention-challenging multiple instance learning for whole slide image classification. In *18th European Conference on Computer Vision*, pages 125–143. Springer, 2024.

[157] J Zheng, Q Sun, J Zhang, and SC Ng. The role of gut microbiome in inflammatory bowel disease diagnosis and prognosis. *United European Gastroenterol Journal*, 10(10): 1091–1102, 2022. URL https://pmc.ncbi.nlm.nih.gov/articles/PMC9752296/.

[158] Yanning Zhou, Simon Graham, Navid Alemi Koohbanani, Muhammad Shaban, Pheng-Ann Heng, and Nasir Rajpoot. Cgc-net: Cell graph convolutional network for grading of colorectal cancer histology images. In *Proceedings of the IEEE/CVF international conference on computer vision workshops*, 2019.

[159] Yuxin Zuo, Shang Qu, Yifei Li, Zhangren Chen, Xuekai Zhu, Ermo Hua, Kaiyan Zhang, Ning Ding, and Bowen Zhou. Medxpertqa: Benchmarking expert-level medical reasoning and understanding. *arXiv preprint arXiv:2501.18362*, 2025.

Appendices

Appendix A

Additional Materials

A.1 Case Study 1

Impact of MLSmote on label distribution

Dataset	n	#ls	#pl	Card	Dens	MeanIR	CVIR	SCUMBLE
orig	324	36	54	0.670	0.096	2.303	0.417	0.012
k3I	424	36	54	0.592	0.085	2.173	0.415	0.009
k3R	424	38	79	0.814	0.116	1.762	0.373	0.007
k3U	424	48	154	1.752	0.250	1.566	0.239	0.008
k5I	424	36	54	0.512	0.073	2.303	0.417	0.009
k5R	424	36	54	0.599	0.086	2.089	0.466	0.008
k5U	424	48	154	1.745	0.249	1.602	0.311	0.009

Table A.1: Main features of considered datasets; n: number of instances, #ls: cardinality of label-set, #pl: number of label-set with more than one label

Performance on classification task

In the following, some additional results and figures relative to the Long-COVID case study are reported. In particular, Tables A.2, A.3, A.4 and A.5 report the performance on the different augmented datasets and metrics.

Interpretability Analysis

This section reports the numerical results of the interpretability analysis. In particular, Figure A.1 reports the 20 most important features given by the SHAP values calculated on the Random Forest model trained on the original dataset. Figure A.2 instead compares

the most important features of different models trained on an augmented variation of the original dataset.

Datasets	BCC(C)	BCC(I)	BR	CC	CDN	LC	PS	RAkEL
orig	0.604	**0.605**	0.420	0.604	0.415	0.604	**0.605**	0.604
k3I	0.646	0.647	0.507	0.649	0.503	0.653	**0.673**	0.651
k3R	0.615	0.615	0.546	0.613	0.527	**0.660**	0.643	0.652
k3U	0.637	0.639	0.577	0.638	0.537	0.637	0.624	**0.651**
k5I	0.666	0.666	0.488	0.669	0.466	0.646	**0.680**	0.641
k5R	0.645	0.648	0.499	0.645	0.496	0.648	**0.668**	0.647
k5U	0.651	0.649	0.582	0.649	0.552	0.648	0.637	**0.666**
Relief	0.556	0.558	0.471	0.559	0.425	0.552	**0.566**	0.539
PCA	**0.606**	**0.606**	0.445	**0.606**	0.432	0.605	0.605	0.603
Avg	0.625	0.626	0.504	0.626	0.484	0.628	**0.633**	0.628

Table A.2: Accuracy (Jaccard Index)

Datasets	BCC(I)	BCC(C)	BR	CC	CDN	LC	PS	RAkEL
orig	0.601	0.603	0.383	0.601	0.377	0.604	**0.605**	0.601
k3I	0.641	0.643	0.475	0.645	0.471	0.644	**0.668**	0.642
k3R	0.590	0.593	0.479	0.590	0.461	0.620	**0.623**	0.614
k3U	0.559	0.564	0.460	0.568	0.431	0.589	**0.594**	0.586
k5I	0.666	0.666	0.473	0.669	0.449	0.646	**0.680**	0.641
k5R	0.641	0.645	0.472	0.640	0.468	0.639	**0.664**	0.640
k5U	0.571	0.571	0.472	0.580	0.435	0.586	**0.594**	0.593
Relief	0.546	0.548	0.435	0.548	0.388	0.544	**0.561**	0.523
PCA	**0.605**	**0.605**	0.406	**0.605**	0.391	**0.605**	**0.605**	0.603
Avg	0.602	0.604	0.451	0.605	0.430	0.609	**0.622**	0.605

Table A.3: Exact Match

Datasets	BCC(I)	BCC(C)	BR	CC	CDN	LC	PS	RAkEL
orig	**0.904**	**0.904**	0.847	0.903	0.844	**0.904**	**0.904**	**0.904**
k3I	0.915	0.915	0.874	0.915	0.875	0.911	**0.922**	0.913
k3R	0.903	0.903	0.874	0.903	0.871	0.908	**0.910**	0.908
k3U	0.873	0.876	0.850	0.876	0.839	0.859	0.856	**0.876**
k5I	0.918	0.918	0.868	0.918	0.869	0.908	**0.922**	0.909
k5R	0.913	0.914	0.872	0.913	0.874	0.909	**0.922**	0.911
k5U	0.881	0.880	0.859	0.882	0.842	0.861	0.858	**0.883**
Relief	0.885	0.886	0.847	0.885	0.846	0.882	**0.891**	0.877
PCA	**0.905**	**0.905**	0.851	**0.905**	0.847	0.904	0.904	0.904
Avg	**0.900**	**0.900**	0.860	**0.900**	0.856	0.899	0.898	0.900

Table A.4: Hamming Score

Datasets	BCC(C)	BCC(I)	BR	CC	CDN	LC	PS	RAkEL
orig	0.504	0.504	0.543	0.503	**0.547**	0.501	0.500	0.501
k3I	0.535	0.537	0.574	0.533	**0.579**	0.563	0.560	0.550
k3R	0.621	0.618	0.727	0.622	**0.731**	0.673	0.639	0.666
k3U	0.763	0.765	**0.847**	0.768	0.836	0.746	0.724	0.778
k5I	0.496	0.496	0.452	0.496	0.459	0.490	**0.497**	0.491
k5R	0.529	0.531	0.544	0.532	**0.560**	0.553	0.551	0.550
k5U	0.785	0.784	**0.870**	0.787	0.846	0.760	0.737	0.797
Relief	0.503	0.504	0.518	0.504	**0.547**	0.495	0.495	0.502
PCA	0.501	0.501	0.588	0.501	**0.589**	0.500	0.500	0.500
Avg	0.582	0.582	0.629	0.583	**0.636**	0.587	0.578	0.593

Table A.5: Area under ROC (macro-averaged)

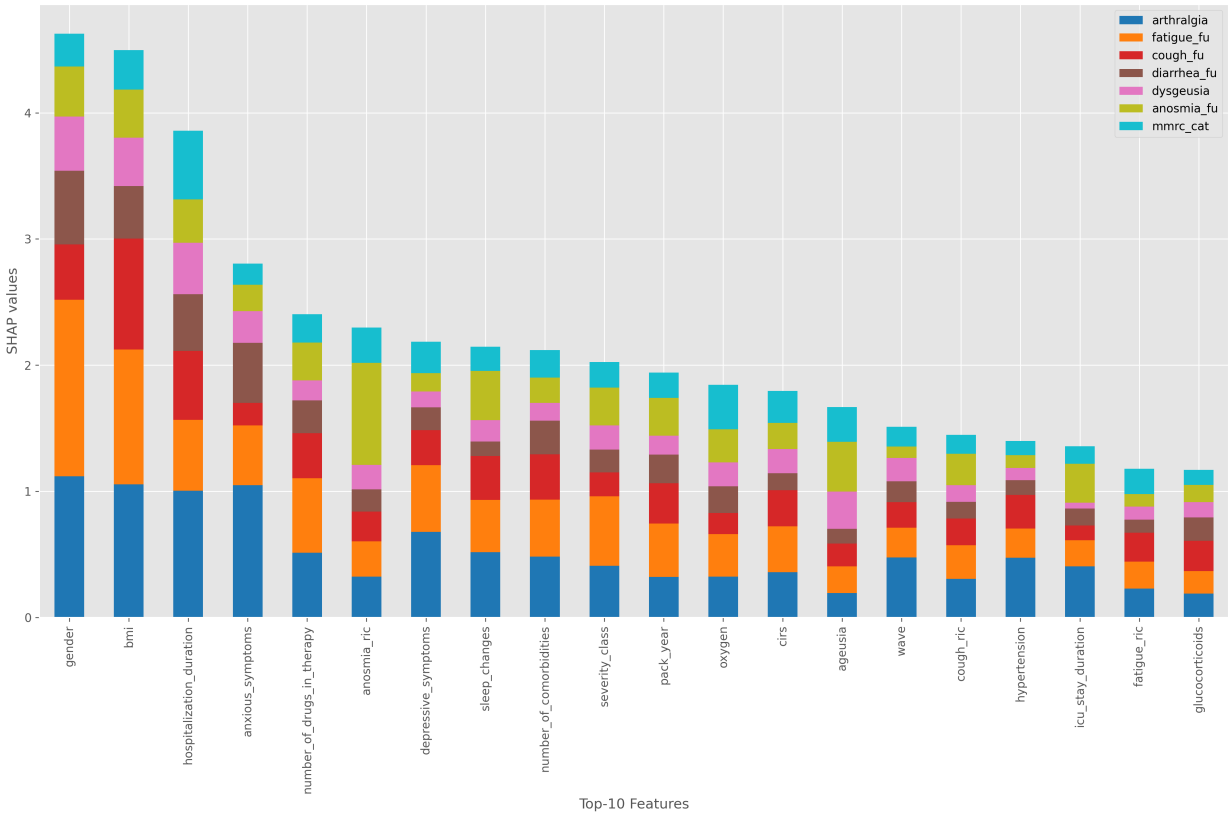

Figure A.1: SHAP values for the original dataset using a Random Forest model.

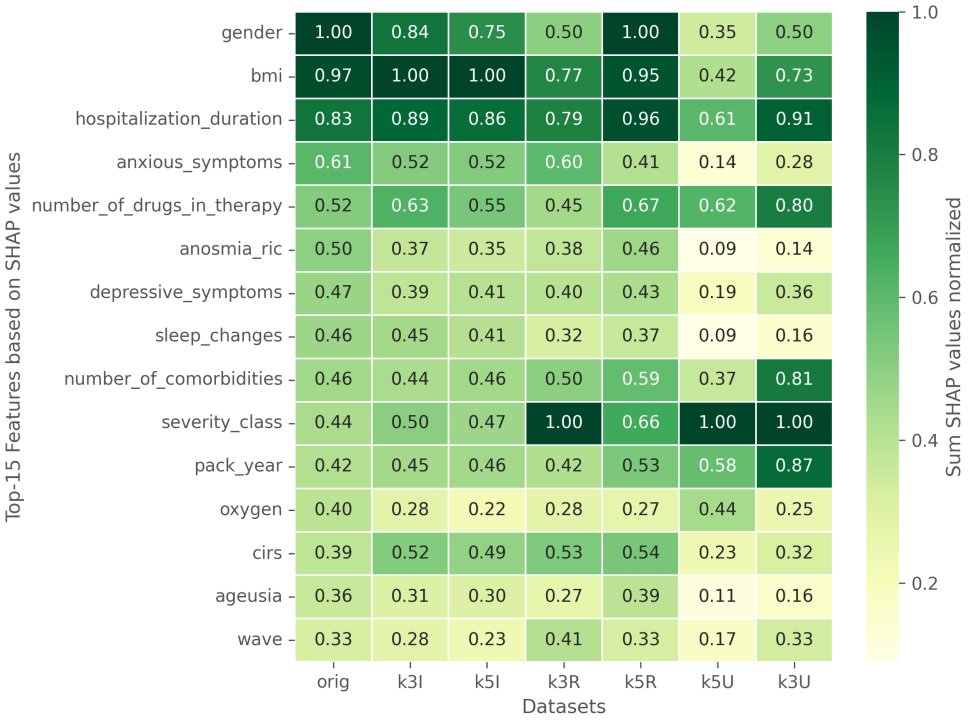

Figure A.2: Comparison of SHAP values between the original dataset and different augmentations. The SHAP values of each dataset have been normalized independently.

A.2 Case Study 2

The following section will present some additional material and plots that complement the main text on the case study of the application SSL methods to small histopathological datasets.

Performance of HMAE. Table A.6 compares the AUC-ROC score and weighted F1-score of HMAE against baseline models on the BRACS dataset for the 3-class task (distinguishing benign, atypical, and malignant tumors). Table A.7 contrasts HMAE with baselines on the 7-class subtype classification task, which differentiates between: Normal, Pathological Benign, Usual Ductal Hyperplasia (UDH), Flat Epithelial Atypia (FEA), Atypical Ductal Hyperplasia (ADH), Ductal Carcinoma In Situ (DCIS), and Invasive Carcinoma.

Interpretability and dimensionality reduction. Figure A.3 reports the t-SNE map of the embeddings produced by the HMAE on the BRACS dataset. Figure A.4 reports the attention-masks produced by the HMAE on 3 RoIs of the BRACS dataset.

Model	**AUC**	**WF1-score**
Max-pooling	0.823±0.033	0.596±0.029
Mean-pooling	0.739±0.007	0.522±0.038
Clam-SB [95]	0.863±0.005	0.631±0.034
TransMIL [122]	0.841±0.006	0.631±0.030
DSMIL [88]	0.816±0.028	0.577±0.028
DTFD-MIL [154]	0.870±0.022	0.612±0.080
IBMIL [89]	0.871±0.014	0.645±0.041
MHIM-MIL [129]	0.865±0.017	0.625±0.060
ABMIL [65]	0.866±0.029	0.680±0.051
ACMIL [156]	**0.888±0.010**	**0.722±0.030**
HMAE	0.866±0.003	0.704±0.009

Table A.6: Classification (3 classes) results: AUC and weighted F1-score comparison.

Model	WF1-score
CLAM-MB/B [95]	0.548±0.010
CGC-Net [158]	0.436±0.005
Patch-GNN [8]	0.521±0.006
TG-GNN [111]	0.559±0.001
CG-GNN [111]	0.566±0.013
HACT-Net [111]	0.615±0.009
TransPath [143]	0.567±0.02
TransMIL [122]	0.575±0.007
ScoreNet [127]	**0.644±0.009**
HMAE	0.578±0.015

Table A.7: Sub-type classification task: weighted F1-score

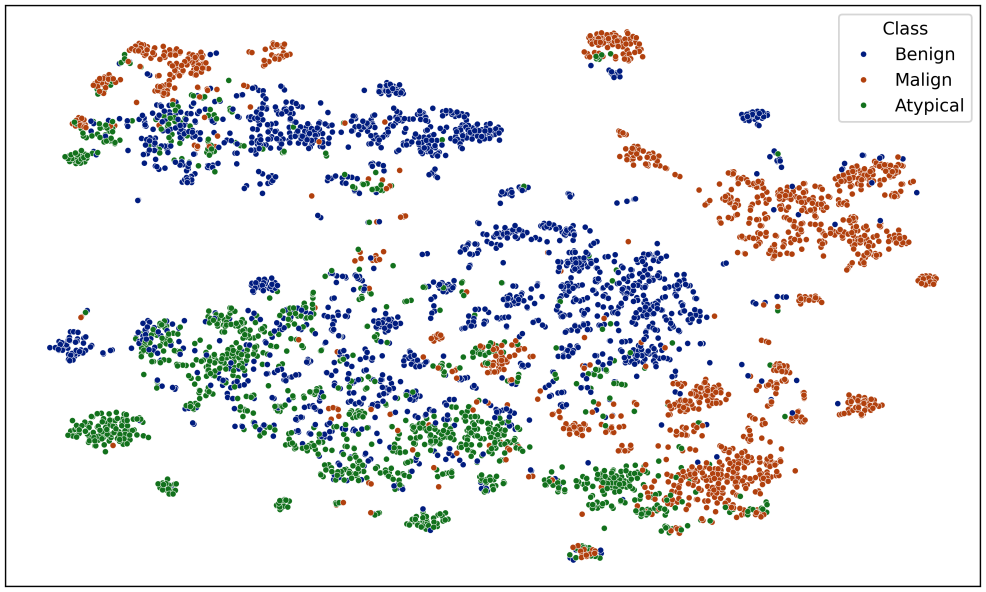

Figure A.3: The t-SNE representation of the RoI embeddings generated by the HMAE.

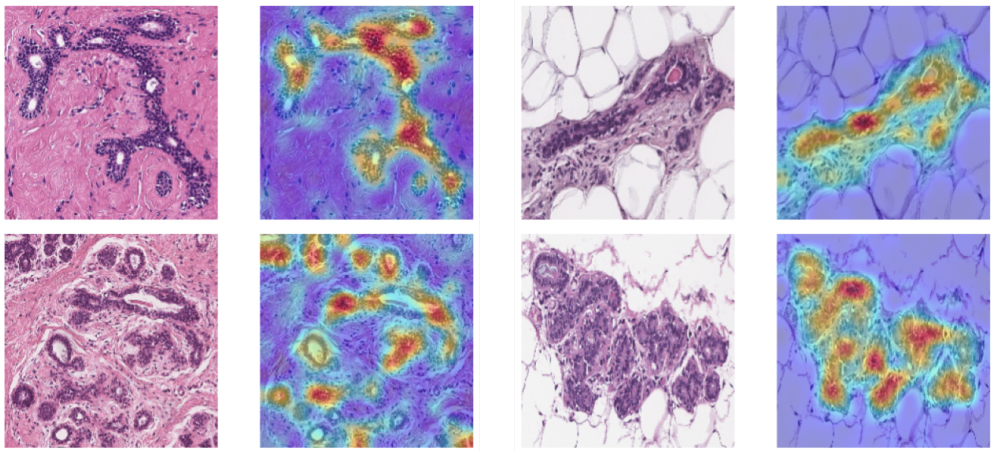

Figure A.4: The attention maps produced by the HMAE on some sample RoIs.

A.3 Chapter 5

Dataset details

In this section, we present the prompts used for data extraction using Gemini.

Prompt used in the experiments to extract the graphical stream

```
For each pdf file that I send you will respond in the following way:
    1. Put the content of the decision tree/graph in a valid JSON format.  It has to be structured as a
       decision tree.  If the branch of a decision is unclear or missing, get it from the text.  The JSON
       must contain these keys:

          a) node:  containing the node name
          b) content:  a brief description of the node
          c) children:  a list of children node (can be optional for leaf nodes)
          d) make a root node with the name of the condition

    2. Make sure that the JSON is valid and well structured.  Make sure that you don't get 'Invalid
       control character at line' errors.
```

The system prompt used to polish raw text extracted from the PDF (textual stream) is:

Prompt used to refine raw text extracted from a PDF

```
The uploaded PDF contains information and decisions to be made about a specific disease/condition.
Each PDF consists of (sorted from the top of the first page):  a title, the authors, a brief description,
observations (begin with a capital letter followed by a period), a decision graph, a continuation of an
observation if it did not fit on the previous page, references.
In some documents there may be tables, ignore them.
```

While the single message with the single condition attached (PDF file):

Prompt to extract the condition from the PDF file

```
Analyze the PDF provided and extract the brief description and observations.  The title, authors, decision
graph and references should NOT be extracted.  There must be ONLY text contained in the provided pdf in
the output.
```

Additionally, we provided an example of path refinement using a real medical case.

System Prompt (path refinement)

```
Given a sequence of reasoning and associated condition, you have to refine it to be uniform with medical
terminology.
The meaning of the sequence must NOT change (at most, you can remove superfluous information or sanitize
the text).  If the reasoning step is poorly explained or ambiguous, refine it using the context and
medical knowledge.
The answer must be a sequence of reasoning with -> indicating the transition between one step and the
next.
Do not add any more text or reasoning in your answer, just the sequence.
```

Pleural Effusion Decision Paths before and after refinement process

Old Path

1. Imaging: PA & lateral CXR; consider decubitus film; consider ultrasound
2. Suspect CHF or viral pleurisy
3. No or atypical progression
4. Thoracentesis
5. Light's criteria: TP_f/TP_s > 0.5. LDH_f/LDH_s > 0.6. LDH_{pf} > 2/3 upper limit of normal.
6. Exudate
7. Cell count & differential
8. Mononuclear cell predominant. Viral infection. Chronic causes

New Path

1. Obtain radiological images: PA & lateral CXR; consider decubitus film or ultrasound
2. Suspect CHF or viral pleurisy
3. No or atypical progression
4. Perform thoracentesis
5. Apply Light's criteria: TP_f/TP_s > 0.5.LDH_f/LDH_s > 0.6. LDH_{pf} > 2/3 upper limit of normal.
6. Exudative effusion identified
7. Perform cell count & differential
8. Mononuclear cell predominance
9. Consider viral infection or chronic etiologies

System Prompt (question generation)

```
Given a sequence of reasoning and a text related to it about how to treat a symptom/condition, generate:
1. a question reflecting the reasoning of the sequence provided. The question must include a clinical case, e.g: "A 67-year-old man is brought to the physician because of increasing forgetfulness, unsteadiness, and falls over the past year..."
2. A set of 5 possible answers (A,B,C,D,E). Should not be too long and should also reflect the sequence of reasoning. One of them must be the correct answer. The other answers need not be correct for the generated question but must be related to the topic of the question.
The sequence does NOT have to be explicit in both question and answers!
The correct option must be the same as the answer.
The output should be structured in the following format: ["question", "answer", "['Option A', 'Option B', 'Option C', 'Option D', 'Option E']", "letter of correct option"]
Do not generate any additional texts.
```

Generation question prompt

```
The reasoning sequence is as follows: {path}, the context associated is: {text} and the symptom/condition to be treated is: {cond}.
```

LLM-as-a-judge implementation

Prompt used for the evaluation phase with the LLM-as-a-judge.

LLM-as-a-judge prompt

```
*** TASK: Based on the following task description and evaluation criteria, generate a detailed Chain of Thought (CoT) that outlines the necessary Evaluation Steps to assess the solution. The CoT should clarify the reasoning process for each step of evaluation.
*** INPUT:
```

```
TASK INTRODUCTION: You are an evaluator for Q&A medical tasks. You will evaluate the quality of answers
to medical questions. You will have some ground truth to help you evaluate.
You will be given 4 inputs:
    • The question
    • The ground truth answer
    • The reasoning path that entails the answer
    • The given answer which needs to be evaluated
The inputs will be given in csv format as a batch of 3 samples). The columns will contain:
    • question
    • reasoning path
    • real answer
    • predicted answer
The output must be a csv formatted including ALL the examples given in the input. But only of a column
indicating the score
*** EVALUATION CRITERIA:
The evaluation criteria must take into account how much the given answer (input 4) is adherent to the
reasoning path and to the ground truth answer.
The evaluation MUST not take into account the "style" of the answer but only the content. The score must
be an integer between 0 to 10.
FINAL SCORE:
IF THE USER'S SCALE IS DIFFERENT FROM THE 0 TO 10 RANGE, RECALCULATE THE VALUE USING THIS SCALE.
SCORE VALUE MUST BE AN INTEGER.
```

Results per category

Figure A.5 reports the breakdown of the performance on different categories off the HealthBranches dataset.

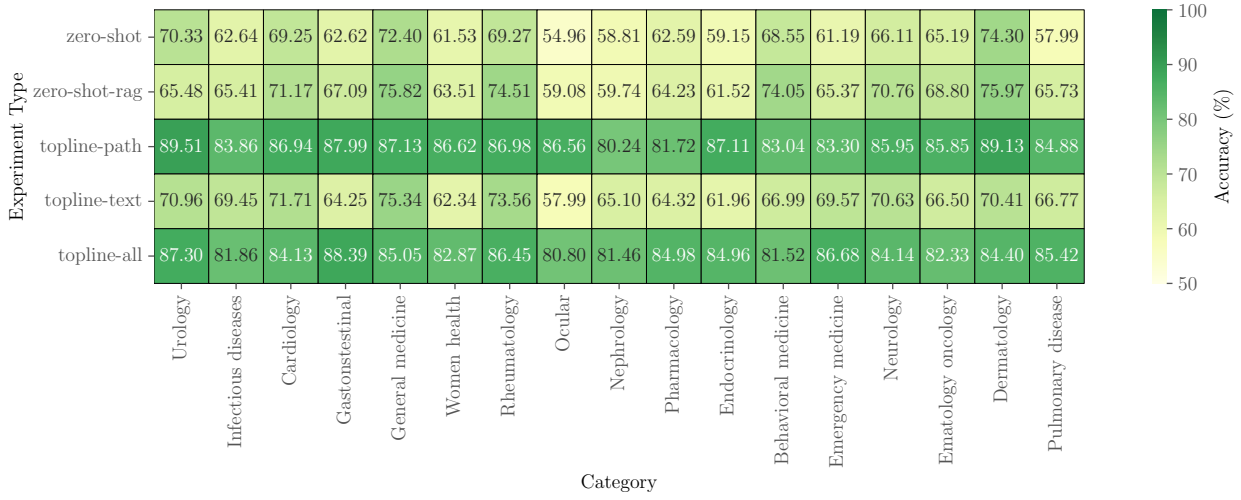

Figure A.5: Average accuracies on different categories and different setups.

Medical Validation Protocol

This section describes the structured protocol used by medical professionals to review and validate the Q&A content in our dataset. The following protocol was administered to students, medical specialists, and doctors using a simple and user-friendly web app to evaluate a sample of Q&A.

1. **Purpose.** To provide a structured framework for medical professionals to assess and ensure the accuracy, clarity, dependability, and identification of potential misinformation or factual inaccuracies in the Q&A content provided to patients and the general public.
2. **Scope.** Applicable to all medical professionals engaged in the evaluation process (physicians, specialists, medical researchers). Focuses on verifying:
   - Content accuracy
   - Adherence to current medical guidelines
   - Suitability for patient education
3. **Reviewer Qualifications.** Individuals must meet at least one of the following criteria:
   - Medical Student
   - Doctor / Physician
   - Medical Specialist
4. **Review Process.**

   4.1 **Selection of Q&A Content.** A curated set of Q&A instances is chosen to span a broad range of medical topics.

   4.2 **Evaluation Parameters and Scoring.** Each Q&A entry is scored on a 1–5 scale (1 = Poor, 5 = Excellent) according to:
   - *Question Medical Accuracy:* Alignment with current medical knowledge and reliable sources.
   - *Answer Medical Accuracy:* Correctness of the diagnosis or explanation. If incorrect, the reviewer selects the appropriate answer.
   - *Path Medical Accuracy:* Correctness and completeness of the management steps outlined.

   Total score ranges from 3 to 15. Entries scoring below a predefined threshold (i.e., 9/15) are flagged as incorrect.

   4.3 **Providing Feedback and Documentation.**

- Entries below threshold of 3/5: reviewers submit detailed feedback highlighting strengths and areas for revision.

4.4 **Structured Review Form.** A web-app form is used to collect expert reviews:

- Each Q&A instance is displayed with fields for scores and free-text comments.
- The form auto-calculates total scores and flags entries below threshold.
- Aggregated results are exported for trend analysis and identification of common issues.

5. **Implementation of Revisions.**
   - The Q&A development team reviews all feedback.
   - Clarifications or further expert input are obtained as needed.
   - A final verification pass is performed before publishing revisions.
6. **Ongoing Review and Quality Assurance.**
   - Periodic re-assessment of Q&A content to ensure continued accuracy and relevance.
   - Analysis of reviewer feedback trends to guide future improvements.
   - Protocol updates to incorporate advances in medical science and best practices.
7. **Confidentiality and Ethical Considerations.**
   - Reviewers must adhere to confidentiality agreements regarding unpublished material.
   - All activities must comply with ethical standards and relevant regulatory requirements for medical information dissemination.

A.3.1 Web App Review Examples and Results

For Q&A pairs that received at least one score of 3 or lower, the involved clinicians provided detailed corrections to address common sources of ambiguity. Their reviews systematically highlighted instances where diagnostic modalities were unspecified, key laboratory results were omitted, clinical reasoning was unclear, specificity was insufficient, or terminology and guidelines were outdated. Following the standardized protocol described previously, these expert reviews provided the field-specific details necessary to thoroughly refine and enhance the final Q&A pairs. Table A.8 provides an overview of the scores from the different doctors for the Q&A dataset.

Table A.8: Reviewer performance and feedback statistics by expertise level for randomly selected questions.

Expertise	**Question**	**Answer**	**Path**	**Mean**	**Reviewer**	**Reviews**
Doctor/Physician	4.83/5	4.67/5	4.67/5	14.17/15	3	18
Medical Specialist	4.32/5	4.20/5	4.16/5	12.68/15	5	25
Medical Student	4.74/5	4.79/5	4.72/5	14.25/15	12	102
Total	13.89/15	13.66/15	13.55/15	13.97/15	22	145

Appendix B

Code and Data Availability

This appendix provides comprehensive information about the datasets, code repositories, and computational resources used throughout this thesis. Where possible, all code and publicly available datasets are provided with detailed documentation to facilitate replication and extension of the work.

Chapter 2: Case study 1

Experimental and Implementation Details. All of the experiments were carried out using the following libraries and frameworks:

- `mldr`: a R package used for the exploratory data analysis.
- `MEKA`: a framework for multilabel learning. It was used specifically for training the different MLC models.
- `MLSMOTE`: a python implementation of the MLSMOTE algorithm.
- `SHAP`; a python library used for the interpretability analysis with SHAP.

Data Availability. The presented "Long-COVID" dataset is not available to the public due to privacy concerns.

Chapter 2: Case study 2

Code Availability. The source code used for pre-training and the sampling module for the patches is available at: github.com/christopher-irw/histo_mae

Data Availability. The "BRACS" dataset used for pre-training the MAE is available at: bracs.icar.cnr.it.

Chapter 3: Knowledge injection for process mining

Code Availability. The source code of the transformer model and the implementation of the SPE module are available at: github.com/christopher-irw/proformer_ce. The repository contains the necessary code to train the transformer model on the BPI 2012 challenge. Additionally, an example of a simple ontology of the activities present in the dataset is provided in order to make the SPE method applicable.

Data availability. The presented "Stroke Management" dataset, on which the experiments were carried out, is unavailable to the public due to privacy concerns.

Chapter 4: Knowledge Injection for High-Dimensional Data

Code Availability. The necessary code for implementing the proposed architecture is available at: github.com/christopher-irw/BYOG.

Data Availability. The IDB dataset used throughout the experiments is available at: ibdmdb.org. The repository provides the data of different omic levels, which were merged during the experiments.

Chapter 5: A Framework for Extracting and Evaluating Clinical Reasoning

Code Availability. The necessary code to run the experiments is available at: anonymous.4open.science/r/HealthBranches-480E. The repository contains the used code to generate the dataset data and to benchmark the presented models.

Data Availability. The HealthBranches dataset is available at Kaggle. While the paths and graphs are available in the git repository.

Experimental Setup. Inference on the LLMs models was performed using the Ollama framework (v0.6.3), with CUDA and CuDNN support. Two quantization schemes were used: `Q_0`: 8-bit weight representation, no further compression (Gemma1/2, Mistral, Llama2, Mistral-nemo). `Q4_K_M`: 4-bit k-means quantization (Gemma3, Qwen2.5, Deepseek-r1, Llama-3.1, Phi4). All the experiments were run on a node equipped with an NVIDIA Quadro RTX 6000 GPU.

The experiments were completed on April 12th, 2025; therefore, models released after this date are not included in the benchmark evaluation.