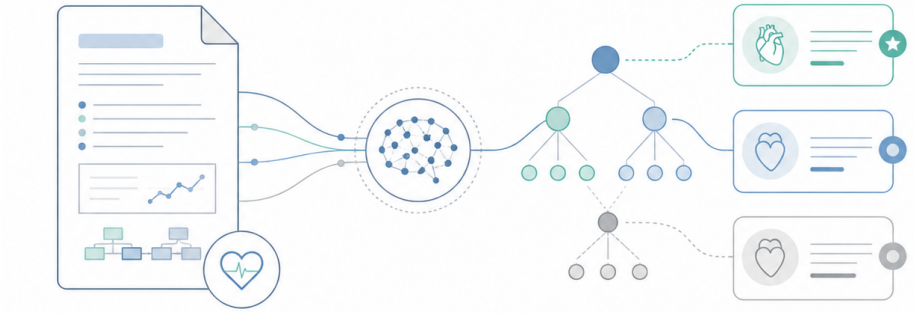
CHALMERS
UNIVERSITY OF TECHNOLOGY



AI-Powered Recommender System for Clinical Trial Protocol Design

A Tool for Medical Practitioners

Master's thesis in Computer science and engineering

ALBIN ENSTRÖM & ROBIN KHATIRI

Department of Computer Science and Engineering
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2026

Master's thesis 2026

AI-Powered Recommender System for Clinical Trial Protocol Design

A Tool for Medical Practitioners

ALBIN ENSTRÖM & ROBIN KHATIRI

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Computer Science and Engineering
Chalmers University of Technology
Gothenburg, Sweden 2026

AI-Powered Recommender System for Clinical Trial Protocol Design
A Tool for Medical Practitioners
ALBIN ENSTRÖM & ROBIN KHATIRI

Supervisor: Farzaneh Jalalypour, CSE
Advisor: Ali Soltani, AstraZeneca/Evinova
Examiner: Ashkan Panahi, CSE

Master's Thesis 2026
Department of Computer Science and Engineering
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Cover: Simplified illustration of the thesis pipeline, showing clinical trial protocol data being transformed through AI-based structuring into hierarchical endpoint recommendations for heart failure studies. The image was generated with assistance from ChatGPT.

Typeset in LaTeX
Gothenburg, Sweden 2026

AI-Powered Recommender System for Clinical Trial Protocol Design
A Tool for Medical Practitioners
ALBIN ENSTRÖM & ROBIN KHATIRI
Department of Computer Science and Engineering
Chalmers University of Technology

Abstract

Clinical trial endpoint selection is a complex protocol-design task that requires clinical relevance, statistical validity, regulatory awareness, and practical feasibility. In heart failure trials, this is particularly challenging because endpoint descriptions are heterogeneous, clinically nuanced, and often expressed using different terminology. This thesis investigates whether historical clinical trial data can be transformed into structured, terminology-aware representations that support secondary-endpoint recommendation for heart failure protocols.

In collaboration with AstraZeneca and Evinova, the study develops an end-to-end proof-of-concept pipeline. Starting from 2 966 raw ClinicalTrials.gov protocol records, the dataset is reduced to 490 Phase II–III heart-failure-focused protocols containing 878 primary endpoints and 3 700 secondary endpoints. Secondary endpoints form a reviewed hierarchy using semantic embeddings, hierarchical clustering, terminology-assisted standardization, and LLM-assisted review. Protocol and endpoint information is standardized against NCIt, CDISC, and LOINC for a two-stage recommendation pipeline: Stage 1 predicts relevant endpoint clusters, while Stage 2 ranks concrete secondary endpoint candidates within the predicted cluster context.

The results show that hierarchical endpoint structuring provides a more interpretable and model-compatible representation than flat clustering or direct prediction over raw endpoint strings. Standardized terminology codes improved semantic consistency and contributed useful supporting features, but were most effective when combined with the reviewed hierarchy and partial endpoint context. Pairwise leave-one-out formulations were better aligned with the intended recommendation setting than direct multilabel prediction, especially for identifying missing endpoint information from a partially specified endpoint design. Full-pipeline evaluation on unseen protocols showed limited exact-match recovery, but qualitative expert review indicated that many recommendations captured clinically relevant endpoint domains, even when they were not specific enough to replace the held-out endpoint directly.

Overall, the thesis demonstrates that historical clinical trial records can be reused more systematically to support endpoint-selection discussions. The proposed pipeline should be interpreted not as a production-ready clinical tool, but as a methodological foundation for AI-assisted endpoint recommendation. Future work should focus on broader therapeutic-area validation, improved terminology resources, stronger expert-labelled evaluation sets, and prospective testing with protocol designers.

Keywords: AI, artificial intelligence, AI Systems, ML, machine, learning, machine learning, clinical, trials, clinical trials, protocol, data science, computer science.

Acknowledgements

The authors would like to express sincere gratitude to everyone who supported this thesis project. The guidance, feedback, clinical insight, technical input, and practical support received throughout the project were highly valuable and greatly appreciated.

Special thanks are extended to Farzaneh Jalalypour, supervisor at Chalmers University of Technology, for academic guidance, continuous feedback, and support with the thesis writing and presentation.

Sincere thanks are also extended to Ali Soltani, supervisor at Evinova, for weekly guidance, technical and domain-specific support, and for helping shape the study throughout all stages of the project.

The authors are grateful to Ashkan Panahi, examiner at Chalmers University of Technology, for his role in the examination of this thesis.

Additional thanks are extended to Ruud Kalis, on-site manager at Evinova in Gothenburg, for practical support, local coordination, and help with setting up the work environment.

Finally, sincere thanks are extended to Niklas Bergh at AstraZeneca for clinical expertise, expert review, and valuable feedback on the relevance and interpretation of the endpoint recommendations.

Robin Khatiri and Albin Enström

Contents

List of Figures

List of Tables

1

Introduction

Clinical trial design is a complex process that requires balancing clinical relevance, statistical validity, regulatory expectations, and practical feasibility. Decisions made at the protocol-design stage shape not only how a study is conducted, but also the quality and interpretability of the evidence it ultimately produces. As the amount of available clinical trial data continues to increase, there is growing interest in how computational methods can support more informed and systematic design decisions.

In a clinical trial, endpoints are the measurable outcomes used to evaluate whether an intervention has the intended effect or produces relevant risks. Endpoints are commonly divided into primary, secondary, and sometimes exploratory endpoints. Primary endpoints define the main evidence question of the trial and are used to determine whether the study meets its main objective, while secondary endpoints provide supporting or additional information about treatment effects, such as symptoms, biomarkers, safety outcomes, hospitalization, or quality of life. Exploratory endpoints are mainly used to investigate additional hypotheses and are not treated as prediction targets in this study [1], [2].

This study examines secondary-endpoint recommendation, specifically for clinical trials within heart failure. In collaboration with AstraZeneca and Evinova, the work investigates how historical trial data, standardized biomedical terminology, and machine learning (ML) can be combined into a staged recommendation pipeline. The focus on secondary endpoints was chosen because they provide a richer and broader design space for learning endpoint patterns. In the reduced dataset used in this study, the 490 retained protocols contain 878 primary endpoints and 3 700 secondary endpoints, corresponding to approximately 1.8 primary endpoints and 7.6 secondary endpoints per protocol. Primary endpoints are therefore used mainly as contextual input to the prediction pipeline, while secondary endpoints form the main recommendation target. The workflow begins with large-scale collection of ClinicalTrials.gov study protocols, reduces and restructures the raw registry data, separates protocol information from endpoint information, standardizes both against controlled biomedical vocabularies, organizes heterogeneous secondary endpoints into a reviewed semantic hierarchy, and finally uses the resulting representations for downstream prediction.

In practical terms, the workflow moves from 2 966 raw ClinicalTrials.gov JavaScript Object Notation (JSON) protocols to a reduced heart-failure-focused cohort of 490 protocols. From there, the prediction task is organized into two linked modeling

views: one focused on predicting clinically relevant secondary-endpoint clusters, meaning groups of secondary endpoints that measure similar concepts, and one focused on ranking concrete candidate secondary endpoints within the predicted cluster context. To better simulate a realistic use case, the final prediction stage is evaluated primarily under a partial-information setting, where some secondary endpoints are treated as already observed and the system attempts to recover or rank the missing endpoint information. Representative examples of the raw JSON structure, reduced protocol records, tabular dataset views, and reviewed hierarchy outputs are provided in Appendix B.

Despite the growing amount of publicly available clinical trial data and the increasing use of computational methods in clinical informatics, endpoint-focused decision support for protocol design remains comparatively underexplored. Existing work has addressed related tasks such as protocol amendment prediction, document similarity, and broader biomedical recommendation, but less attention has been given to transforming raw trial protocols into structured, terminology-aware endpoint recommendations [3], [4]. The present study addresses this through an end-to-end workflow that combines registry-data processing, hierarchical endpoint representation, terminology-based standardization, and two-stage endpoint recommendation. The full pipeline developed in this work is summarized in Figure 1.1.

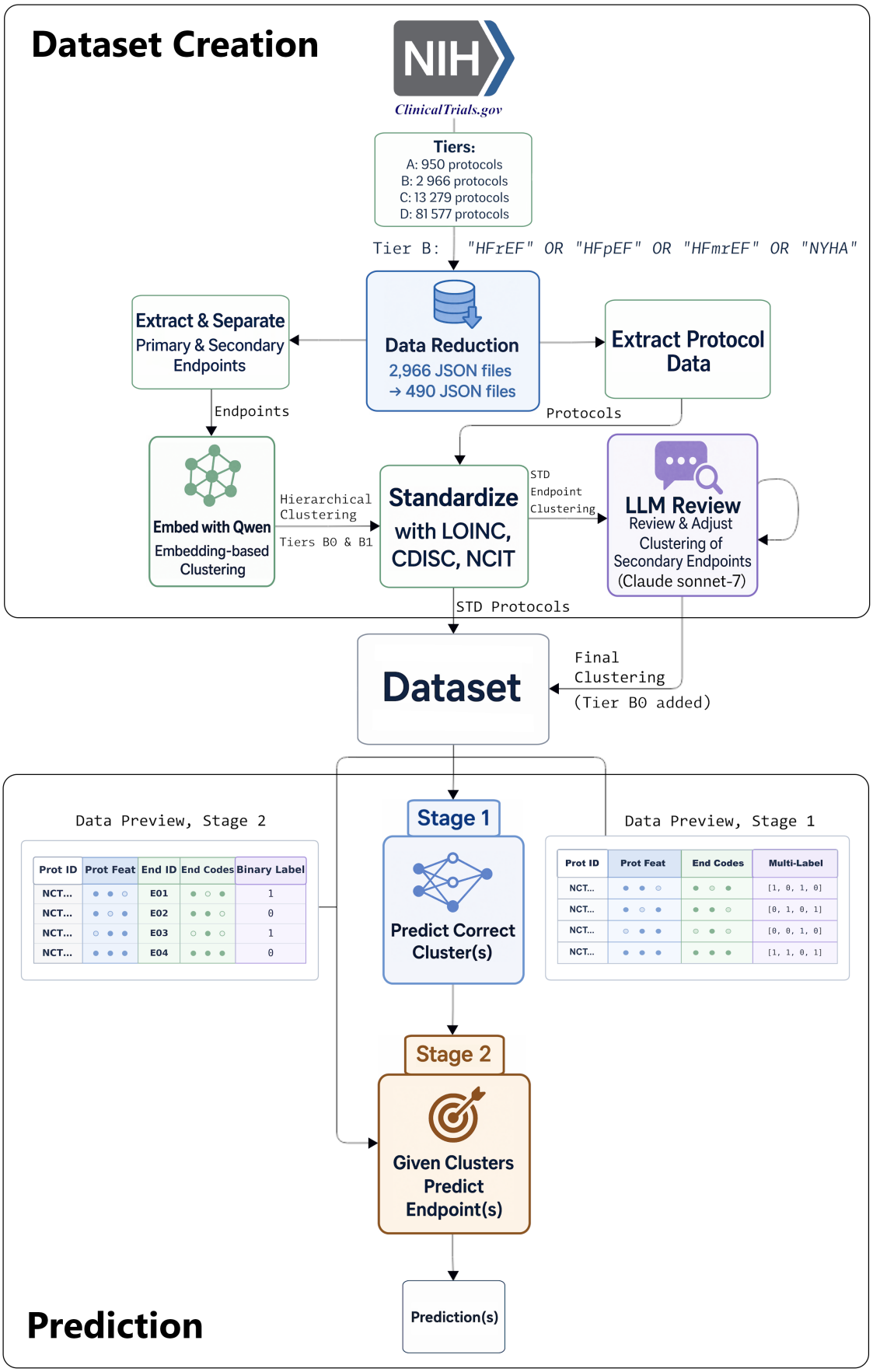


Figure 1.1: End-to-end overview of the proposed pipeline for transforming raw ClinicalTrials.gov protocols into a clinical trial endpoint prediction system.

1.1 Background

Clinical trials are the primary mechanism through which new medical interventions are evaluated before they can be adopted in clinical practice. Their protocols define the scientific and operational blueprint of a study, specifying the target population, intervention strategy, eligibility criteria, study procedures, endpoints, and statistical analyses. As a result, protocol design is not merely an administrative step, but a decisive factor in whether a trial can generate credible, interpretable, and regulatorily acceptable evidence. Poorly specified protocols can lead to inefficiencies, avoidable amendments, delays, and substantial additional cost. Recent work on clinical trial amendments emphasizes that protocol modifications remain frequent and burdensome, and that many of these issues may be more effectively addressed if weaknesses can be identified earlier in the design process [3].

To make this motivation more concrete, Table 1.1 summarizes selected amendment-related evidence from industry and academic settings. Since this study is conducted in collaboration with AstraZeneca and Evinova and is motivated by endpoint support in drug-development protocol design, the industry evidence is the most directly relevant context. The academic evidence is included as complementary support, showing that amendment burden is not limited to commercial trials.

Source / setting	**General amendment burden**	**Design-relevance for this study**
Tufts CSDD industry protocols [5] (Industry)	57% of 836 Phase I–IV protocols had at least one substantial amendment. Phase II/III protocols averaged 2.2/2.3 amendments, with median direct costs of $141k/$535k. About 45% were considered avoidable.	Trial-population changes were reported as the most common amendment type, although no separate eligibility-change percentage was provided.
Tufts CSDD oncology/non-oncology benchmark [6] (Industry)	At least one amendment occurred in 91.1% of oncology and 72.1% of non-oncology protocols, with means of 4.0 and 3.0 amendments per protocol.	Assessment-related changes were common; 16.7% involved efficacy assessments and 36.0% modified statistical methods or analysis.
Updated Tufts CSDD amendment-practice benchmark [7] (Industry)	Across 950 protocols and 2,188 amendments, amendment prevalence increased from 57% to 76%, with a mean of 3.3 amendments per protocol. The amendment process averaged 260 days from need identification to final oversight approval, and sites operated with different protocol versions for a mean of 215 days.	This reinforces that protocol amendments remain frequent and operationally burdensome, motivating earlier design-stage support.
NHS-sponsored academic trials [8] (Academic)	In 53 NHS-sponsored trials, 53% submitted at least one amendment. Across 242 analyzed amendments, 57% were substantial, with a mean of 4.5 amendments per trial.	Adding sites and achieving recruitment targets were the most common amendment change and reason. Eligibility changes were noted qualitatively but not quantified as dominant.

Table 1.1: Summary of amendment-related evidence motivating earlier protocol-design decision support.

The evidence in the table is not heart-failure-specific, but describes protocol amendments across broader clinical trial settings. This is still relevant because the argument made here is general: protocol amendments are common, costly, and often connected to design choices that must be made before or during protocol development. The table should therefore be interpreted as contextual motivation rather than as direct evidence that endpoint changes are always the dominant amendment type.

Taken together, these findings support the broader motivation for design-stage decision support. They show that protocol changes are frequent and operationally burdensome across several clinical trial settings, especially in industry-sponsored drug-development contexts. The present study does not attempt to predict amendments directly, nor does it claim that endpoint changes are the only or main cause of amendments. Instead, it focuses on one important protocol-design component where earlier computational support may be valuable: endpoint selection.

1.1.1 Endpoint Selection as a Protocol-Design Challenge

Within this broader protocol-design problem, endpoint selection is especially important because endpoints determine how treatment benefit or risk is operationalized and what evidence the trial ultimately produces. The choice of endpoints therefore affects statistical validity, clinical interpretability, and downstream regulatory decision-making.

In practice, endpoint selection remains difficult to standardize. Protocol designers often rely on prior experience, therapeutic-area conventions, and manual review of historical studies. Although this can be effective in expert hands, it is difficult to scale and reproduce when large volumes of prior trial data must be considered. In the context of the workflow shown in Figure 1.1, this means that meaningful endpoint recommendation cannot begin at the modeling stage alone. Data reduction, endpoint separation, standardization, and semantic organization are also necessary parts of the endpoint-selection problem.

1.1.2 Heart Failure as a Heterogeneous Endpoint Domain

These challenges are particularly evident in heart failure. Heart failure is a chronic and progressive clinical syndrome in which the heart is unable to pump blood sufficiently to meet the body's metabolic demands. It is associated with high morbidity, frequent hospitalizations, substantial mortality, and major healthcare burden. Its clinical severity and long-term progression make it an important domain for well-designed clinical trials, where endpoint selection directly affects the ability to detect meaningful treatment effects.

A central source of complexity is that heart failure includes several clinical phenotypes. It is commonly categorized by left ventricular ejection fraction (LVEF), most notably into heart failure with reduced ejection fraction (HFrEF), heart failure with preserved ejection fraction (HFpEF), and heart failure with mildly reduced ejection fraction (HFmrEF). These subtypes differ in pathophysiology, patient characteristics, treatment response, and expected clinical outcomes. For example, HFrEF has a

more established evidence base with endpoints such as mortality and hospitalization, whereas HFpEF is often more heterogeneous and may rely more heavily on functional, symptomatic, or biomarker-based endpoints.

As a result, heart failure trials use a broad range of outcome measures, including mortality, hospitalization, symptom burden, functional capacity, imaging-derived cardiac function, circulating biomarkers, and patient-reported outcomes. Similar endpoint concepts may also be expressed using different wording across studies. The historical endpoint landscape is therefore rich but difficult to compare directly, which motivates the need for semantic structuring and terminology-aware representation before downstream recommendation can be attempted.

1.1.3 Heart Failure as a Suitable Use Case for Data-Driven Recommendation

At the same time, heart failure provides a particularly suitable setting for data-driven analysis and recommendation. Compared to many other therapeutic areas, it benefits from a relatively large volume of publicly available clinical trial data, well-established clinical endpoints, and clear regulatory precedents. This combination makes it possible to construct a sufficiently rich dataset while still working within the constraints of a master's thesis. In addition, the coexistence of standardized outcomes and heterogeneous endpoint formulations makes heart failure an ideal test case for evaluating whether semantic structuring, terminology normalization, and ML can bridge the gap between raw trial descriptions and actionable design recommendations.

1.1.4 Computational Decision Support and Biomedical Recommendation

The digitization of biomedical and clinical data has created new opportunities for computational decision support. In life and health sciences, recommender systems have been used to help decision-makers navigate large spaces of biomedical items, such as drugs, diseases, genes, patients, or health information [4]. This makes recommender-system thinking relevant to endpoint selection, where the goal is to identify suitable trial design components from historical protocol patterns.

At the same time, biomedical recommendation is more difficult than many conventional recommendation tasks. The data is often sparse, heterogeneous, difficult to standardize, and not naturally organized as a clean user-item-rating matrix [4]. For clinical trial design, this means that endpoint recommendation requires more than a predictive model alone. As mentioned, it also requires structured representations and evaluation methods that reflect practical recommendation usefulness.

1.1.5 Clinical-Trial NLP and Amendment Prediction

Recent clinical-trial NLP research further supports the idea that protocol design can be approached computationally. For example, AMEND++ formulates eligibility-

criteria amendment prediction as a supervised NLP task and shows that information in an initial protocol section can contain signals about later amendments [3]. Although AMEND++ focuses on eligibility criteria rather than endpoints, it is relevant here because it demonstrates how historical protocol data can be used for earlier design-stage decision support. The present study addresses a neighboring problem: instead of predicting amendment risk, it investigates how historical protocol and endpoint information can support endpoint recommendation.

1.1.6 Endpoint Recommendation in This Study

Against this background, this study investigates how a data-driven recommender system can support endpoint selection for heart failure clinical trials. Rather than treating raw endpoint strings as isolated labels, the study assumes that endpoint recommendation requires semantic structuring, terminology-aware normalization, and models that can operate under realistic small-data and high-heterogeneity conditions.

The following literature review situates this study within three intersecting research areas: clinical trial design and protocol optimization, biomedical recommender systems, and ML methods for structuring and modeling clinical text. It provides the conceptual basis for the methodological choices developed later in Chapter 4.

1.2 Problem Statement

Although large volumes of historical clinical trial data are publicly available, selecting appropriate endpoints for new protocols remains largely manual, expert-driven, and difficult to systematize. Protocol designers must interpret heterogeneous trial records, reconcile inconsistent endpoint terminology, and decide which previous outcomes are relevant for a new study. This is particularly challenging in heart failure, where endpoint suitability depends on phenotype, intervention, study phase, and trial objective.

The problem addressed in this study is therefore both a representation problem and a prediction problem. More specifically, this study investigates how endpoint recommendation for heart failure trials can be supported through a data-driven pipeline that combines semantic endpoint structuring, terminology-aware normalization, cluster-level prediction, and pairwise endpoint ranking. Without such structuring, ML models risk learning from superficial wording patterns rather than clinically relevant endpoint relationships. The goal is to determine whether historical protocol data can be used to recommend clinically relevant endpoint clusters and candidate secondary endpoints in a way that is both practically useful and methodologically transparent.

Several challenges make this problem non-trivial:

- Endpoint descriptions are highly heterogeneous, often differing in wording, granularity, and clinical specificity even when referring to closely related concepts.

- The available dataset is limited in size relative to the complexity of the task, which could increase sensitivity to sparsity, class imbalance, and overfitting.
- Endpoint recommendation is inherently a multi-label problem, since a single protocol may require multiple complementary endpoints rather than one isolated outcome.
- Clinical trial data is strongly dependent on domain-specific standards and terminology systems, which must be incorporated to achieve meaningful normalization and interoperability.
- Recommendations must be interpretable enough to support expert review, rather than functioning as opaque predictions detached from clinical reasoning.

Taken together, these challenges motivate an exploratory approach in which data preparation, semantic structuring, standardization, and modeling are developed as connected parts of the same pipeline. The study therefore aims not only to build a proof-of-concept recommendation system, but also to identify which representations and modeling choices are most suitable for endpoint-focused decision support in clinical trial design.

1.3 Research Objectives

The overall objective of this study is to investigate how AI and ML can support clinical trial protocol design through data-driven endpoint recommendation in heart failure studies. As illustrated in Figure 1.1, this objective spans the entire workflow from raw ClinicalTrials.gov JSON protocols to structured endpoint representations and downstream prediction, rather than being limited to just training the best ML model.

To achieve this objective, the study pursues the following goals:

- Construct a curated dataset of heart failure clinical trials and derive structured representations of protocol characteristics and endpoint descriptions.
- Develop a semantically meaningful organization of heterogeneous endpoint descriptions in order to reduce linguistic variability and create clinically interpretable target labels.
- Examine how standardized clinical vocabularies and terminology systems can be used to normalize endpoint concepts and improve downstream modeling.
- Design and evaluate ML approaches for cluster-level and endpoint-level recommendation, including both from-scratch prediction and partial-information recommendation settings.
- Compare alternative representation and modeling choices under realistic small-data conditions, with particular attention to robustness, interpretability, and practical usefulness.

- Identify the main methodological challenges, limitations, and opportunities involved in applying AI-based recommendation methods to clinical trial design.

More concretely, the study is guided by the following research questions:

- How can unstructured and heterogeneous endpoint descriptions from heart failure trials be transformed into a structured representation suitable for computational analysis?
- To what extent can protocol characteristics, primary endpoints, and partially observed secondary-endpoint information be used to recommend relevant endpoint clusters and candidate secondary endpoints for heart failure studies?
- How do different representation choices, clustering strategies, and ML methods affect recommendation performance?
- What role do standardized clinical terminologies play in improving semantic consistency, interpretability, and model utility?

Rather than assuming that a single method will provide a definitive solution, this study compares multiple design choices and evaluates their strengths and limitations under realistic data constraints. In one sentence, the study asks whether historical heart-failure trial data can be transformed into a structured and terminology-aware representation from which clinically relevant secondary endpoint clusters and candidate endpoints can be recommended for new protocols.

1.4 Scope and Limitations

This study is situated as a proof-of-concept study rather than as a fully developed production system. The empirical investigation is confined to heart failure clinical trials, which were selected because heart failure is both clinically significant and methodologically suitable for endpoint recommendation. It has a substantial body of historical studies, diverse endpoint practices, and clear variation across protocols. By restricting the analysis to a single therapeutic area, the study remains bounded while still addressing a meaningful real-world problem.

However, although the empirical evaluation is limited to heart failure, the pipeline itself is not hardcoded to this therapeutic area. The main methodological components, including data extraction, endpoint separation, terminology standardization, hierarchy construction, cluster prediction, and candidate endpoint ranking, are designed as general steps that could be re-applied to other therapeutic areas if suitable protocol and endpoint data are available. However, such transfer would require new data processing, hierarchy review, model training, and validation before the approach could be considered reliable outside the heart-failure setting.

The work is limited to endpoint recommendation and does not attempt to optimize the entirety of protocol design. Although endpoint selection is closely related to objectives, eligibility criteria, intervention strategies, and statistical analysis plans, these aspects are not modeled exhaustively. The resulting system should therefore

be understood as one component of a broader decision-support framework rather than a replacement for expert protocol design.

The study is based on retrospective data derived entirely from publicly available clinical trial protocols. It does not include prospective validation in a live protocol-design environment, real-time deployment in industrial software, or direct clinical use during study planning. More detailed methodological limitations and threats to validity are discussed separately in Section 4.12.

1.5 Contributions

This study makes the following contributions:

- It presents an end-to-end pipeline for collecting, filtering, and transforming heart failure clinical trial data into ML-ready protocol and endpoint representations.
- It develops a structured approach for organizing heterogeneous endpoint descriptions into semantically meaningful and clinically interpretable groups, thereby reducing variation in raw outcome terminology.
- It investigates how standardized clinical terminology systems can be incorporated into the endpoint representation process to improve consistency, normalization, and interoperability.
- It formulates endpoint recommendation through both multi-label cluster prediction and pairwise candidate ranking, and evaluates ML approaches for predicting relevant endpoint clusters and ranking candidate secondary endpoints.
- It contributes a generalizable methodological framework for endpoint recommendation that is not directly hardcoded to heart failure, and that could be re-applied to other therapeutic areas given suitable data, terminology resources, hierarchy review, and validation.
- It analyzes the practical challenges of applying AI methods to clinical trial design under realistic data constraints, including sparsity, heterogeneity, interpretability requirements, and semantic ambiguity.
- It provides a proof of concept for how historical clinical trial data can be reused more systematically to support endpoint-focused protocol design in heart failure.

Taken together, these contributions position the study at the intersection of clinical informatics, natural language processing, and recommender systems. Beyond the specific models evaluated, the work contributes a methodological foundation for future research on AI-assisted protocol design and offers a framework that can, with further validation and extension, be adapted to other therapeutic areas.

In practical terms, the system developed in this study is intended as a decision-support tool. A realistic use case is to help protocol designers identify plausible endpoint families, detect potentially missing secondary endpoints, and ground endpoint-selection

discussions in patterns observed across historical trials. The main methodological novelty lies in the combination of endpoint hierarchy construction, terminology-aware standardization, and two-stage endpoint recommendation under realistic small-data and partial-information conditions.

2

Literature Review

This chapter focuses more specifically on the literature that supports the methodological framing of the study. It reviews three related areas: computational support for clinical trial protocol design, recommender systems in biomedical domains, and semantic retrieval methods for clinical trial text. The purpose is not to repeat the full problem background, but to clarify where the present study fits in relation to existing work.

2.1 Clinical Trial Protocol Design and the Need for Decision Support

Clinical trial protocols define the scientific and operational structure of a study, including the target population, intervention, eligibility criteria, outcomes, study procedures, and statistical analysis plan. Because these components determine how evidence is generated and interpreted, protocol quality has direct implications for trial efficiency, scientific validity, and regulatory acceptability. As noted in the Introduction, endpoint selection is one important part of this broader design problem.

A relevant recent contribution in clinical-trial NLP is the AMEND++ framework by Das et al. [3], which was also introduced briefly in Section 1.1.5. In the literature-review context, the important point is that AMEND++ formulates protocol amendment prediction as a supervised NLP task: given the initial eligibility criteria of a clinical trial protocol, the model predicts whether that section will later be amended. The benchmark consists of AMEND, which captures eligibility-criteria version histories and amendment labels from public trial records, and AMEND_LLM, a denoised subset intended to better isolate substantive eligibility-criteria changes. The authors also propose Change-Aware Masked Language Modeling (CAMLM), a revision-aware pretraining strategy for learning representations that are sensitive to historically unstable protocol text.

AMEND++ is important for the present study for two reasons. First, it demonstrates that protocol text can contain predictive signals about later design changes, supporting the broader idea that clinical trial design can be studied computationally rather than only descriptively. Second, it shows that careful target construction matters: the authors distinguish between raw amendment labels and denoised labels that isolate substantive changes. This is conceptually similar to the present study's

need to transform noisy endpoint text into a more stable and clinically meaningful prediction target.

At the same time, AMEND++ is scoped to eligibility criteria. The authors identify other protocol sections, including outcomes, study design elements, and interventions, as directions for future work [3]. The present study addresses one of these neighboring design components: clinical trial endpoints. Whereas AMEND++ predicts whether a protocol section will change, this study investigates whether historical protocol and endpoint information can be structured and used to recommend endpoint clusters and candidate secondary endpoints.

The wider relevance of AMEND++ is therefore methodological rather than task-identical. It supports the idea that historical trial records can be used for proactive protocol-design support, but it does not directly solve endpoint recommendation. This distinction motivates the endpoint-focused representation and recommendation pipeline developed in Chapter 4.

2.2 Recommender Systems in Biomedical and Clinical Domains

Recommender systems are information-filtering systems that suggest relevant items based on prior information, similarity patterns, or observed interactions. In conventional domains, such systems often recommend products, media, or services. In biomedical and health contexts, the same general idea can be adapted to recommend biomedical items such as drugs, diseases, genes, treatments, patients, or health information [4]. From the perspective of clinical trial design, endpoints can similarly be treated as recommendation targets.

Pato et al. [4] provide a survey of recommender systems for biomedical items in life and health sciences. They organize the field around common recommender-system paradigms, including collaborative filtering, content-based filtering, and hybrid approaches. Collaborative filtering relies on interaction or preference patterns across users and items, while content-based filtering uses item or user attributes. Hybrid approaches combine multiple sources of information to reduce the limitations of individual methods.

Endpoint recommendation in the present study is closest to a content-based or hybrid recommendation setting. The system does not operate on a conventional user-item-rating matrix. Instead, it uses protocol characteristics, standardized terminology codes, endpoint hierarchy labels, and observed endpoint context to recommend endpoint clusters and candidate secondary endpoints. This differs from classical collaborative filtering, but it still follows the broader recommender-system principle of ranking plausible items from a larger candidate space.

A key point in Pato et al.'s survey is that biomedical recommendation differs substantially from recommendation in conventional consumer domains. Biomedical datasets often do not follow the standard ⟨user, item, rating⟩ format, and they are frequently heterogeneous, incomplete, sparse, and difficult to standardize [4]. The survey also

notes that most reviewed studies used model-based collaborative filtering approaches, that many datasets were not available in a standard recommender-system format, and that evaluation often relied heavily on classification metrics [4].

This distinction is also relevant for interpreting the later future-work discussion. The implemented pipeline in this study primarily uses protocol and endpoint content as structured input, but the expert-review discussion in Section 5.7 points toward a more clinically guided organization of the endpoint space, where endpoints and protocols could first be grouped by factors such as intervention mechanism, patient population, disease process, measurement purpose, and time frame before finer endpoint clustering is applied. In recommender-system terms, this would move the approach further toward a hybrid design that combines content-based features with stronger domain-structured grouping.

These observations are directly relevant to endpoint recommendation. The target is also not a single correct item in the ordinary sense: several endpoint combinations may be clinically reasonable for a protocol. This makes endpoint recommendation both a prediction problem and a representation problem. It also explains why this study emphasizes semantic hierarchy construction, terminology standardization, and top-k ranking metrics rather than only thresholded classification accuracy.

Finally, the biomedical recommender-system literature emphasizes interpretability and reproducibility. In clinical settings, recommendations must be understandable to domain experts and should be grounded in representations that can be inspected. This supports the methodological choices made in this study: endpoints are organized into a reviewed hierarchy, terminology codes are used as structured semantic features, and predictions are evaluated not only quantitatively but also through qualitative inspection.

2.3 Controlled Terminologies and Structured Biomedical Knowledge

Before discussing structured biomedical knowledge more broadly, it is useful to introduce the three terminology systems used in this study. The National Cancer Institute Thesaurus (NCIt) is a biomedical ontology and reference terminology used to represent clinical, biomedical, drug, disease, and research concepts [9]. The Clinical Data Interchange Standards Consortium (CDISC) Controlled Terminology provides standardized terms used with CDISC-defined clinical research data standards and is maintained in collaboration with the National Cancer Institute Enterprise Vocabulary Services [10]. Logical Observation Identifiers Names and Codes (LOINC) is a terminology standard for identifying health measurements, observations, laboratory tests, clinical measurements, and documents [11]. In this study, these terminology systems are used as structured semantic resources for representing protocol and endpoint content. Concrete examples of how raw protocol and endpoint text are mapped to NCIt, CDISC, and LOINC codes are provided in Appendix B.8.

Structured biomedical knowledge is an important theme in biomedical recommender

systems. Pato et al. [4] discuss the role of knowledge graphs, where entities and relations are represented in graph form. In biomedical contexts, such graphs can connect entities such as diseases, drugs, genes, proteins, pathways, and clinical concepts. This makes them useful for representing domain structure that would be difficult to capture using flat feature vectors alone.

Knowledge graphs have been proposed as a way to address several limitations of traditional recommender systems, including limited content analysis, overspecialization, and cold-start problems [4]. These issues are relevant in clinical trial design because many protocols and endpoints are sparse, heterogeneous, and only partially comparable. By adding structured side information, knowledge-based representations can make recommendations more interpretable and more robust when direct historical evidence is limited.

The survey distinguishes between embedding-based, connection-based, and unified knowledge-graph recommendation approaches [4]. Embedding-based methods learn vector representations of graph entities and relations, connection-based methods exploit graph paths or relational patterns, and unified methods combine these ideas. The present study does not implement a full knowledge-graph recommender. However, it follows the same general principle: recommendation quality depends on representing biomedical concepts in a structured and semantically meaningful way.

In this study, the structured knowledge layer is implemented through a reviewed endpoint hierarchy and standardized terminology codes rather than through a complete graph database. Endpoint clusters provide parent-child relations between broader and narrower measurement concepts, while NCIt, CDISC, and LOINC codes provide standardized semantic signals. This is not equivalent to a full biomedical knowledge graph, but it serves a related methodological purpose: it gives the prediction models access to structured domain information beyond raw text.

The literature also cautions that structured biomedical knowledge introduces its own challenges. Knowledge-graph-based systems depend on data integration, graph quality, domain expertise, interpretability, scalability, and evaluation design [4]. These concerns also apply to this study. The endpoint hierarchy and terminology mappings improve structure and interpretability, but they are not perfect ground truth. Their construction and limitations are therefore treated explicitly in Chapter 4.

2.4 Evaluation Challenges in Biomedical Recommendation

Evaluation is particularly difficult in biomedical recommendation because the goal is often not only to recover one exact historical item. In many clinical contexts, several recommendations may be plausible depending on the disease subtype, intervention mechanism, study phase, regulatory context, and clinical objective. This makes purely exact-match evaluation informative but incomplete.

Pato et al. [4] distinguish between offline and online evaluation. Offline evaluation uses pre-collected datasets and is easier to repeat, while online evaluation measures

user behavior, feedback, or satisfaction in a deployed or user-facing setting. The survey also describes several metric families, including error metrics, classification metrics such as precision, recall, and F1, and ranking metrics such as mean reciprocal rank and normalized discounted cumulative gain. More details about what these metrics are and why they are relevant can be seen in Chapter 3.

The present study is evaluated primarily offline because the system is a proof of concept rather than a deployed clinical tool. However, the evaluation is not limited to automatic offline metrics. The Results chapter also includes qualitative expert review of ranked endpoint outputs, especially in Section 5.6.2. This expert review is not online evaluation in the strict deployed-system sense, but it serves a related role: it assesses whether the recommendations are clinically meaningful, whether they capture the correct endpoint domain, and whether they would be useful as decision-support signals beyond exact-match recovery.

The recommender-system framing means that ranking metrics are especially important. A model may be useful if it places a clinically relevant cluster or endpoint near the top of a recommendation list, even if it does not produce a perfect thresholded label vector. For that reason, the evaluation later includes top-k metrics, candidate-pool recall, pairwise classification metrics, and qualitative inspection of ranked outputs.

The importance of label quality is also reinforced by AMEND++ [3]. In that work, LLM-based denoising is used to separate substantive eligibility-criteria changes from noisier amendment labels. This illustrates a broader lesson that is directly relevant here: model evaluation depends strongly on how the prediction targets are constructed. If endpoint clusters are noisy, overly broad, or semantically inconsistent, the model may appear weaker or stronger for reasons that reflect target quality rather than only model quality.

For endpoint recommendation, evaluation must therefore be interpreted carefully. Exact recovery of a historical endpoint is a strict criterion, especially when several endpoint formulations may measure closely related clinical concepts. A prediction that misses the exact endpoint string but identifies the correct measurement family may still be useful for protocol design. This is why the Results chapter reports both quantitative metrics and qualitative review of endpoint-level outputs.

2.5 Semantic Similarity Search for Clinical Trial Retrieval

A further related area is semantic retrieval of clinical trial documents using transformer-based sentence embeddings. Unlike keyword search, semantic retrieval aims to identify conceptually related trials even when they use different wording. This is relevant to clinical trial design because similar protocols, interventions, and outcomes may be described using heterogeneous language.

Majumdar presents a notebook-style pipeline for finding semantically similar clinical trials using sentence embeddings and a transformer model [12]. The pipeline repre-

sents clinical trials as dense vectors derived from textual fields such as study title, brief summary, conditions, interventions, and primary outcome measures. These fields are preprocessed and combined into a unified textual representation. The combined text is then encoded using the sentence-transformer model `all-MiniLM-L6-v2`, and similar trials are retrieved using cosine similarity.

This work is relevant because it demonstrates how transformer-based sentence embeddings can support clinical-trial retrieval beyond surface-level keyword matching. It also shows the practical value of combining several trial fields into one representation, rather than relying on a single title or summary field. For the present study, this directly supports the use of semantic representation learning as part of the upstream endpoint-structuring pipeline.

The use of `all-MiniLM-L6-v2` is also methodologically relevant to this study because a compact sentence-transformer setup was used during the early clustering experiments before the final hierarchy was constructed with a larger embedding model. In that sense, Majumdar's pipeline helps motivate the initial use of compact sentence-transformer embeddings as a practical baseline for semantic clinical-trial representation. As described in Section 4.3.2, this study first used the smaller `all-MiniLM-L6-v2`-based setup to compare early clustering strategies, before refining the selected hierarchical strategy with a larger embedding model. The later move toward a larger encoder should therefore be understood as an extension of the same underlying idea: endpoint text can be mapped into a semantic vector space and then clustered or compared using geometric similarity.

However, Majumdar's pipeline addresses trial-level semantic similarity rather than endpoint recommendation. It retrieves trials that are textually and conceptually similar, but it does not define endpoint clusters, standardize endpoint concepts against biomedical terminologies, or train a supervised model to recommend endpoint categories or candidate secondary endpoints. It is therefore best understood as adjacent work: it supports the motivation for semantic trial representation, while the present study extends this direction toward terminology-aware endpoint hierarchy construction and downstream recommendation.

This distinction is important for positioning the contribution of the study. Semantic retrieval helps identify similar studies, while endpoint recommendation requires an additional target representation and a ranking mechanism over endpoint candidates. In the proposed pipeline, embeddings are therefore used as part of endpoint organization rather than as the final output of the system.

2.6 Positioning the Present Study

The reviewed literature supports three main observations. First, clinical trial design is increasingly being treated as a valid target for computational support, but much existing work focuses on specific protocol sections or related tasks such as amendment prediction rather than endpoint recommendation [3]. Second, recommender-system methods provide a useful conceptual foundation, but biomedical recommendation requires special attention to sparse data, non-standard item structures, interpretability,

and evaluation [4]. Third, semantic representation methods can help compare heterogeneous clinical trial text, but trial-level retrieval is not the same as endpoint-level recommendation [12].

This study brings these ideas together in the specific context of heart failure endpoint recommendation. It treats historical trials as an evidence base, transforms raw endpoint descriptions into a reviewed semantic hierarchy, standardizes protocol and endpoint concepts using biomedical terminology systems, and evaluates both cluster-level and endpoint-level recommendation models. In this sense, the study addresses a gap between clinical-trial NLP, biomedical recommender systems, and semantic retrieval.

The concrete realization of this research position is presented in Chapter 4, where Sections 4.1–4.11 describe the dataset construction, semantic hierarchy development, terminology standardization, and two-stage recommendation pipeline.

3

Theory

This chapter introduces the theoretical concepts needed to understand the methodology developed in Chapter 4. The purpose is not to repeat the clinical and literature background from Chapters 1 and 2, but to define the main technical concepts used later in the pipeline: endpoint representation, semantic embeddings, clustering, terminology standardization, supervised learning, ranking, partial-information evaluation, and evaluation metrics.

3.1 Clinical Trial Endpoints as Prediction Targets

Clinical trial endpoints are measurable outcomes used to evaluate the effect, safety, or broader clinical relevance of an intervention. As introduced in Chapter 1, endpoints are commonly divided into primary, secondary, and sometimes exploratory endpoints. Primary endpoints define the main evidence question of the trial, while secondary endpoints provide additional information about treatment effects, safety, symptoms, biomarkers, hospitalization, or quality of life [1], [2].

From a modeling perspective, endpoints are difficult targets because their raw textual descriptions do not always correspond directly to stable clinical concepts. The same measurement may be expressed using different wording, abbreviations, timeframes, or levels of specificity, while similarly worded endpoints may still measure different constructs. Endpoint recommendation therefore requires an intermediate representation that is more structured than raw endpoint text, but still clinically interpretable.

In this study, the final prediction target is not free-text endpoint generation. Instead, secondary endpoints are first organized into a reviewed hierarchy of endpoint clusters, and the models later predict or rank these structured endpoint representations. This makes the task a combination of representation learning and recommendation rather than ordinary text classification.

3.2 Semantic Representation and Text Embeddings

Text embeddings are vector representations of text. Their purpose is to place semantically related texts closer together in a numerical feature space, making

it possible to compare text using geometric measures such as cosine similarity. Sentence-BERT, for example, was designed to produce sentence embeddings that can be compared efficiently for semantic similarity and clustering tasks [13].

In this study, embeddings are used as an upstream representation tool for endpoint descriptions. The theoretical motivation is that endpoint strings with related clinical meaning should ideally be closer in embedding space than unrelated endpoints, even when their exact wording differs. This supports clustering and semantic organization before supervised prediction.

However, embeddings are not treated as a complete solution. In clinical endpoint data, linguistic similarity and clinical equivalence do not always align. Two endpoints may share a common wording template while measuring different constructs, or may measure the same construct while using different terminology. For this reason, embeddings are combined with terminology signals, hierarchical clustering, and LLM-assisted review in the methodology.

3.3 Clustering and Hierarchical Endpoint Structure

Clustering is an unsupervised learning approach that groups observations according to similarity without requiring predefined labels. In this study, clustering is used to transform heterogeneous secondary-endpoint descriptions into more coherent endpoint groups before prediction.

The final endpoint representation follows a hierarchical structure. Hierarchical clustering refers to clustering approaches that organize observations into nested groups, often represented as a tree-like structure with broader groups above more specific groups [14]. This is suitable for endpoint data because endpoint concepts naturally exist at different levels of granularity. For example, functional-capacity endpoints form a broad family, while six-minute-walk-distance or peak oxygen-consumption endpoints are more specific measurement concepts.

A hierarchical representation reduces the complexity of the prediction problem. Instead of requiring the model to predict only fine-grained endpoint labels directly, the hierarchy allows the endpoint space to be represented from coarse to fine levels. This also improves interpretability: a prediction may still be clinically useful if it identifies the correct broad measurement family, even when it misses the exact fine-grained cluster. Hierarchical classification literature similarly emphasizes that class labels may have parent-child dependencies that should be considered during prediction and evaluation [15].

3.4 Standardized Clinical Terminologies

Clinical trial text often contains heterogeneous terminology. Standardized clinical terminologies reduce this variability by providing controlled identifiers for biomedical,

clinical, protocol-related, and measurement-related concepts.

This study uses three terminology systems:

- **National Cancer Institute Thesaurus (NCIt)**, a biomedical ontology and reference terminology for clinical, biomedical, drug, disease, and research concepts [9].
- **Clinical Data Interchange Standards Consortium (CDISC) Controlled Terminology**, which provides standardized terminology for clinical research data standards and trial-related concepts [10].
- **Logical Observation Identifiers Names and Codes (LOINC)**, a terminology standard for identifying health measurements, observations, laboratory tests, clinical measurements, and documents [11].

The theoretical role of terminology standardization is to connect variable free-text expressions to more stable semantic identifiers. In the pipeline, terminology codes serve two purposes: they support semantic consistency during endpoint organization, and they become structured input features for prediction. At the same time, terminology matching is an approximation. Some mappings may be incomplete, overly broad, or context-dependent, so standardized codes are treated as useful semantic signals rather than as flawless ground truth.

3.5 LLM-Assisted Structured Review

Large language models (LLMs) can be used for structured text-review tasks when the task requires semantic interpretation, comparison of textual alternatives, and generation of constrained outputs. In this study, the LLM is not used to generate endpoint recommendations directly. Instead, it is used as a constrained reviewer during hierarchy construction, where it helps inspect clusters, approve or reject proposed merges, suggest endpoint moves, and improve cluster names.

The theoretical advantage of this design is that the LLM contributes semantic judgement while deterministic code controls the allowed action space. This reduces the risk of treating the model as an unconstrained source of truth. The prompt design therefore follows general prompt-engineering principles: clear task instructions, explicit constraints, structured input separation, and machine-readable output formats [16], [17]. The Methods chapter describes how this was implemented using the project ADK model alias `sonnet-7`.

3.6 Supervised Learning Models

Supervised learning refers to learning a mapping from input features to known targets. In this thesis, the inputs include protocol characteristics, terminology-code features, primary-endpoint information, observed secondary-endpoint context, and candidate-specific features. The targets are either endpoint clusters or candidate endpoint relevance labels.

Several model families are relevant to the methods:

- **Logistic Regression (LR)** is a linear classification model that estimates the probability of a binary outcome from weighted input features. In this study, it functions mainly as a simpler baseline.
- **Random Forest (RF)** is an ensemble of randomized decision trees whose predictions are aggregated across trees, making it useful for nonlinear feature interactions and mixed feature types [18].
- **eXtreme Gradient Boosting (XGBoost)** is a scalable gradient-boosted tree method that builds trees sequentially so that later trees correct earlier errors; it is also designed to handle sparse input efficiently [19].
- **Classifier Chains** are multi-label models that predict labels sequentially and use earlier label predictions as additional input for later predictions, allowing dependencies between labels to be modeled [20].

These models are appropriate for different reasons. Linear models provide interpretable baselines, tree ensembles handle nonlinear relationships, and classifier-chain variants provide a way to model label dependencies. The final model choice is therefore an empirical question, guided by validation performance, ranking behavior, and expert-assessed usefulness.

3.7 Multi-Label and Hierarchical Prediction

In single-label classification, each input belongs to one class. In multi-label classification, an input may be associated with several labels at once. This is appropriate for clinical trial endpoint recommendation because a protocol usually contains multiple complementary endpoints rather than one isolated outcome.

For protocol p, a multi-label target can be written as

$$\mathbf{y}_p = (y_{p,1}, y_{p,2}, \ldots, y_{p,C}), \qquad y_{p,c} \in \{0, 1\},$$

where C is the number of possible labels and $y_{p,c} = 1$ means that label c is relevant for protocol p.

The endpoint labels in this study are also hierarchical. A fine-grained Tier B2 label belongs to a Tier B1 parent, and a Tier B1 label belongs to a broader Tier B0 parent. A simple way to express hierarchical consistency is:

$$y_{p,c}^{(h)} \leq y_{p,\pi(c)}^{(h-1)},$$

where $y_{p,c}^{(h)}$ is a label at hierarchy level h, and $\pi(c)$ is its parent label at the previous level. This means that a child label should not be active unless its parent label is also active.

Hierarchical multi-label prediction is useful here because it reflects the semantic structure of the endpoint space. It also supports more informative error interpretation: predicting the correct parent category but the wrong child cluster is different from predicting a completely unrelated endpoint family.

3.8 Pairwise Candidate Scoring and Ranking

While multi-label classification predicts a set of relevant labels, recommendation often requires ranking candidate items. In a pairwise candidate-scoring formulation, the model evaluates one protocol–candidate pair at a time:

$$f(\mathbf{x}_p, z_c) \rightarrow s_{p,c},$$

where $\mathbf{x}_p$ is the protocol representation, z_c is a candidate cluster or endpoint, and $s_{p,c}$ is a relevance score. Candidates can then be sorted by score to form a ranked recommendation list.

This formulation is useful for endpoint recommendation because it allows the model to compare many possible candidate clusters or endpoints for the same protocol. It also makes it possible to include candidate-specific features, such as candidate hierarchy labels and terminology-code overlap between the protocol context and the candidate endpoint.

For code-overlap features, one simple similarity measure is the Jaccard index:

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|},$$

where A and B are two code sets. In this thesis, such overlap features help represent whether a candidate endpoint shares terminology evidence with the protocol, primary endpoints, or observed secondary endpoints.

3.9 Partial-Information Recommendation and Leave-One-Out Evaluation

The final recommendation setting is based on partial endpoint information. This reflects a realistic protocol-design scenario where some endpoints may already be drafted, while additional endpoint information still needs to be identified.

In leave-one-out evaluation, one known secondary endpoint is hidden from a protocol and the remaining secondary endpoints are used as observed context. If protocol p has secondary endpoints

$$E_p = \{e_1, e_2, \ldots, e_N\},$$

then one endpoint is withheld and the model is evaluated on whether it can recover or rank the withheld endpoint, or its corresponding cluster, highly.

This differs from strict from-scratch prediction. In from-scratch prediction, the model only uses protocol-level and primary-endpoint context. In leave-one-out prediction, the model also receives partial secondary-endpoint context. This makes the task closer to an incremental recommendation setting, where the system suggests what may still be missing from a partially specified endpoint design.

3.10 Evaluation Metrics

Because this thesis includes both classification and ranking tasks, several metric families are needed.

For binary or multi-label classification, precision, recall, and F1-score are central:

$$\text{Precision} = \frac{TP}{TP + FP}, \qquad \text{Recall} = \frac{TP}{TP + FN},$$

$$\text{F1} = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}.$$

Precision measures how many predicted positives are correct, while recall measures how many true positives are recovered. F1 summarizes the balance between them.

For multi-label evaluation, averaging matters. **Micro**-averaging pools decisions across all labels before computing the metric, making it sensitive to common labels. **Macro**-averaging computes the metric separately per label and then averages, giving rare labels more equal influence. **Weighted** averaging adjusts the per-label average by label support. **Sample-wise** metrics compute performance per protocol and then average across protocols.

Other classification metrics capture different error views. **Subset accuracy** requires the entire predicted label set to match exactly, making it strict in multi-label settings. **Hamming loss** measures the fraction of individual label decisions that are incorrect. **ROC-AUC** evaluates how well positive and negative examples are separated across thresholds, while **PR-AUC** focuses on the precision–recall trade-off and is often more informative when positive examples are rare [21], [22]. **Log loss** and the **Brier score** evaluate probabilistic predictions rather than only thresholded class decisions [23].

For ranking, top-k metrics are more appropriate than only thresholded classification metrics. **Hit@**k checks whether at least one correct target appears in the top k recommendations. **Recall@**k measures what fraction of expected targets appears in the top k, and **Precision@**k measures what fraction of the top k recommendations is correct. **Mean reciprocal rank (MRR)** rewards systems that place the first correct item near the top of the ranked list, while **mean average precision (MAP)** summarizes precision across relevant ranks [24].

Finally, **candidate-pool recall** is important for the two-stage pipeline. Stage 2 can only rank endpoints that Stage 1 makes available. Candidate-pool recall therefore measures whether the correct hidden endpoint is present in the Stage 2 candidate pool at all. This separates candidate-generation failure from endpoint-ranking failure, which is essential when interpreting the final two-stage results.

3.11 Offline Evaluation and Expert Review

Most evaluation in this thesis is offline: the models are tested on held-out historical data rather than in a deployed protocol-design system. Offline evaluation is repro-

ducible and suitable for comparing model configurations, but it cannot fully measure clinical usefulness in a real design workflow [25].

This matters because endpoint recommendation is not always uniquely defined. A predicted endpoint may fail exact-match evaluation while still being clinically plausible, or it may identify the right measurement family but use a different historical endpoint formulation. Therefore, the quantitative metrics are complemented by qualitative inspection and expert review. In this thesis, expert review is used to assess whether ranked endpoint recommendations are clinically meaningful and useful beyond exact historical recovery.

4

Methods

The methodology of this study follows the same logic as the problem formulation introduced earlier in Sections 1.2 and 1.3: data collection, semantic structuring, terminology-aware standardization, and prediction are treated as interdependent stages of one integrated pipeline rather than as isolated tasks.

The overall workflow consisted of five main parts. First, a heart-failure-focused ClinicalTrials.gov dataset was constructed and reduced to a cleaner working cohort. Second, endpoint information was transformed from raw free-text descriptions into a reviewed hierarchical semantic structure through iterative clustering and LLM-assisted correction. Third, protocols and endpoints were standardized against established biomedical terminology systems in order to reduce lexical variation, extract clinically meaningful concepts, and improve interoperability. Fourth, several prediction formulations were explored, including from-scratch multi-label prediction, cluster-level prediction under partial endpoint information, and endpoint-level pairwise scoring. Finally, the prediction pipeline was organized into two downstream stages: Stage 1 predicts relevant endpoint clusters, while Stage 2 ranks candidate secondary endpoints within the cluster context supplied by Stage 1.

Where the methodology uses LLM-assisted hierarchy correction, the calls were orchestrated through Agent Development Kit (ADK), which is a framework for building and running AI-agent workflows [26]. In this project, ADK was used within the Evinova project environment to access the internal model alias `sonnet-7`. This alias refers to the Sonnet-class LLM made available through the internal platform, rather than to a separately defined public model name. The same `sonnet-7` alias was used throughout the hierarchy-construction pipeline, including the main hierarchy-review process, outer hierarchical tier construction, and later merge-only postprocessing stage. The model choice and prompt-design rationale are described in more detail in Section 4.4.1.

In line with the discussion in Sections 2.2, 2.3, and 2.4, interpretability and semantic consistency were treated as first-class methodological requirements throughout the work. This meant that the objective was not merely to optimize predictive performance, but to construct representations and models that remained clinically understandable, auditable, and suitable for expert review.

Figure 1.1 should be read from top to bottom. It summarizes the study as a staged transformation of raw protocols into recommendation-ready datasets. The present

chapter follows that same overall logic, but with the individual parts unpacked in greater methodological detail.

4.1 Data Collection, Filtering, and Processing

Before endpoints could be clustered, standardized, or predicted, it was necessary to construct a clinically coherent study cohort and a reproducible processing pipeline. Although ClinicalTrials.gov provides a large amount of protocol information, the raw registry records are not immediately suitable for modeling. Study protocols vary in completeness, phase labeling, recruitment status, endpoint reporting quality, and textual specificity, and many fields are either irrelevant or too inconsistent to be used directly in downstream machine learning.

For that reason, the first methodological part of this study was not modeling, but dataset construction. This part had three purposes: to define the heart-failure-focused search space, to reduce the raw protocol pull into a cleaner working cohort, and to retain a structured subset of protocol information suitable for later semantic structuring, terminology standardization, and prediction. The present section therefore describes not only how protocols were collected, but also why the working dataset took its final form and how the processed protocol information was prepared for the later parts of the pipeline.

4.1.1 Search strategy and tier design

The first part of the pipeline in Figure 1.1 was the construction of a heart-failure-focused ClinicalTrials.gov dataset that was large enough to capture realistic endpoint heterogeneity, but still narrow enough to remain clinically coherent. Study protocols were collected through the platform's expert search interface using progressively broader cardiovascular search tiers. The search tiers were defined using domain terms such as HFrEF, HFpEF, HFmrEF, and New York Heart Association (NYHA) terminology. The search queries for the tiers were as follows:

- **Tier A:** `"HFrEF" OR "HFpEF" OR "HFmrEF"` *(950 resulting protocols)*
- **Tier B:** `"HFrEF" OR "HFpEF" OR "HFmrEF" OR "NYHA"` *(2 966 resulting protocols)*
- **Tier C:** `"HFrEF" OR "HFpEF" OR "HFmrEF" OR "NYHA" OR "heart failure"` *(13 279 resulting protocols)*
- **Tier D:** `"HFrEF" OR "HFpEF" OR "HFmrEF" OR "NYHA" OR "heart failure" OR "cardiovascular"` *(81 577 resulting protocols)*

These tiers were created to balance specificity and coverage. Narrower tiers provided higher topical precision but fewer studies, whereas broader tiers increased coverage at the cost of more noise.

Among these alternatives, Tier B was selected as the main working dataset because it provided the most useful balance between topical precision and dataset coverage.

Compared with Tier A, Tier B retained substantially more protocols and a broader range of endpoint formulations, which was important for clustering and representation learning. Compared with Tiers C and D, it remained sufficiently constrained to avoid the much larger amount of generic cardiovascular noise introduced by broader search terms such as `"heart failure"` and `"cardiovascular"`. Tier B was therefore used as the principal empirical basis for the subsequent reduction, clustering, standardization, and modeling stages.

Although Tier B was used as the principal dataset for training, validation, testing, clustering, standardization, and model development, the broader Tier C query was later retained as an external unseen-data source for final pipeline checks. Tier C was not used for fitting vocabularies, tuning thresholds, selecting models, or constructing the reviewed endpoint hierarchy. It was instead used to test whether the exported pipeline could be applied to protocols outside the final Tier B modeling cohort.

4.1.2 Dataset reduction and retained fields

The raw Tier B collection contained 2 966 protocol JSON files. Before applying the final study-selection criteria, the distribution of phases and recruitment statuses in the raw Tier B set was inspected in order to understand the composition of the retrieved data and to guide the reduction strategy. As shown in Figure 4.1 (top two pie charts), the raw collection contained a large proportion of studies with missing phase information (`NA`) as well as studies outside the final scope of this study, such as Phase I, Early Phase I, and Phase IV trials. The raw dataset also contained studies with a wide range of recruitment statuses, including completed, recruiting, terminated, withdrawn, and unknown-status protocols.

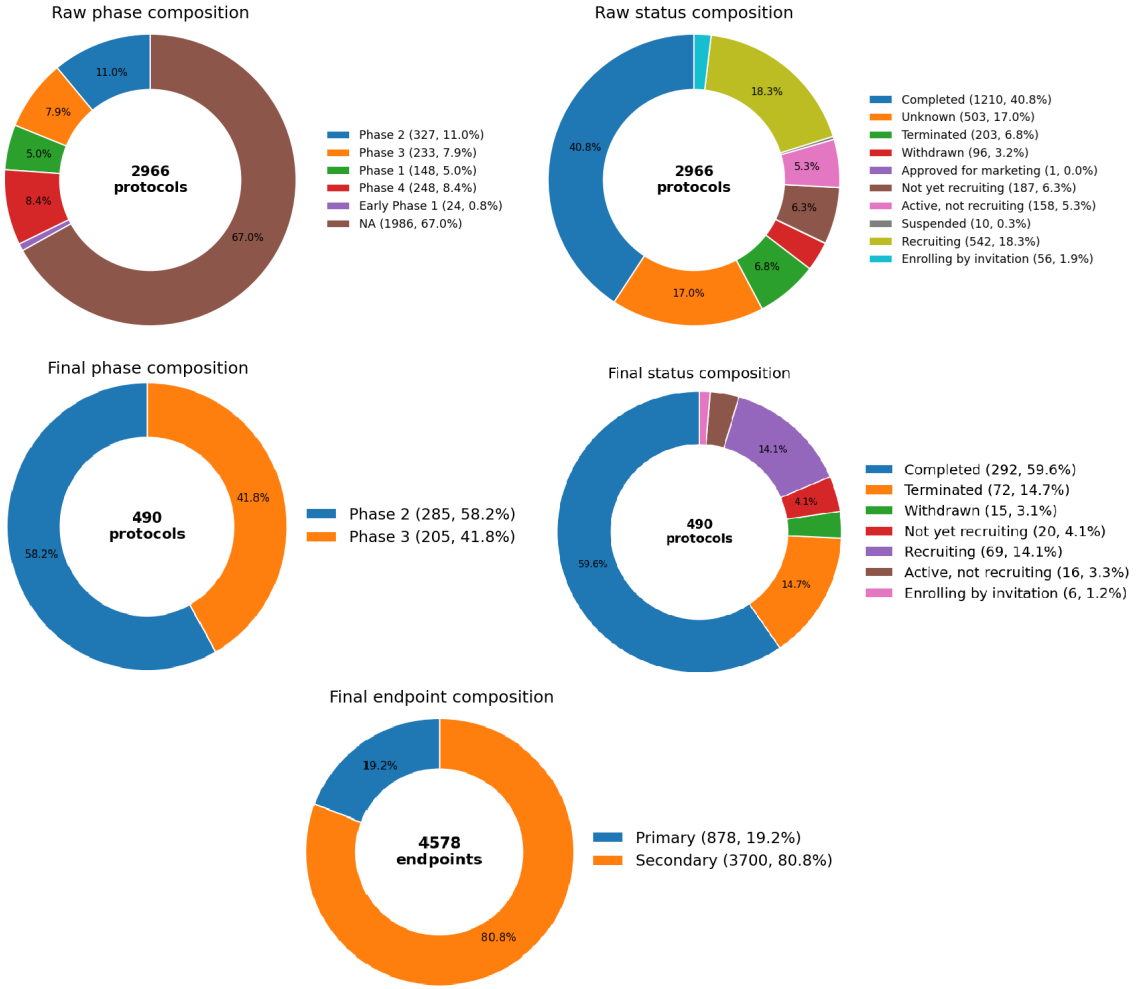


Figure 4.1: Composition of the raw and final Tier B dataset. The figure summarizes phase and recruitment-status distributions before and after reduction, as well as the endpoint composition of the final reduced dataset.

A dedicated reduction script was then used to create a cleaner working dataset. The reduction process preserved a compact subset of clinically relevant fields from each JSON record, including study identification, sponsor information, study status, descriptions, conditions, design characteristics, interventions, endpoints, eligibility criteria, references, Medical Subject Headings (MeSH)-based browse terms, and compact statistical-method summaries when results data were available. Representative examples of the raw ClinicalTrials.gov JSON structure and the reduced protocol JSON structure are provided in Appendix B.2 and Appendix B.3, respectively.

Filtering was performed at the protocol level. The reduction pipeline supported interactive exclusion of phase values and study statuses, optional enforcement of actual completion and enrollment fields, and automatic exclusion of records without primary endpoints. In the officially used Tier B reduction run, the following settings were applied:

- protocols in phases `NA`, `EARLY_PHASE1`, `PHASE1`, and `PHASE4` were excluded,

- only studies containing `PHASE2` and/or `PHASE3` were retained,
- protocols with `overallStatus = UNKNOWN` were excluded,
- protocols with missing or empty primary outcome lists were automatically removed.

This produced a reduced Tier B dataset containing 490 protocols. Among the retained protocols, 285 were labeled with Phase II and 205 with Phase III. Because ClinicalTrials.gov phase labels are not mutually exclusive, some protocols carried both Phase II and Phase III labels in the original registry data. Across the retained protocols, the reduced dataset contained 878 primary endpoints and 3 700 secondary endpoints. This reduction step is important to keep in mind for the remainder of the chapter, since all later clustering, standardization, and modeling stages operate on representations derived from these 490 retained protocols rather than from the full raw registry pull. Examples of the raw and reduced JSON protocol representations are provided in Appendix B.2 and Appendix B.3, while examples of the derived protocol-level and endpoint-level tabular representations are provided in Appendix B.4 and Appendix B.5.

The final filtered dataset composition is also illustrated in Figure 4.1. The final phase distribution shows that the retained protocol set was limited to Phase II and Phase III studies only, while the final recruitment-status distribution shows the remaining status composition after excluding protocols with `overallStatus = UNKNOWN`. The same figure also summarizes the endpoint composition of the final reduced dataset, which contained 878 primary endpoints and 3 700 secondary endpoints. The distribution of the number of secondary endpoints per protocol was highly skewed, ranging from 0 to 49.

4.1.3 Descriptive characteristics of the reduced dataset

Beyond the protocol counts and endpoint counts described above, the reduced Tier B dataset also retained structured protocol-level metadata that was later used to characterize studies and support downstream feature construction. In addition to outcome information, the retained records included eligibility-related fields, arm-group structure, intervention information, study status, phase labels, and descriptive study text.

Eligibility-related metadata was particularly well covered in the final dataset. Core fields such as free-text eligibility criteria, sex restrictions, standardized age groups, and minimum age were available for nearly all retained protocols, whereas more specialized fields such as gender-based eligibility descriptions were rare. This high coverage made eligibility data a useful source of study-population context.

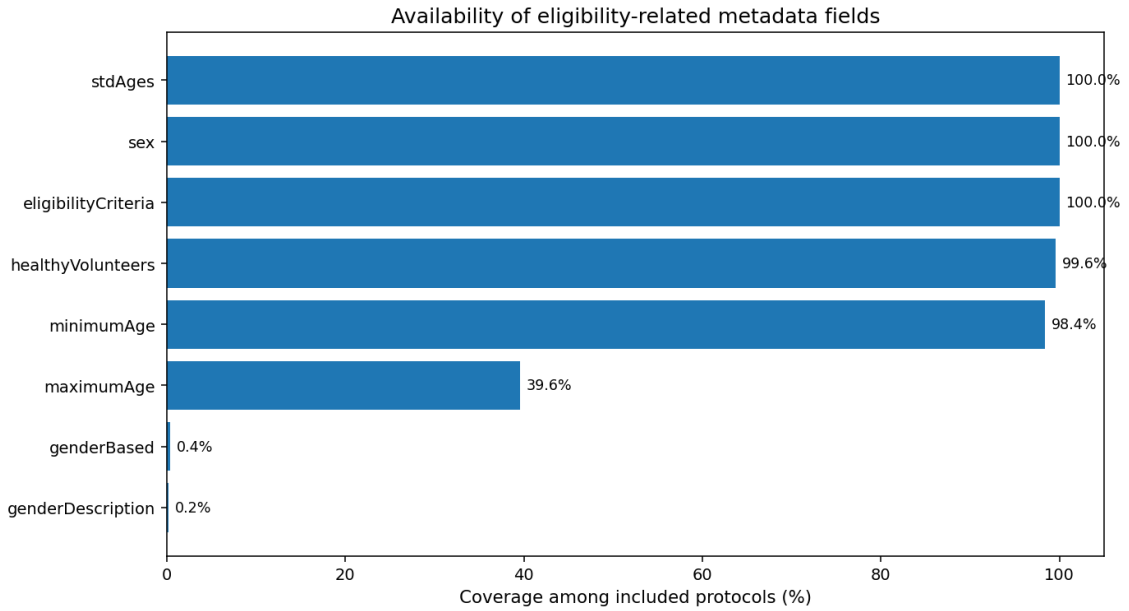


Figure 4.2: Availability of key eligibility-related metadata fields in the reduced Tier B dataset. Core fields such as eligibility criteria text, sex restrictions, standardized age groups, and minimum age were present for most retained protocols, whereas specialized gender-related fields were rare.

Arms and interventions also provided important structural information about study design. In a clinical trial, an arm refers to a group or subgroup of participants that receives a specific intervention, comparator, placebo, sham treatment, or no intervention, according to the study protocol. Interventions describe what is actually assigned or administered within these arms, for example a drug, device, biological product, procedure, behavioral intervention, or other treatment strategy [27].

Most retained protocols contained a small number of arms and interventions, with two-arm and two-intervention studies being the most common. The arm-group metadata was dominated by experimental, placebo-comparator, and active-comparator settings, meaning that many studies compared an investigated treatment against placebo or an established active treatment. Intervention metadata was dominated by drug studies, followed by smaller numbers of device, biological, and procedure-based interventions. These fields therefore provided study-level context about the treatment structure of each protocol, complementing endpoint text and ontology-derived information.

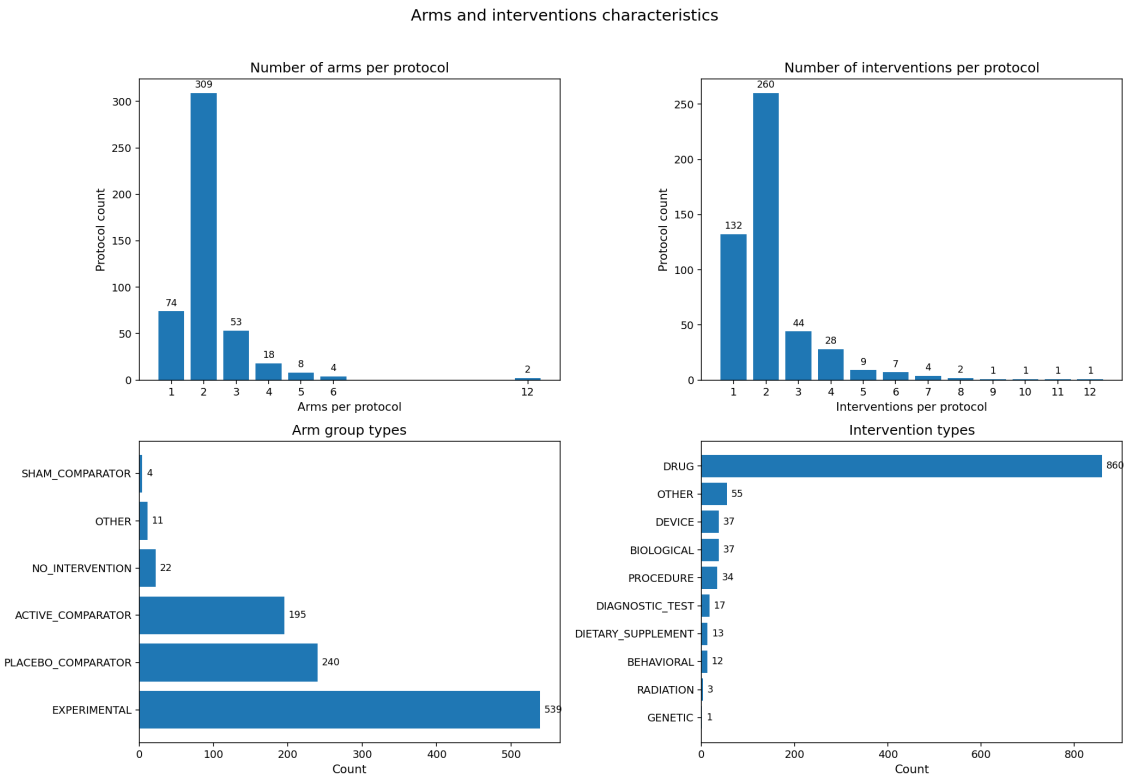


Figure 4.3: Summary of arm-group and intervention characteristics in the reduced Tier B dataset. Top: distributions of the number of arms and interventions per protocol. Bottom: distributions of arm-group types and intervention types. Most retained studies contained a small number of arms and interventions, and the dataset was dominated by experimental arms and drug-based interventions.

Together, these metadata fields helped capture aspects of study population, treatment structure, and trial design that are not fully represented by endpoint text alone. They therefore formed an important complement to the endpoint standardization and clustering pipeline described in the following sections.

4.2 Data Representation

Once the reduced dataset had been created, the next methodological question was how to represent the data for the later stages of the pipeline. The answer was not a single representation, but two complementary ones. This distinction is also reflected in the lower middle part of Figure 1.1, where the dataset is shown as supporting two downstream table views. In other words, two complementary data representations were used throughout the project.

4.2.1 Protocol-level representation

At the protocol level, each row represented one study identified by its `nctId`. Protocol-level rows contained study-design and cohort characteristics such as phase, study

type, allocation, masking, intervention information, enrollment, age restrictions, and other structured variables derived from the reduced JSON files and later standardized tables. Protocol-level rows were used as the main input representation for prediction.

This representation was used in both downstream prediction stages. In Stage 1, one protocol row was combined with primary-endpoint information and, in the leave-one-out setting, partially observed secondary-endpoint information to predict relevant endpoint clusters. In Stage 2, the same protocol representation was combined with candidate endpoint features to score protocol-endpoint pairs. A representative protocol-level row is shown in Appendix B.4.

4.2.2 Endpoint-level representation

At the endpoint level, each row represented one outcome entry extracted from a clinical trial record. The main endpoint fields included the outcome type (`primary` or `secondary`), the original `measure` string, any accompanying description, timeframe information, and protocol identifier. This representation was necessary for semantic clustering, endpoint-specific terminology matching, and later aggregation into protocol-level targets.

This endpoint-level view was required for three purposes. First, it provided the raw material for clustering and hierarchy construction. Second, it provided endpoint-specific terminology codes used as observed secondary-endpoint context in the leave-one-out experiments. Third, it supplied the candidate endpoint catalogue used in Stage 2, where each candidate secondary endpoint could be paired with a protocol and scored as relevant or not relevant. A representative endpoint-level row is shown in Appendix B.5.

The distinction between endpoint-level and protocol-level data was methodologically important. Endpoint-level representations were needed to solve the semantic structuring problem, while protocol-level representations were needed to solve the final recommendation problem. The workflow therefore moved repeatedly between these two views of the data.

4.3 Hierarchical Clustering as the Structural Basis for Recommendation

Before endpoint groupings could be reviewed, standardized, or used as prediction targets, the raw secondary-endpoint descriptions had to be transformed into a more stable and clinically interpretable label space. As discussed in Chapter 1 and Section 2.2, raw endpoint descriptions are highly heterogeneous: similar measurements may be expressed using different wording, abbreviations, timeframes, or levels of clinical specificity. Predicting raw endpoint strings directly would therefore create an unnecessarily fragmented and noisy target space. Clustering was consequently treated as a representation-learning step rather than as an end in itself.

The methodological foundation of the endpoint representation was a *hierarchical*

clustering strategy. This design choice followed directly from the problem formulation introduced in Section 1.2: endpoint recommendation in this setting is difficult not only because of label sparsity, but also because the endpoint space contains meaningful structure at multiple levels of granularity. Some endpoints belong to broad measurement families, while others are highly specific variants of those families. Representing all endpoints at only one level would therefore risk either excessive fragmentation or excessive oversimplification.

4.3.1 Rationale for a hierarchical endpoint representation

The endpoint hierarchy was designed as a *coarse-to-fine representation*. The idea was to first organize endpoints into broader clinical measurement families and then subdivide those families into more specific endpoint clusters. Methodologically, this offered several advantages. First, it made the target space more manageable for downstream prediction, since models would not need to predict a very large number of highly specific endpoint categories directly from the outset. Second, it improved interpretability, because broader parent groups provided a more understandable clinical context for finer subclusters. Third, it created a structure that could later support hierarchical prediction, where coarse predictions constrain or guide finer-grained ones.

The initial hierarchy was constructed in two levels. The first level, **Tier B1**, was intended to capture broad measurement categories, for example functional, biomarker, imaging, safety, or patient-reported measurement families. The second level, **Tier B2**, was intended to capture more specific endpoint concepts nested within those broader categories. In practice, this meant that endpoints were not treated as an unstructured list of labels, but as members of a tree-like organization in which a finer endpoint cluster inherited context from a broader parent category.

The reasoning behind this design was closely tied to the intended recommender-system use case. If the model were required to predict only very fine-grained endpoint clusters, the label space would become large, sparse, and difficult to learn reliably under the small-data conditions described in Section 1.4. By contrast, a hierarchical label space allows the recommendation problem to be decomposed into more feasible steps. At a coarse level, the model can first estimate which broad kinds of measurements are relevant for a protocol. At a finer level, it can then identify more specific endpoint clusters within those relevant regions of the hierarchy.

A further motivation for the hierarchical design was that it provided a natural basis for confidence interpretation and error analysis. If a model predicts the correct broad family but misses the exact fine-grained subcluster, the prediction may still be clinically informative. In other words, hierarchical prediction makes it possible to distinguish between errors that are *structurally close* and errors that are fully unrelated. This is aligned with the evaluation concerns discussed in Section 2.4, where exact label matching alone may not fully capture practical usefulness in biomedical recommendation problems.

In methodological terms, the hierarchy was therefore not only a clustering output

but also a design decision about how the recommendation problem itself should be represented. The hierarchy served two roles simultaneously: it acted as a semantic organization of heterogeneous endpoint language, and it acted as a structural scaffold for the later recommendation models.

4.3.2 Exploratory clustering strategies and embedding setup

Before the final hierarchy was constructed, three exploratory clustering strategies were compared: flat clustering, single-cluster exclusion, and hierarchical clustering. These early experiments used the sentence-transformer model `all-MiniLM-L6-v2` [28], following the clinical-trial semantic-similarity setup discussed in Section 2.5 and described by Majumdar [12]. This smaller model provided a computationally efficient and literature-grounded baseline for evaluating which clustering strategy was most suitable for the endpoint representation task.

The earliest experiments attempted to partition endpoint descriptions directly in a single flat clustering step. In this formulation, all endpoints were assigned to one clustering layer, without first separating broad endpoint families from more specific endpoint variants. Several feature configurations were evaluated, including endpoint-text embeddings, protocol-derived features, and weighted combinations of the two. In particular, the relative weighting between endpoint-derived and protocol-derived information was varied from approximately balanced settings to endpoint-dominant settings, ranging from 50/50 to 90/10 endpoint-protocol feature weighting.

The flat formulation imposed a single granularity level on the full endpoint space. Broad endpoint families, such as mortality, biomarkers, imaging, functional testing, and patient-reported outcomes, had to coexist in the same clustering layer as much more specific measurement variants. This made the representation difficult to stabilize and motivated alternative strategies.

A second exploratory strategy was therefore tested in which visually or semantically separable endpoint groups were extracted one at a time. The motivation was that some endpoint families appeared easier to isolate than others. Instead of forcing the entire endpoint space into one global partition, this approach attempted to first remove the clearest endpoint groups and then recluster the remaining endpoints. Although this produced cleaner clusters in the early extraction steps, it also distorted the remaining endpoint pool. Later extraction rounds produced less coherent groups, and the proportion of endpoints classified as noise increased. This showed that sequentially extracting one cluster at a time was not a stable solution.

These experiments motivated the final hierarchical strategy. Flat clustering forced all endpoint concepts into one global partition, while single-cluster exclusion showed that some broad endpoint families could be separated cleanly but that sequential removal distorted the remaining endpoint space. The hierarchical approach was therefore introduced to preserve useful coarse separation while allowing finer-grained refinement within each broad group. The smaller embedding setup, based on `all-MiniLM-L6-v2`, was motivated by the semantic clinical-trial retrieval pipeline described by Majumdar [12]. However, after the hierarchical strategy had proven

more promising than the flat and single-cluster exclusion approaches, we also tested whether a larger embedding model could produce a stronger semantic representation for the final hierarchy. The final hierarchical clustering pipeline was therefore rerun and refined using `Qwen3-Embedding-8B` [29]. The larger model was applied to the strategy that had already shown the best structure in the smaller-model experiments, rather than being used to repeat every early exploratory configuration.

4.3.3 Pre-LLM hierarchical semantic clustering pipeline

Before introducing the LLM review stage, a semantic clustering pipeline was developed to implement the selected hierarchical strategy. This pipeline was designed around the idea that clustering quality would improve if endpoint strings were first transformed into a cleaner semantic representation and then enriched with terminology-aware structure, consistent with the semantic-similarity perspective discussed in Section 2.5.

The pipeline began with *pattern-based concept extraction*. Rather than relying on manually curated drug lists or hardcoded measurement taxonomies, endpoint texts were cleaned through structural normalization rules. These rules aimed to remove repeated scaffolding such as objective prefixes, timeframe suffixes, and proportion-style sentence templates while preserving the clinically relevant core phrase. The purpose was not to solve the semantic problem completely through rules, but to reduce obvious linguistic noise before embedding-based analysis.

After this preprocessing stage, endpoint concepts were embedded using transformer-based sentence embeddings. The exploratory comparison used `all-MiniLM-L6-v2`, while the final pre-LLM hierarchical clustering pipeline used `Qwen3-Embedding-8B` as the main encoder. The same pipeline also incorporated the standardized terminology mappings described in Section 4.5. During clustering, these mappings supported canonical naming, synonym discovery, broader-concept lookup, and hierarchy navigation. This made it possible to combine dense embedding similarity with structured biomedical knowledge rather than relying on embeddings alone.

A dedicated semantic analyzer was then used to perform several concept-level operations before the actual clustering step. First, it attempted *synonym discovery*, so that semantically equivalent formulations such as abbreviations and full measurement names could be normalized toward a shared representation. Second, it attempted *parent-child discovery*, for example by detecting when a more specific measurement formulation could be interpreted as a subtype of a broader concept. Third, it performed *concept decomposition*, whose goal was to distinguish what was being measured from method, qualifier, or contextual phrasing. The underlying motivation was that cluster names should ultimately reflect the clinical construct being measured rather than the syntactic form in which that construct happened to be described.

The resulting normalized concepts were then clustered hierarchically with agglomerative clustering using cosine distance. In the final pre-LLM version of this pipeline, the hierarchy was constructed with two clustering layers: a broader Tier B1 and a more specific Tier B2. Tier B1 used a looser clustering threshold in order to form broader measurement categories, whereas Tier B2 used a tighter threshold

to preserve more specific distinctions. After this first clustering step, additional merge passes were applied. One merge pass used semantic similarity to combine clusters that appeared to measure the same underlying construct, and another used shared terminology evidence to merge clusters whose members mapped to the same standardized concepts.

The output of this stage was not the final hierarchy used for modeling. It was a preliminary Tier B1–Tier B2 structure without the outer Tier B0 layer. Although this represented a clear improvement over the earlier flat and single-cluster exclusion experiments, the generated hierarchy was still too wide and fragmented for downstream use. It contained more than 150 Tier B1 clusters and more than 500 Tier B2 clusters, including many singleton and two-endpoint clusters. Some of these represented genuinely rare endpoint concepts, but many reflected over-fragmentation caused by wording differences rather than clinically meaningful separation.

The most significant limitation was that the clustering still sometimes grouped endpoints by *surface-form similarity* rather than by the underlying construct being measured. Endpoints expressed with common syntactic templates or shared linguistic patterns could end up together even when they did not represent the same clinical measurement. Conversely, endpoints measuring the same construct could remain separated if their wording or surrounding protocol context differed substantially. Table 4.1 gives illustrative examples from the endpoint catalogue. These examples are not included as an evaluation of individual final clusters, but to show why purely embedding- or wording-based similarity was insufficient and why an additional review stage was needed.

Endpoint issue	**Example `measure` strings**	**Reason for issue**
Similar wording, different measurements (ended up in the same clusters)	• Change From Baseline in Echocardiography Parameters: Ratio of E to A Velocity, E/e' Ratio • Change From Baseline in Echocardiography Parameters: Tricuspid Regurgitation Velocity • Change From Baseline in Echocardiography Parameters: Left Ventricular Mass Index	The shared template and echocardiography wording are similar, but the endpoints measure different cardiac constructs.
Different wording, same measurement construct (ended up in different clusters)	• Peak VO_2 • Change From Baseline in Maximal Oxygen Consumption (VO2max) at Peak Exercise • Functional Capacity (Change in Peak Oxygen Uptake, VO_2)	The wording varies substantially, but the endpoints refer to closely related peak oxygen uptake / VO_2 exercise-capacity measurements.

Table 4.1: Examples of endpoint wording patterns that complicated the pre-LLM clustering stage. Similar surface forms did not always imply the same measurement, while the same measurement construct could be expressed using very different terminology.

These limitations motivated the LLM-assisted review stage described next.

4.4 LLM-Assisted Hierarchy Construction and Review

After the Qwen-based hierarchical clustering stage had produced a preliminary Tier B1–Tier B2 endpoint hierarchy, the hierarchy was refined using a LLM-assisted review pipeline. This stage was introduced because the automatically generated hierarchy, although more useful than the earlier flat clustering attempts, was still too wide, fragmented, and clinically inconsistent for downstream modeling. The purpose of the LLM stage was therefore not to generate the hierarchy from scratch, but to repair and consolidate an already constructed two-tier structure.

The review stage had two related but distinct roles. First, it repaired the existing Tier B1 and Tier B2 structure through iterative endpoint moves, cluster merges, cluster modifications, and renaming. Second, after a reviewed Tier B1–Tier B2 hierarchy had been produced, a coarser outer Tier B0 layer was introduced to group related Tier B1 clusters into broader endpoint families.

Only after a reviewed Tier B1–Tier B2 hierarchy had been produced was a coarser outer Tier B0 layer introduced above Tier B1. Tier B0 was therefore not produced directly by the original embedding-based clustering. Instead, it was added as a practical and methodological navigation layer over the more detailed hierarchy, making the endpoint space easier to inspect, visualize, filter, and use in hierarchical prediction. The Tier B0 construction used summaries derived from all endpoints contained in each Tier B1 cluster, including dominant measures, concepts, terminology-code signatures, and nearest-neighbor relations between Tier B1 groups. A language model was then used to propose Tier B0 groups and assign each Tier B1 cluster exactly once, after which the output was validated and repaired deterministically to guarantee complete Tier B1 coverage, prevent duplicate assignment, and preserve the endpoint set.

The overall hierarchy-development process was iterative. Several alternative Tier B1–Tier B2 hierarchy configurations were produced by applying different amounts of LLM-assisted merge, move, split, update, and rename operations to the same underlying Qwen embedding representation. For each candidate Tier B1–Tier B2 hierarchy, a corresponding Tier B0 layer was generated, and the resulting full hierarchy was then evaluated in the downstream modeling pipeline. The final hierarchy was therefore selected not only because it appeared more coherent during manual inspection and expert review, but also because it gave the strongest downstream modeling behavior among the tested hierarchy configurations. An excerpt of the final reviewed hierarchy is provided in Appendix B.7.

4.4.1 LLM model and prompt design

All LLM-assisted hierarchy-construction steps in this study were run through the project ADK using the model alias `sonnet-7`. This includes the main hierarchy-review process, the construction of the outer Tier B0 layer, and the merge-only postprocessing stage. The alias refers to the Sonnet-class LLM available through

Evinova's internal platform. The exact provider-side model string is therefore treated as an implementation detail, while the reproducible methodological detail is that the hierarchy-review pipeline consistently used the same ADK model alias across the LLM-assisted stages.

A Sonnet-class model was selected because the task required strong instruction following over structured clinical text, the ability to compare multiple endpoint-cluster summaries, and reliable production of machine-readable correction outputs. These requirements applied throughout the entire LLM-assisted hierarchy pipeline, not only to the merge-only stage. In the main review stage, the model was used to assess cluster coherence, suggest endpoint moves, merge over-fragmented clusters, and improve cluster naming. In the Tier B0 stage, it was used to group reviewed Tier B1 clusters into broader endpoint families. In the merge-only stage, it was used to approve or reject residual candidate merges. This is consistent with public documentation describing Claude Sonnet models as suitable for complex reasoning, instruction-following, coding-oriented workflows, and tool-supported tasks [30].

The prompt design was intentionally restrictive across the LLM-assisted stages. Rather than asking the model to freely redesign the hierarchy, each prompt framed the task as a bounded correction problem: inspect the provided hierarchy information, apply only the requested operation type, preserve clinically meaningful distinctions, and return a structured output that could be validated programmatically. This design follows general prompt-engineering recommendations: provide clear task instructions, specify the desired output format, separate instructions from input data, and use structured examples or schemas to improve consistency [16], [17].

The merge-only stage provides the clearest example of this prompt design because its allowed action space was especially narrow. Candidate merge pairs were generated deterministically before the LLM call, and the model was only asked to approve merges that represented true same-construct duplicates. Because these candidate pairs were reviewed in chunks rather than in one single global prompt, the stability of the resulting merge decisions was later assessed through the repeated-run reliability analysis described in Section 4.4.4. An abbreviated version of the Tier B2 merge-only prompt is shown below.

```
Task: Approve ONLY Tier2 merges that are true same-construct duplicates.

Return ONE JSON patch between markers:

BEGIN_PATCH_JSON
{"renames":[],"merges":[],"moves":[]}
END_PATCH_JSON

For this pass:
- Use ONLY "merges" with level=2.
- Allowed merge forms:
  {"level":2,
   "tier1":"...",
   "t2_list":["...","..."],
```

```
   "new_t2_name":"...",
   "confidence":0.0-1.0,
   "reason":"..."}

Constraints:
- You MUST NOT propose merges outside the provided pairs.
- Merge ONLY if they measure the SAME construct.
- Do NOT merge merely related method, safety, biomarker,
  imaging, or physiology clusters.
- Avoid protected distinctions such as LV vs RV,
  systolic vs diastolic, peak vs rest.

CANDIDATE PAIRS JSON:
{"pairs":[...]}
```

The LLM output was not applied directly. Across the LLM-assisted hierarchy pipeline, model outputs were treated as proposed corrections rather than final authority. The scripts parsed the returned output, handled malformed responses through retry or repair logic where applicable, filtered proposed operations using task-specific validity checks, and verified that endpoint assignments were preserved correctly. This made the LLM a constrained reviewer inside a deterministic processing pipeline rather than an unconstrained generator of the final hierarchy.

4.4.2 Hierarchy review pipeline

The main hierarchy-review script operated as a staged correction pipeline over the preliminary Tier B1–Tier B2 hierarchy produced by the Qwen-based clustering stage. The process began with a deterministic auto-move pass that could identify obvious endpoint misplacements using local signal patterns. The script then performed a dedicated Tier B2 audit in which every Tier B2 cluster was inspected for measurement coherence. Clusters judged to be invalid, over-fragmented, or non-measurement-based were flagged, and their endpoints were rehomed to more appropriate destinations across the hierarchy.

This was followed by iterative endpoint-level move rounds, Tier B2 merge rounds, a final Tier B2 merge pass to catch merge opportunities created by earlier moves, Tier B1 merge rounds, and finally a rename stage that proposed more coherent cluster names based on summaries derived from *all* endpoints in each cluster. Together, these steps reduced fragmentation, merged duplicate or near-duplicate concepts, corrected misplaced endpoints, and made the hierarchy more clinically interpretable.

Several robustness mechanisms were built into this process. The pipeline used protected distinction rules to prevent clinically inappropriate merges, chunked review to stay within model context limits (for which an LLM-reliability check was conducted, discussed in Section 4.4.4), parse-failure handling and retry logic for malformed LLM output, and an endpoint-preservation invariant that checked that no endpoints were lost during patch application. In other words, the hierarchy reviewer was not allowed to silently change the endpoint multiset.

4.4.3 Merge-only postprocessing

After the main review stage, an additional merge-only postprocessor was used to further improve cluster coherence without introducing new moves or renames. This script computed an all-vs-all similarity matrix between clusters using signatures derived from *all* endpoints assigned to each cluster. Candidate merges were then sent to the same ADK-based LLM for approval in iterative rounds using the constrained prompt-and-patch design described in Section 4.4.1.

The rationale for this separation was practical. The main review pipeline handled the broader structural problem of endpoint reallocation and cluster repair, whereas the merge-only stage focused specifically on detecting residual duplicate or near-duplicate clusters that remained after the more complex restructuring stages. Like the main reviewer, the merge-only script included strong validation gates and endpoint multiset invariance checks.

4.4.4 Reliability Analysis of the Merge-Only LLM Review

Because the final reviewed endpoint hierarchy was refined using an LLM, an additional reliability analysis was performed to assess whether the resulting late-stage merge behavior was stable across repeated runs, rather than being an artifact of one particularly favorable run. The purpose of this analysis was not to test whether the merge-only stage was fully deterministic, but rather to evaluate whether repeated executions of the same merge-only review procedure produced (i) a stable core of recurring merge decisions and (ii) sufficiently similar final endpoint assignments to support the use of the LLM-reviewed hierarchy in the downstream modeling pipeline. This was particularly relevant because the merge-only script first reduced the hierarchy to deterministic Tier B1 and Tier B2 candidate pairs, and then reviewed those pairs in chunks, meaning that both accepted merge sets and final assignments could potentially vary across repeated LLM calls. More broadly, the analysis provided a limited robustness check for the use of LLM-assisted review in the hierarchy pipeline by testing whether repeated LLM-based refinement produced structurally stable outcomes rather than arbitrary hierarchy changes.

This reliability analysis was performed on the final reviewed hierarchy *before* construction of Tier B0, that is, on the reviewed two-tier hierarchy containing only Tier B1 and Tier B2. The merge-only postprocessor was then rerun five times on the same input hierarchy using identical recorded settings. In this way, the analysis isolated the reproducibility of the *late-stage merge-only review step* rather than the full end-to-end hierarchy-construction pipeline.

For each run, the merge-only postprocessor produced a corrected hierarchy as well as intermediate and diagnostic outputs. These run artifacts were then analyzed automatically using a dedicated comparison script. The analysis focused on four complementary perspectives.

First, **procedural consistency** was checked by comparing the recorded parameter hashes, parse-failure counts, warning files, and final endpoint counts across runs.

This part of the analysis was intended to verify that the repeated runs were in fact comparable and that no run failed because of malformed output or lost data.

Second, **merge reproducibility** was evaluated by comparing the actual merge decisions proposed and accepted across runs. Two levels of merge comparison were used:

- a **decision-level** representation, where the key question was whether the same source clusters were merged across runs regardless of the final merge name, and
- a **full** representation, where both the merged source clusters and the resulting new cluster name had to match.

For a pair of runs r and s, similarity between their merge sets was summarized using the Jaccard index

$$J(r, s) = \frac{|M_r \cap M_s|}{|M_r \cup M_s|},$$

where M_r and M_s denote the merge sets extracted from the two runs. This was computed both for decision-level merges and for full merges.

Third, **final structural stability** was evaluated at the endpoint-assignment level rather than only at the merge-list level. Let $h_r(e)$ denote the final Tier B1-Tier B2 assignment of endpoint e in run r. For each pair of runs r and s, endpoint-assignment agreement was defined as

$$A(r, s) = \frac{1}{|E|} \sum_{e \in E} \mathbf{1}\{h_r(e) = h_s(e)\},$$

where E is the common set of endpoints and $\mathbf{1}\{\cdot\}$ is the indicator function. This measure captures whether two repeated runs ultimately place the same endpoints into the same final hierarchy locations, even when the exact intermediate merge history differs.

Fourth, **stable core decisions** were examined by counting how often each merge decision recurred across the five runs. This made it possible to distinguish between:

- merges that appeared in all or most runs and therefore represent a stable core of the review behavior, and
- merges that appeared only occasionally and therefore represent more marginal, stochastic decisions.

This reliability analysis was focused on the merge-only stage. It should therefore be interpreted primarily as a robustness check on the reproducibility of the final LLM-assisted merge refinement step, and only secondarily as supporting evidence for the broader use of constrained LLM review in the hierarchy pipeline. The corresponding empirical results are presented in Section 5.2.

4.4.5 Iterative hierarchy variants and downstream selection

The final hierarchy was not produced in a single pass. Instead, several hierarchy variants were created and evaluated during development. The underlying endpoint

embeddings were kept fixed, meaning that the main difference between hierarchy variants was not the semantic encoder, but the amount and outcome of LLM-assisted hierarchy refinement. In practice, different variants reflected different numbers or combinations of merge, move, split, update, and rename operations applied to the preliminary Tier B1-Tier B2 structure.

For each reviewed Tier B1-Tier B2 hierarchy variant, a new Tier B0 layer was generated so that the outer hierarchy remained consistent with the current mid-level and fine-grained cluster structure. Each resulting three-level hierarchy was then used in downstream modeling experiments. This made it possible to compare hierarchy configurations not only by manual inspection, but also by their effect on predictive performance.

The final hierarchy used in the modeling pipeline was therefore selected as the most useful configuration among the tested variants. It preserved the same Qwen-based endpoint embedding foundation, but used the LLM-assisted review process to obtain a more compact, coherent, and model-compatible hierarchy.

4.5 Terminology Standardization

Consistent with the discussion in Sections 2.2, 2.3, and 3.4, semantic normalization and interpretability were treated as essential requirements rather than optional refinements. Raw protocol and endpoint language is too variable to serve as a stable foundation for clustering and especially prediction on its own. Standardization against established terminology systems was therefore used to anchor protocol and endpoint concepts to more reproducible biomedical representations.

As introduced in Section 3.4, the standardization pipeline used NCIt, CDISC, and LOINC as complementary terminology systems. In the study methodology, these terminology mappings are treated operationally: not as theoretical resources in themselves, but as structured code features that support clustering, hierarchy review, and downstream prediction.

Standardization served two main roles in the overall pipeline. First, it improved semantic consistency during cluster construction and review by providing structured terminology signals beyond raw text. Second, it created coded feature inputs for the prediction models. In the strict from-scratch setting, protocol-level codes and primary-endpoint codes formed the main standardized input representation. In the leave-one-out setting, codes from the observed secondary endpoints were also included as partial endpoint context, while the withheld endpoint or cluster remained the prediction target. This allowed the model to use already specified endpoint information without leaking the endpoint being evaluated.

Within the full workflow, standardization sits between raw data extraction and the reviewed modeling dataset. It affects both the protocol branch and the endpoint branch, and therefore acts as a convergence stage rather than as an isolated preprocessing detail. Representative standardization outputs for both protocol-level and endpoint-level records are shown in Appendix B.8.

4.5.1 Protocol-level standardization

A dedicated protocol-level standardization pipeline was developed to transform protocol information into a machine-learning-ready coded representation. The pipeline first aggregated the endpoint-level raw data into one row per protocol. It then combined this protocol table with structured variables, target labels, terminology resources, and information enriched from the reduced ClinicalTrials.gov JSON files.

The protocol-level standardizer extracted candidate phrases from several protocol sources, including summaries, eligibility criteria, intervention descriptions, arm descriptions, MeSH terms, drug names and synonyms, and outcome-related text. These sources were used because relevant biomedical concepts may appear in different parts of a trial record rather than in one fixed field. Long text fields were decomposed into medical n-grams, where an n-gram is a contiguous sequence of tokens such as a one-word term, two-word phrase, or longer phrase. This allowed the matcher to recover clinically meaningful expressions embedded inside longer sentences.

Before terminology matching, the extracted candidate phrases were normalized and expanded. This included abbreviation expansion, synonym handling, and terminology-specific enrichment where available. For example, LOINC classes and abbreviation mappings were used to improve the recognition of measurement-related concepts, while NCIt and CDISC resources supported broader biomedical, clinical, protocol, and trial-design terminology.

The resulting candidate phrases were then matched against NCIt, CDISC, and LOINC using a shared exact-token and fuzzy-matching engine. Matching was controlled using thresholds, token-overlap constraints, and per-system code budgets in order to avoid assigning an excessive number of weak terminology matches. The protocol standardization stage also supported excluding endpoint-derived text from protocol-level matching. This was important for prediction experiments where endpoints were used as targets, because it reduced the risk that protocol-level features would leak information from the endpoint being predicted. The output of this stage was a standardized protocol table containing structured protocol variables, terminology code sets, and coverage diagnostics.

4.5.2 Endpoint-level standardization

A second standardization pipeline was applied at the individual endpoint level. In contrast to the protocol-level pipeline, the endpoint-level standardizer used only endpoint-specific fields:

- `measure`,
- `concept`,
- `concept_normalized`,
- `description`,
- `timeframe`.

It did *not* use protocol identifier, study title, conditions, or outcome type during matching. This design decision ensured that endpoint codes reflected the content of the endpoint itself rather than broader study context.

Candidate extraction followed the same general logic as in the protocol-level standardization stage. The measure string was treated as the primary endpoint descriptor, while normalized concept forms, descriptions, timeframe strings, abbreviation expansions, and medical n-grams provided additional candidate phrases. These candidates were then matched to NCIt, CDISC, and LOINC using the same shared matching engine as the protocol-level pipeline. Reusing the same matcher helped keep terminology assignment consistent across protocol-level and endpoint-level records.

4.6 Problem Formulation

Consistent with Section 1.2, the central difficulty was that endpoint recommendation was not only a prediction problem, but also a representation problem. For this reason, the project progressed through several prediction formulations. Each pivot changed what information the model was allowed to condition on, what target it was expected to predict, and how the corresponding dataset had to be constructed. In this sense, the modeling work was also a data-management process: new target definitions, feature sets, and leakage constraints had to be handled whenever the formulation changed.

4.6.1 Evolution of explored prediction formulations

Table 4.2 summarizes the main previous prediction formulations explored during the project. The progression moved from strict cold-start primary-endpoint prediction toward the final secondary-endpoint recommendation setting, where primary endpoints and partially observed secondary endpoints can be used as context.

Stage	Prediction formulation	Purpose and interpretation
Previous	Predict primary endpoints from scratch	Strict cold-start formulation using protocol information alone. This was the most ambitious setting, but exact primary-endpoint recovery was too difficult because primary endpoints are highly specific and the available protocol representation did not provide enough signal.
Previous	Predict primary endpoints given cluster information	Tested whether the endpoint hierarchy could make the prediction target more learnable by replacing raw endpoint strings with structured cluster information. This showed that endpoint structure was useful, but the task still remained difficult and was not yet aligned with the final secondary-endpoint recommendation use case.
Previous	Predict primary endpoints given endpoint-code subsets	Tested whether standardized endpoint-code information could provide useful partial endpoint context. This formulation helped motivate later use of terminology codes, but it still focused on primary endpoints rather than the final secondary-endpoint completion task.
Current	Predict secondary endpoints given primary endpoints	From-scratch secondary-endpoint formulation. Here, primary endpoints are treated as known context, while the model predicts the secondary-endpoint space. This better reflects protocol design, where primary endpoints are often defined earlier than secondary endpoints.
Current	Predict secondary endpoints given primary endpoints and partial secondary information	Final leave-one-out formulation. Some secondary endpoints are treated as already observed, and the model attempts to recover the missing endpoint information. This simulates an incremental protocol-completion workflow where the system recommends what may still be missing.

Table 4.2: Evolution of the prediction task during the study. Red stages indicate earlier primary-endpoint formulations, while green stages indicate the current secondary-endpoint recommendation formulations. The progression moved from strict cold-start primary-endpoint prediction toward secondary-endpoint recommendation trained on primary endpoints and partially observed secondary-endpoint context.

The table should be interpreted as a progression of modeling assumptions rather than as a runtime pipeline. The earlier formulations were useful diagnostically because they showed the limits of strict from-scratch prediction. The later formulations were more aligned with the intended recommender-system use case, where the model supports an incremental protocol-design workflow rather than predicting all endpoints from a blank slate.

Within the current secondary-endpoint setting, both direct multilabel prediction and

pairwise candidate scoring were evaluated. Pairwise scoring was ultimately more aligned with the recommendation objective because it allowed candidate ranking, top-k evaluation, and hierarchy-aware hard-negative sampling.

4.6.2 Final two-stage recommendation formulation

After the prediction problem had shifted toward secondary-endpoint recommendation, the final implementation was organized as a two-stage ranking workflow. The final formulation focused on **secondary-endpoint recommendation conditioned on protocol information, primary endpoints, and partially observed secondary-endpoint context**. This setting was chosen because it more closely resembles a realistic protocol-design workflow. In practice, a study team may already have drafted some endpoints, while other relevant endpoints remain unspecified. The model should therefore not only predict from a blank protocol, but also use the partially available endpoint design to recommend what may still be missing.

The final exported pipeline therefore does not treat endpoint recommendation as a single flat multilabel classification task. Instead, it uses the reviewed endpoint hierarchy as a candidate-generation and ranking structure: Stage 1 identifies likely endpoint regions, and Stage 2 ranks concrete endpoints within the candidate space induced by those regions. The pairwise formulation therefore allowed the system to answer a practical ranking question: given a partially specified endpoint design, which remaining cluster or endpoint should be ranked highly? This made the final formulation naturally compatible with top-k recommendation metrics, candidate-pool inspection, and qualitative expert review.

4.7 Construction of Modeling Inputs and Targets

This section describes how the reviewed endpoint hierarchy, standardized terminology codes, and protocol-level representations were converted into supervised modeling datasets. The construction differs between the cluster-level Stage 1 task and the endpoint-level Stage 2 task, but both are built from the same underlying principle: a protocol provides a fixed base representation, while secondary endpoints define either observed context, withheld targets, or candidate items to be ranked.

4.7.1 Prediction cohort and target rationale

The prediction cohort follows directly from the final problem formulation described in Section 4.6.1. Primary endpoints were used mainly as contextual input, while secondary endpoints formed the main recommendation target. This choice was also practical; the retained protocols contained approximately 1.8 primary endpoints per protocol, compared with approximately 7.6 secondary endpoints per protocol. A leave-one-out formulation is therefore not well suited to primary endpoints at this scale, since many protocols have too few primary endpoints to create meaningful observed and withheld endpoint subsets. Secondary endpoints provide a richer within-protocol endpoint set, making it possible to simulate partially observed

endpoint designs and ask the model to recover missing endpoint information. The final prediction datasets were therefore constructed from the subset of protocols that contained secondary-endpoint information. This resulted in 444 protocols with secondary endpoints, which were used for both the cluster-level Stage 1 task and the endpoint-level Stage 2 task.

The reviewed endpoint hierarchy provided the supervised target space for Stage 1 and the candidate endpoint catalogue for Stage 2. Before supervised Stage 1 training, a minimum-support filter with `min_label_support = 3` was applied to remove clusters with fewer than three positive protocols. This reduced the effective supervised label space while preserving hierarchical parent-child consistency between Tier B0, Tier B1, and Tier B2.

4.7.2 Base protocol and primary-endpoint features

Both prediction stages used the same base representation for each protocol. This representation contained protocol-level structured variables, protocol-level terminology codes, primary-endpoint terminology codes, and simple numeric aggregates. The base features therefore described the trial context and the already defined primary endpoint strategy before any secondary-endpoint context was added.

The base feature space consisted of:

- structured numeric and boolean protocol variables derived from the reduced and standardized protocol tables,
- protocol-level NCIt, CDISC, and LOINC code features,
- primary-endpoint NCIt, CDISC, and LOINC code features,
- numeric aggregates such as the number of primary endpoints and the number of codes per terminology system.

All free-text protocol fields were excluded from the modeling feature matrix. Standardized terminology codes were represented as sparse binary features, and the code vocabulary was fitted only on the training split.

4.7.3 Leave-one-out endpoint context

Let protocol p have a set of secondary endpoints

$$E_p = \{e_1, e_2, \ldots, e_N\}.$$

In the leave-one-out setting, one secondary endpoint was withheld and the remaining endpoints were treated as observed. More generally, for a leave-k-out sample, the endpoint set was partitioned into an observed subset O_p and a withheld subset H_p, where

$$O_p \cup H_p = E_p, \qquad O_p \cap H_p = \emptyset.$$

The final experiments used $k = 1$, so H_p contained one withheld secondary endpoint. The implementation also supported leave-more-out settings with $k > 1$, which allowed

the same construction to study how performance changed as less secondary-endpoint context was available. The main final setting, however, remained leave-one-out.

The observed endpoints were converted into partial endpoint-context features. These consisted primarily of the union of their standardized secondary-endpoint codes and, in some Stage 1 experiments, the set of already observed cluster labels. The withheld endpoint was not included in the observed feature representation. It was used only to define the target.

As a concrete example, consider one protocol p_1 with five secondary endpoints:

$$E_{p_1} = \{se_1, se_2, se_3, se_4, se_5\}.$$

A leave-one-out construction creates one partially observed sample for each possible withheld endpoint. The observed subsets are:

$$[se_1, se_2, se_3, se_4], \quad [se_1, se_2, se_3, se_5], \quad [se_1, se_2, se_4, se_5],$$

$$[se_1, se_3, se_4, se_5], \quad [se_2, se_3, se_4, se_5].$$

Each subset forms a different protocol context. The protocol-level and primary-endpoint features remain fixed, while the observed secondary-endpoint code bag changes depending on which endpoint is withheld.

4.7.4 Stage 1: cluster-level target construction

Stage 1 used the reviewed endpoint hierarchy to define cluster-level targets. Let $\ell_h(e)$ denote the hierarchy label of endpoint e at tier h, where $h \in \{0, 1, 2\}$. For a protocol p, the observed and withheld label sets at tier h were defined as

$$L_h(O_p) = \{\ell_h(e) : e \in O_p\},$$

and

$$L_h(H_p) = \{\ell_h(e) : e \in H_p\}.$$

Two cluster-target definitions were explored. In the `full` target mode, the target was the full protocol-level cluster set:

$$Y_{p,h}^{\text{full}} = L_h(E_p).$$

This formulation asks whether partial endpoint information can help reconstruct the complete cluster profile of the protocol.

In the `missing` target mode, the target was only the cluster information present in the withheld endpoint and not already represented by the observed endpoints:

$$Y_{p,h}^{\text{missing}} = L_h(H_p) \setminus L_h(O_p).$$

This formulation is closer to the intended recommendation question: given the endpoints already specified for a protocol, which additional endpoint cluster is still missing?

For example, assume that the five endpoints of protocol p_1 together define the full Tier B1 cluster set

$$Y(p_1) = \{c_a, c_b, c_c, c_d\}.$$

In the partial-information multilabel formulation, each observed subset produced one training row,

$$(p_1, \text{observed code bag}) \rightarrow Y(p_1) = \{c_a, c_b, c_c, c_d\}.$$

The target remained the full protocol-level cluster vector, while only the observed secondary-endpoint code bag varied across rows.

The final Stage 1 formulation used the `missing` target mode, because this most directly represents the task of identifying cluster information that is absent from the current partial endpoint design.

4.7.5 Stage 1: pairwise cluster-scoring rows

The final Stage 1 model used a pairwise candidate-scoring format. Instead of predicting all clusters in one multilabel output vector, each training row represented one leave-one-out sample and one candidate cluster:

$$(p, O_p, c),$$

where p is the protocol, O_p is the observed secondary-endpoint subset, and c is a candidate cluster at the selected hierarchy tier.

For one observed context from protocol p_1, candidate clusters $c_1, \ldots, c_m$ were expanded into rows of the form

$$(p_1, \text{observed code bag}, c_1),\ (p_1, \text{observed code bag}, c_2),\ \ldots,\ (p_1, \text{observed code bag}, c_m).$$

The binary target was

$$y^{(1)}_{p,O,c} = \begin{cases} 1, & \text{if } c \in Y^{\text{missing}}_{p,h}, \\ 0, & \text{otherwise.} \end{cases}$$

For the `missing` target mode, candidate clusters that were already represented in the observed endpoint subset were excluded from the candidate pool. This prevented the model from being rewarded for recommending information already present in the partial endpoint design.

Each Stage 1 pairwise row contained base protocol and primary-endpoint features, observed secondary-endpoint code features, a one-hot representation of the candidate cluster, and overlap features comparing the observed secondary-endpoint codes with a training-derived prototype for the candidate cluster.

For each candidate cluster c, a prototype code set was constructed from training protocols only:

$$P_c = \bigcup_{e \in \mathcal{E}^{\text{train}}_c} C(e),$$

where $C(e)$ is the standardized code set of endpoint e, and $\mathcal{E}_c^{\text{train}}$ denotes training endpoints assigned to cluster c. The overlap between the observed endpoint code set and P_c was represented using count, Jaccard similarity, observed-coverage, and prototype-coverage features.

4.7.6 Stage 2: endpoint-level target construction

Stage 2 used the endpoint-level analogue of the Stage 1 pairwise construction. Instead of pairing a protocol context with a candidate cluster, each row paired the protocol context with one concrete candidate secondary endpoint:

$$(p, O_p, e_c),$$

where e_c is a candidate endpoint from the endpoint catalogue.

For one observed context from protocol p_1, candidate endpoints $e_1^{cand}, \ldots, e_m^{cand}$ were expanded into rows of the form

$$\begin{aligned}
&(p_1, \text{observed code bag}, e_1^{cand}),\\
&(p_1, \text{observed code bag}, e_2^{cand}),\\
&\quad\vdots\\
&(p_1, \text{observed code bag}, e_m^{cand}).
\end{aligned}$$

In the from-scratch Stage 2 baseline, a candidate endpoint was labeled positive if it was one of the true secondary endpoints of the protocol. In the final leave-one-out Stage 2 setting, a candidate endpoint was labeled positive if it corresponded to the withheld endpoint:

$$y^{(2)}_{p,O,e_c} = \begin{cases} 1, & \text{if } e_c \in H_p,\\ 0, & \text{otherwise.} \end{cases}$$

This turned endpoint recommendation into a binary relevance-ranking problem. Stage 1 operates at the cluster level and identifies likely endpoint regions. Stage 2 then scores concrete candidate endpoints within the candidate space and ranks them by predicted probability.

4.7.7 Stage 2 candidate and negative construction

For Stage 2 training, positive protocol-endpoint pairs were paired with negative candidate endpoints. A positive row represented a candidate endpoint that belonged to the true secondary endpoint set of the protocol, or, in the leave-one-out setting, to the withheld endpoint set. A negative row represented a candidate endpoint that was not the recorded target endpoint for that protocol context, although it could still be clinically plausible or semantically similar.

Negative candidates were sampled from increasingly broader regions of the reviewed endpoint hierarchy:

- endpoints from the same Tier B2 cluster but not present in the protocol,
- endpoints from other Tier B2 clusters under the same Tier B1 parent,
- endpoints from other Tier B2 clusters under the same Tier B0 parent,
- optionally, endpoints sampled globally from the full endpoint catalogue.

This hierarchy-aware hard-negative strategy was used because clinically difficult errors are often local. A wrong endpoint from the same or nearby cluster may be much harder to distinguish from the true endpoint than a completely unrelated endpoint. Sampling local negatives therefore created a more realistic training problem than using only easy random negatives.

The number of negative candidates per positive endpoint was treated as a dataset-construction parameter. Smaller values create a simpler pairwise classification problem, while larger values expose the model to a broader and more deployment-like candidate space. In the final Stage 2 construction, each positive endpoint pair was expanded with 20 negative candidates. These negatives followed a hierarchy-aware sampling distribution: 80% same Tier B2, 15% same Tier B1, and 5% same Tier B0. This configuration emphasized hard local distinctions while still including some broader alternatives from the surrounding hierarchy.

Sampled versus expanded candidate-pool evaluation. The sampled hard-negative construction above was also used as one evaluation view, measuring whether the model could distinguish the true endpoint from a controlled set of difficult alternatives. In this view, each positive endpoint was evaluated against the fixed sampled-negative structure used during training.

A second evaluation view used an expanded candidate pool. Instead of ranking the hidden endpoint only against the sampled negatives, Stage 2 ranked a substantially larger set of concrete candidate endpoints. This pool was constructed by expanding candidate Tier B2 clusters into all endpoints contained in those clusters. It therefore more closely resembled the intended prediction setting, where Stage 1 may return multiple plausible clusters and Stage 2 must prioritize endpoints across the resulting candidate space.

In the leave-one-out setting, endpoints already observed in the partial protocol context were removed from the expanded pool. This ensured that the model was evaluated on its ability to recover missing endpoints rather than repeat endpoints that were already present.

4.7.8 Vocabulary fitting and code pruning

Binary code vocabularies were fitted exclusively on the training split. This applied to protocol codes, primary-endpoint codes, observed secondary-endpoint codes, and candidate endpoint codes. Codes could be pruned based on minimum document frequency and maximum allowed prevalence. This reduced very rare code noise and, where configured, overly common protocol-derived codes.

In Stage 1, source-aware pruning was used so that protocol, primary-endpoint, and secondary-endpoint codes could have different minimum-frequency requirements. In Stage 2, the feature builder similarly fitted train-only code vocabularies and transformed validation, test, and inference rows using the training-derived vocabulary.

4.8 Label Filtering and Split Strategy

As introduced in Section 4.7.1, Stage 1 used a minimum-support threshold of `min_label_support = 3`. This filtering was applied only to the supervised cluster-prediction targets, not to the full reviewed endpoint hierarchy. Filtering was propagated hierarchically: a Tier B1 label was retained only if its Tier B0 parent was retained, and a Tier B2 label was retained only if its Tier B1 parent was retained. The full hierarchy was still retained as the semantic endpoint catalogue used by the broader pipeline and by Stage 2 candidate construction.

4.8.1 Protocol-level splitting and cross-validation

All splits were performed at the protocol level. This was essential because each protocol could generate many endpoint rows, cluster rows, and leave-one-out samples. Row-level splitting would therefore have created direct leakage between training and evaluation.

The final hold-out design used:

- 70% training protocols,
- 15% validation protocols,
- 15% test protocols.

In addition to the main train-validation-test split, optional group-based cross-validation experiments were run for selected configurations. These experiments used protocol IDs as grouping variables, ensuring that all rows derived from the same protocol remained in the same fold. Cross-validation was used as a stability check rather than as a replacement for the final held-out test evaluation.

4.9 Stage 1: Cluster-Level Candidate Prediction

Stage 1 was responsible for predicting relevant endpoint clusters. Several Stage 1 variants were explored, including from-scratch multi-label prediction, partial-information multi-label prediction, and pairwise candidate-cluster scoring. The final Stage 1 formulation used pairwise leave-one-out cluster scoring, because it most directly represented the practical question of which additional cluster is missing from a partially specified endpoint set.

4.9.1 From-scratch multi-label baseline

This formulation did not use any secondary-endpoint information as input. It was useful as a cold-start baseline, but it had to predict a large sparse label vector without knowing anything about the already drafted secondary endpoints.

Models were trained separately for Tier B0, Tier B1, and Tier B2. Prediction proceeded from coarse to fine labels, and hierarchical consistency could be imposed by masking child labels whose parent label was not active.

4.9.2 Hierarchical support variants

To study the value of hierarchy information, three support settings were evaluated:

- **No support** (`Prev = None`): the model received no previous-tier cluster information.
- **Predicted support** (`Prev = Pred`): lower-tier models received predicted higher-tier labels from the upstream model.
- **True support** (`Prev = True`): lower-tier models received true higher-tier labels.

The `Prev = True` setting is not deployable because true higher-tier labels are not available for new protocols. It was therefore interpreted as an oracle upper bound on the value of perfect hierarchical information. The `Prev = Pred` setting more closely reflects realistic deployment, but it can suffer from error propagation.

4.9.3 Pairwise leave-one-out cluster scoring

The final Stage 1 experiments used the pairwise candidate rows defined in Section 4.7.5. In this formulation, the model scored one candidate cluster at a time and estimated whether that cluster represented missing endpoint information for the current protocol context.

This reformulation was selected because the intended output of Stage 1 was not a complete endpoint design, but a ranked set of plausible cluster regions to pass to Stage 2. Compared with direct multilabel prediction, pairwise scoring made the output easier to rank, reduced dependence on global thresholding, and aligned better with the final candidate-generation role of Stage 1.

4.9.4 Model families and training

The explored model families included Random Forest (RF), eXtreme Gradient Boosting (XGBoost), logistic regression (LR), and classifier-chain (CC) variants. The final Stage 1 pairwise experiments focused primarily on RF and XGBoost. XGBoost was selected for the final exported Stage 1 model because it provided the best practical balance between ranking quality, generalization, and usable top-k cluster outputs.

For pairwise Stage 1 training, class imbalance was handled using inverse-frequency sample weighting. In the final inverse-frequency variant, positive rows belonging to rarer candidate clusters received additional weight. This was introduced to reduce the tendency to over-concentrate on highly prevalent endpoint clusters and to improve the usefulness of the ranked candidate clusters passed to Stage 2.

4.10 Stage 2: Endpoint-Level Candidate Scoring

Stage 2 was responsible for ranking concrete secondary endpoints. In the final exported pipeline, it operated after Stage 1 had narrowed the search space to a set of likely Tier B2 clusters. In the standalone Stage 2 experiments, the same endpoint-scoring formulation was also evaluated under from-scratch and leave-one-out settings. Each Stage 2 row represented a candidate match between a protocol context and one candidate secondary endpoint.

The Stage 2 pairwise construction described in Section 4.7.6 was evaluated in both from-scratch and leave-one-out settings. In the from-scratch variant, candidate endpoints were paired with a protocol without observed secondary-endpoint context. In the leave-one-out variant, one secondary endpoint was hidden and the remaining secondary endpoints were used as observed context. The leave-one-out setting was used as the final Stage 2 formulation because it best matched the intended use case of recommending missing secondary endpoints from a partially specified endpoint design.

4.10.1 Stage 2 pairwise dataset and feature construction

The Stage 2 dataset builder converted each protocol context into protocol–candidate-endpoint rows. Each row contained a protocol identifier, a candidate endpoint identifier, and a binary target `FIT`, indicating whether the candidate endpoint was relevant for that protocol context.

The feature matrix combined protocol-level, primary-endpoint, candidate-endpoint, and pairwise compatibility information. The main feature groups were:

- structured protocol variables,
- binary protocol-code features,
- binary primary-endpoint-code features,
- binary candidate-endpoint-code features,
- candidate Tier B0, Tier B1, and Tier B2 one-hot features,
- pairwise code-overlap features between protocol codes and candidate endpoint codes,
- pairwise code-overlap features between primary-endpoint codes and candidate endpoint codes,

- in the leave-one-out setting, observed secondary-endpoint code features and overlap features between observed secondary-endpoint codes and candidate endpoint codes,
- flags indicating whether the candidate endpoint or its hierarchy parents were already represented in the observed endpoint subset.

In the leave-one-out setting, the row also retained metadata identifying the observed and withheld endpoint subsets, such as observed endpoint IDs, withheld endpoint IDs, and observed hierarchy IDs. These fields were used for sample tracking, target construction, and evaluation, but the withheld endpoint content itself was not included as an input feature.

The overlap features included overlap count, Jaccard similarity, candidate-code coverage, and source-code coverage. These features were included because a candidate endpoint may be more plausible when its terminology-code profile overlaps with the protocol context, the primary-endpoint evidence strategy, or the already observed secondary endpoints.

4.10.2 Stage 2 model selection and export

Stage 2 model selection was performed on the validation split. RF and XGBoost configurations were evaluated across negative-sampling and code-vectorization settings. Thresholds were optimized on validation data only, and the final test split was used once after configuration selection.

The selected model was exported as a `joblib` bundle together with the fitted feature builder, selected feature names, threshold, endpoint catalogue, dataset metadata, and schema information required for inference. The exported bundle also included the required standardized-code input schema, making it possible to apply the model to new ClinicalTrials.gov JSON protocols after external standardization of protocol and endpoint codes.

4.10.3 Full-pipeline inference on unseen protocols

After Stage 1 and Stage 2 had been trained and exported, the two models were connected into a full endpoint-recommendation pipeline. The purpose of this full-pipeline setup was to evaluate how the system behaves when candidate generation and endpoint ranking are performed end to end rather than in isolation.

The full-pipeline procedure was applied to selected protocols from the broader Tier C dataset, which were not included in Tier B and therefore had not been used for model fitting, threshold tuning, vocabulary construction, hierarchy construction, or validation-based model selection. This made Tier C useful as an external unseen-data check of the exported pipeline.

For each unseen protocol, the same preprocessing and standardization logic was applied as in the main pipeline. In the leave-one-out setting, one secondary endpoint

was hidden from the protocol, while the remaining secondary endpoints were retained as observed endpoint context. The inference procedure was:

1. standardize the unseen protocol and its endpoint information,
2. hide one secondary endpoint and retain the remaining secondary endpoints as observed context,
3. use Stage 1 to score candidate Tier B2 endpoint clusters,
4. select a candidate cluster set using a thresholded-then-top-k strategy,
5. expand the selected Tier B2 clusters into concrete endpoint candidates,
6. use Stage 2 to score all remaining protocol–endpoint pairs,
7. rank candidate endpoints by predicted score.

In the final full-pipeline check, Stage 1 retains up to 25 Tier B2 clusters that are later passed to Stage 2 for candidate-pool construction. Stage 2 then ranks all endpoint candidates derived from those clusters. This setup was intentionally stricter than isolated Stage 2 evaluation, because Stage 2 could only rank endpoints made available by Stage 1. Therefore, full-pipeline performance depended on both Stage 1 candidate-pool recall and Stage 2 endpoint-ranking quality.

4.11 Experimental Design and Evaluation

Different metrics were used for the different prediction formulations. For multi-label cluster-prediction baselines, the evaluation used standard multi-label metrics:

- micro-F1,
- macro-F1,
- weighted F1,
- sample-wise F1,
- micro-precision and micro-recall,
- hamming loss,
- subset accuracy,
- exact-match rate,
- average false negatives and false positives per protocol.

For pairwise Stage 1 and Stage 2 models, the row-level binary classification metrics included:

- pair accuracy,
- pair precision,
- pair recall,

- pair F1,
- Area Under the Receiver Operating Characteristic Curve (ROC-AUC),
- Area Under the Precision-Recall Curve (PR-AUC),
- log loss and Brier score where applicable.

Because the final system is a recommender system rather than only a classifier, ranking metrics were especially important. The main ranking-oriented metrics included:

- Hit@k,
- Recall@k,
- Precision@k,
- mean reciprocal rank (MRR),
- mean average precision (MAP),
- candidate-pool recall.

Hit@k measured whether at least one true target appeared among the top k recommendations. Recall@k measured the fraction of expected target endpoints recovered in the top k. Candidate-pool recall measured whether the correct hidden endpoint was present in the candidate pool at all, which is important because Stage 2 can only rank endpoints that Stage 1 has made available.

4.11.1 Exact-match and qualitative endpoint evaluation

Endpoint-level evaluation was based primarily on exact recovery of recorded historical endpoints. In the automated evaluation, a candidate endpoint was counted as correct only if it matched the hidden endpoint identifier or belonged to the expected withheld target set. This made the evaluation reproducible, but also strict.

This strictness is important because endpoint recommendation is not always uniquely defined. Two endpoint descriptions may measure the same clinical concept while differing in wording, time frame, population, intervention context, or source protocol. Conversely, a predicted endpoint may be clinically plausible for the target protocol even if it was not the exact endpoint recorded in the historical data. Exact-match metrics therefore measure recovery of the dataset endpoint, not the full clinical usefulness of the recommendation.

For this reason, quantitative metrics were complemented by qualitative inspection and expert review. These reviews examined whether predicted endpoints were clinically meaningful, whether they belonged to the correct endpoint domain, whether they added new information beyond the observed endpoint context, and whether they were specific enough for the target protocol. This was especially important for the final full-pipeline evaluation, where the ranked endpoint candidates were inherited from historical protocols and could require adaptation before being usable as protocol-specific endpoint suggestions.

4.11.2 Model selection

Model selection was primarily guided by validation performance, but the final choice was not based on a single validation score alone. For the pairwise Stage 2 model, deployment-shaped ranking behavior was prioritized because it most closely reflected the intended use case. Validation metrics such as Recall@10, Hit@5, Recall@20, MRR, PR-AUC, and pair F1 were considered together with overfitting gaps, candidate-pool size, and qualitative expert-review feedback.

This avoided selecting a model only because it performed well on sampled-negative binary classification. A model can separate sampled pairs effectively while still producing less useful rankings in a larger deployment-style candidate pool. The final selection therefore emphasized practical recommendation quality and expert-assessed usefulness, rather than only thresholded row-level classification performance.

4.11.3 Diagnostics and explainability

Several diagnostic artifacts were produced to support interpretation of the results. These included:

- label-support summaries across train, validation, and test splits,
- train/validation/test metric tables,
- threshold curves,
- top-k prediction tables,
- per-protocol and per-sample audit tables,
- confusion matrices and score histograms,
- false-positive and false-negative summaries,
- feature-importance summaries.

For Stage 2, during the evaluation, additional human-readable side-by-side outputs were generated to compare hidden endpoints with the highest-ranked predicted endpoints. This was important because exact endpoint matching can be too strict: a prediction may miss the exact endpoint string but still be clinically close in measurement type, domain, timeframe, or description.

4.12 Methodological Limitations and Threats to Validity

Although the proposed pipeline was designed to be systematic, terminology-aware, and auditable, several limitations affect how the results should be interpreted.

4.12.1 Dataset scope, transferability, and registry-derived data quality

The empirical study is restricted to a reduced Tier B cohort of heart-failure-related ClinicalTrials.gov protocols and to the Phase II–III subset retained after the filtering procedure described in Section 4.1. This improves topical coherence, but it also narrows the scope of the conclusions. As also noted in Section 1.4, the pipeline itself is not hardcoded to heart failure, but transfer to another therapeutic area would require new data, hierarchy review, model training, and validation.

The protocol data from ClinicalTrials.gov vary in completeness, wording quality, endpoint granularity, and metadata consistency, and they do not provide the full protocol-version or amendment history that would be needed to study how endpoint decisions changed over time. In addition, registry records may differ substantially from how individual sponsors, including AstraZeneca and Evinova, internally write, structure, and maintain clinical trial protocols. These differences limit how directly the results can be generalized to internal protocol-design workflows.

4.12.2 Hierarchy construction and reviewer dependence

The reviewed endpoint hierarchy is more clinically meaningful than the raw flat endpoint space, but it is still the outcome of methodological choices. Embedding models, clustering thresholds, merge logic, terminology signals, and LLM-assisted review all influence the final hierarchy. Even with invariance checks and deterministic validation steps, the final grouping structure cannot be considered uniquely correct; an alternative review process could reasonably produce somewhat different cluster boundaries or names.

This dependence was examined through the repeated-run reliability analysis of the merge-only LLM review stage in Section 5.2. That analysis showed stable core behavior and high endpoint-assignment agreement, but not exact deterministic reproducibility. The hierarchy should therefore be interpreted as a reviewed and practically useful representation, not as a definitive clinical taxonomy.

4.12.3 Terminology matching uncertainty

The terminology-standardization stage improved semantic consistency, but matching endpoint and protocol text to NCIt, CDISC, and LOINC codes still involved approximation, thresholding, and ambiguity resolution. Some mappings may therefore be incomplete, overly broad, or missed altogether. As discussed further in Section 5.3 and in the future-work discussion, these vocabularies should be interpreted as useful structured semantic signals rather than as flawless ground truth, and other or more domain-aligned terminology resources may improve future versions of the pipeline.

4.12.4 Aggregation, pairwise abstraction, and partial-context effects

Several prediction formulations aggregate endpoint information to protocol-level labels or transform endpoint recommendation into pairwise candidate scoring. These abstractions make the problem computationally tractable, but they also change what the model learns. Protocol-level aggregation removes local endpoint ordering and may collapse closely related endpoint formulations into the same label. Pairwise scoring makes ranking possible, but depends strongly on how negatives are sampled and how candidate pools are constructed.

The leave-one-out setting also represents only one type of partial information. It simulates the case where all but one secondary endpoint are already known. This is useful for controlled evaluation, but it does not cover every realistic design scenario, especially earlier stages where fewer endpoints may already be specified.

4.12.5 Candidate-pool dependence and Stage 1–Stage 2 error propagation

Because the final pipeline uses Stage 1 for cluster prediction and Stage 2 for endpoint ranking, endpoint-level performance depends on the quality of the Stage 1 candidate pool. If Stage 1 does not include the correct Tier B2 region among its top predictions, Stage 2 cannot recover the correct endpoint since it will not be in the candidate pool.

For this reason, Stage 2 performance must be interpreted together with candidate-pool recall and Stage 1 quality. Oracle-style candidate-pool experiments are useful for isolating the ranking ability of Stage 2, but they should not be interpreted as full deployment performance. Full pipeline performance depends on both candidate generation and endpoint ranking.

4.12.6 Small-data, class-imbalance, and ranking constraints

Even after reduction, the final dataset remains relatively small for a high-dimensional endpoint-recommendation problem. Rare Tier B1 and Tier B2 labels are difficult to estimate reliably, some labels must be filtered out because of insufficient support, and the endpoint catalogue is large relative to the number of protocols.

Class imbalance is also present in several forms. Some endpoint clusters are much more common than others, and pairwise candidate-scoring datasets contain many more negative than positive protocol–candidate pairs. These constraints limit both the complexity of the models that can be trained and the certainty with which performance differences between configurations can be interpreted. They also mean that ranking metrics and qualitative review are necessary complements to thresholded classification metrics.

4.12.7 Evaluation limits, expert review, and external validity

The evaluation is retrospective and mostly offline. The reported results therefore measure how well the pipeline reconstructs or ranks historically observed endpoint patterns under held-out evaluation, not how well it would perform prospectively in a live protocol-design workflow. Expert review was included and did affect the interpretation of model usefulness and final recommendation behavior, especially where exact-match metrics were too strict. However, this review was still qualitative and limited in scope, rather than a prospective deployment study.

Practical deployment would require broader testing across therapeutic areas, prospective validation with protocol designers, and assessment in real decision-support settings. The thesis should therefore be interpreted as a proof-of-concept study and methodological investigation rather than as a finished production system.

4.12.8 Hierarchical conditioning and oracle reference settings

Because the endpoint space is organized hierarchically, the experiments included three reference settings: no previous-tier support, predicted previous-tier support, and true previous-tier support. The true previous-tier setting is treated as an oracle reference in the current automated pipeline because true higher-tier labels are not available for new protocols by default.

In a more interactive future system, a clinical user could potentially upload endpoints and manually assign or confirm higher-tier clusters, which would make this information user-provided rather than purely oracle. However, this functionality is not part of the current implementation. The true-support setting should therefore be interpreted as an upper-bound reference for this study, while the predicted-support setting is closer to the automated deployment scenario.

4.13 Summary of the Final Pipeline

In summary, the methodology developed in this study transforms raw heart-failure trial protocols into a structured endpoint-recommendation pipeline. The workflow combines dataset reduction, hierarchical endpoint representation, terminology standardization, and two linked prediction stages.

The final system first predicts relevant endpoint clusters and then ranks concrete candidate secondary endpoints within the predicted cluster context. This reflects the central methodological idea of the study: endpoint recommendation requires not only predictive modeling, but also a structured and clinically interpretable representation of heterogeneous trial data.

4.13.1 Computational implementation and software

The pipeline was implemented in Python using a focused set of scientific-computing, machine-learning, and natural-language-processing libraries. Tabular data processing

and numerical operations were handled primarily with `pandas` and `NumPy` [31], [32]. Sparse feature matrices and numerical utilities were supported by `SciPy` [33]. Endpoint embeddings were generated using transformer-based sentence-embedding tooling through `sentence-transformers`, with PyTorch used as the underlying deep-learning backend where required [13], [34]. Hierarchical clustering, baseline machine-learning models, train–validation–test splitting utilities, cross-validation utilities, and evaluation metrics were implemented using `scikit-learn` [35]. Fuzzy terminology matching in the protocol-level and endpoint-level standardization stages used `RapidFuzz` [36]. The final tree-based ranking models were trained using `XGBoost` [19]. Model artifacts and intermediate objects were serialized with `joblib` [37]. Diagnostic figures and plots were generated with `matplotlib` [38]. Where SHAP-based feature-importance analyses were used, these were computed with `SHAP` [39].

5

Results

5.1 Clustering Analysis

The clustering analysis evaluates whether the endpoint hierarchy developed in Section 4.3 produced clinically interpretable endpoint groups that could support downstream prediction. The main empirical finding was that the earlier flat and sequential extraction strategies produced unstable or overly noisy groupings, while the hierarchical strategy produced a more coherent endpoint representation.

The final hierarchy was therefore selected because it provided a better balance between interpretability, coverage, and downstream usability. Some noise and overlap remained, especially for rare or highly specific endpoints, but the hierarchical structure made the endpoint space easier to inspect and more suitable as a prediction target.

5.1.1 Hierarchical strategy across tiers

As described in Section 4.3.2, the hierarchical strategy was selected after comparing flat clustering, single-cluster exclusion, and hierarchical clustering. The exploratory results showed that a coarse-to-fine structure captured endpoint relationships more effectively than a single global partition.

The smaller embedding setup was useful for comparing clustering strategies, while the final secondary-endpoint hierarchy was constructed using the larger embedding model. The results below therefore focus mainly on the hierarchy that was ultimately used in the downstream recommendation pipeline. However, the earlier visualizations are included only to briefly illustrate the development path.

Figure 5.1 shows an early hierarchical clustering result obtained with the smaller embedding setup. Although exploratory, it demonstrated that a hierarchical organization produced more coherent top-level structure than the previous flat approaches and therefore motivated the transition toward a fully hierarchical pipeline.

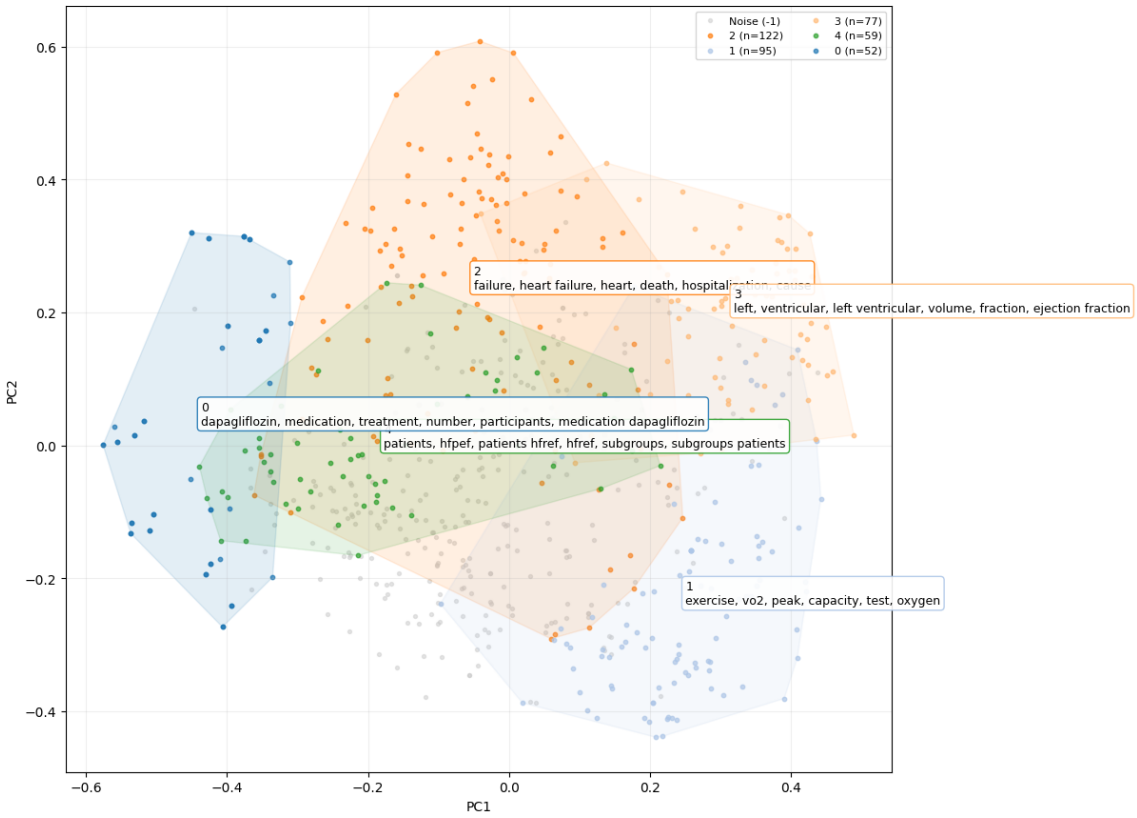


Figure 5.1: Example of an early hierarchical Tier A1 clustering result using the smaller embedding setup. This exploratory result is not part of the final secondary-endpoint hierarchy, but illustrates why the project moved away from flat clustering, as described in Section 4.3.2.

Figure 5.2 shows an earlier radial visualization from the primary-endpoint prediction phase of the project. This figure is included to illustrate the type of hierarchical structure that motivated the later coarse-to-fine representation, but it should not be interpreted as the final secondary-endpoint hierarchy used in the downstream recommendation pipeline. Some labels overlap because the plot attempts to display many hierarchy nodes in a single static figure; it is therefore intended as a structural overview rather than a label-by-label inspection figure.

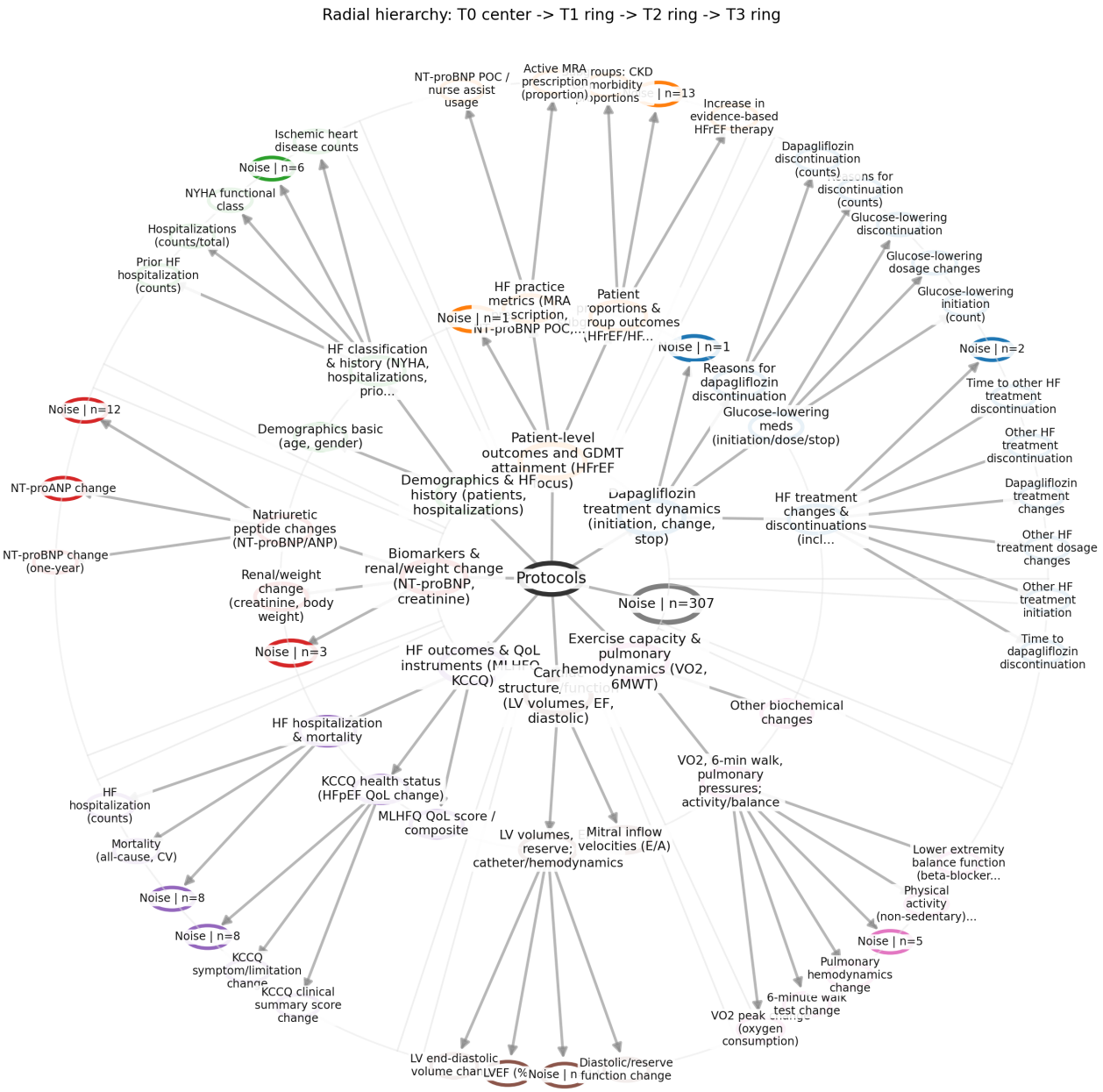


Figure 5.2: Earlier radial hierarchy visualization from the exploratory clustering phase discussed in Section 4.3.2.

Figure 5.3 shows the final Tier B0 partition of the secondary-endpoint hierarchy, where endpoints are separated into broad families later refined into Tier B1 and Tier B2 clusters. The clustering plots here are 2D Uniform Manifold Approximation and Projection (UMAP) representations of the endpoint space, where UMAP reduces high-dimensional data into a lower-dimensional visualization [40]. Axes should be interpreted as visualization coordinates, not directly meaningful clinical dimensions.

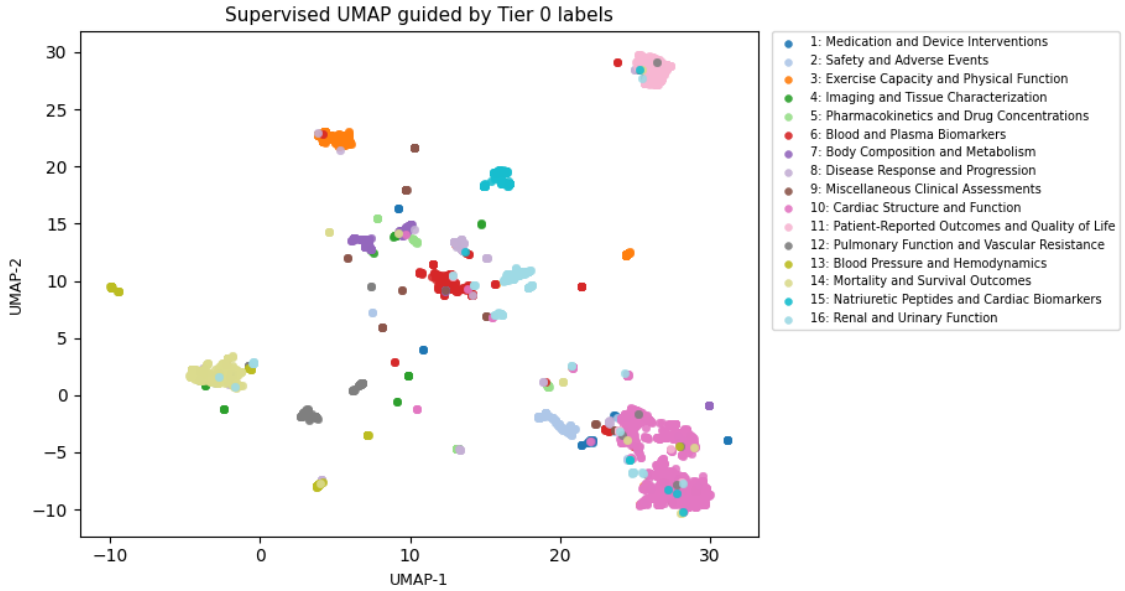


Figure 5.3: Final Tier B0 clustering produced using the larger embedding model. This top-level partition defines the broad endpoint families that form the starting point of the hierarchical pipeline.

The final hierarchy contained 16 Tier B0 clusters, 61 Tier B1 clusters, and 314 Tier B2 clusters. At the top level, it separated broad endpoint themes such as composite cardiovascular outcomes, exercise capacity and functional testing, cardiac structure and function, biomarkers, and patient-reported outcomes. Since the full hierarchy is too large to visualize completely in the main text, Figures 5.4, 5.5, and 5.6 show representative Tier B0 branches and their corresponding Tier B1 clusters.

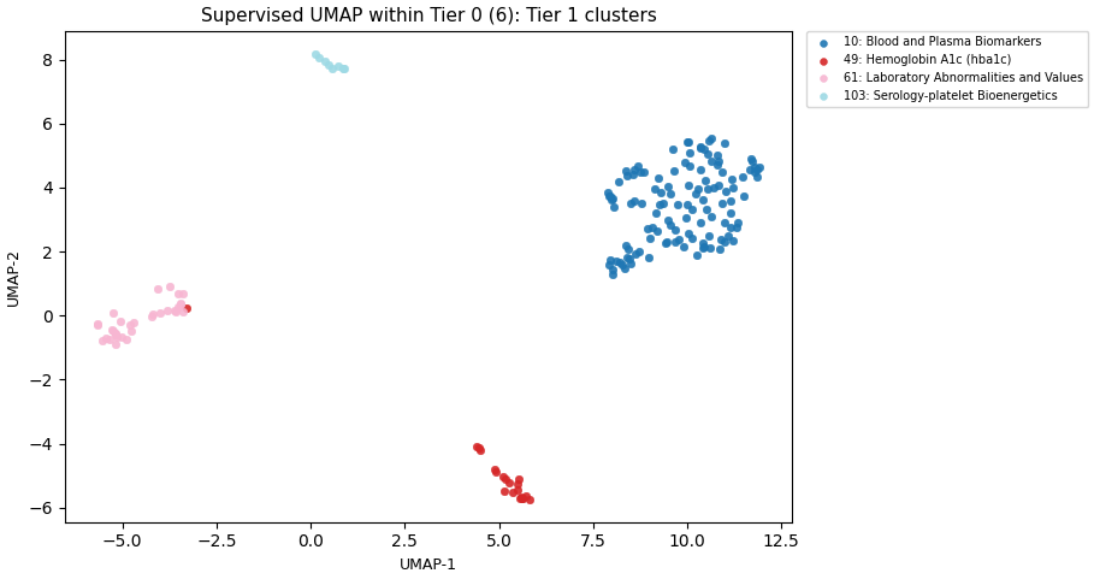


Figure 5.4: Example of a Tier B0 cluster and its corresponding Tier B1 clusters for Tier B0 ID 6 in the final hierarchical clustering. The figure illustrates how one broad top-level endpoint family is refined into more specific subgroups.

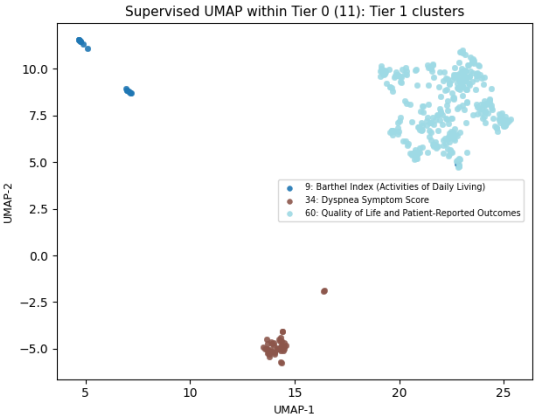


Figure 5.5: Example of a Tier B0 cluster and its corresponding Tier B1 clusters for Tier B0 ID 11 in the final hierarchical clustering.

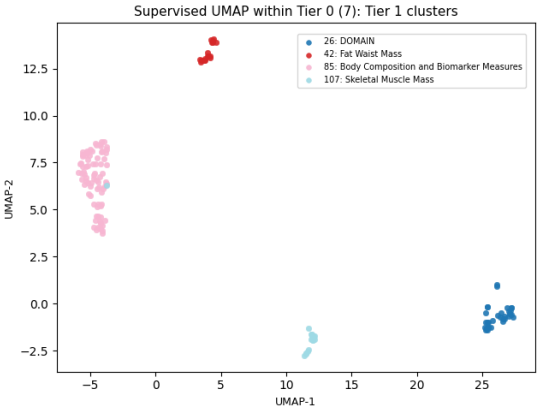


Figure 5.6: Example of a Tier B0 cluster and its corresponding Tier B1 clusters for Tier B0 ID 7 in the final hierarchical clustering.

Figure 5.7 shows a more detailed example within Tier B0 ID 7 by focusing on Tier B1 ID 85 and its Tier B2 clusters. This illustrates how the final hierarchy represents both broad endpoint families and more specific endpoint clusters without requiring all branches to have the same granularity. The resulting Tier B2 clusters capture distinct but related endpoint types. Representative examples are shown in Table 5.1.

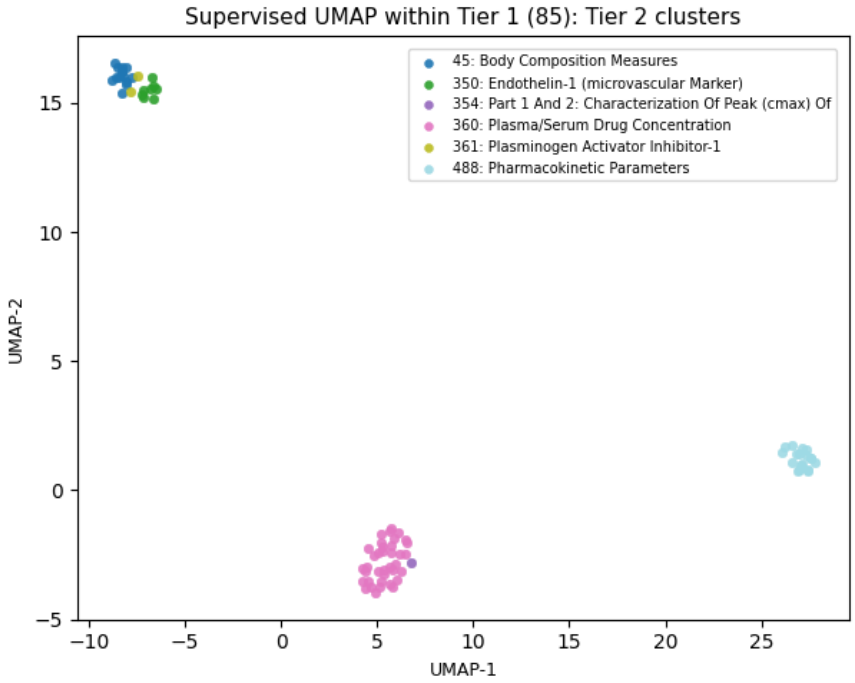


Figure 5.7: Tier B2 clusters in Tier B1 ID 85 under Tier B0 ID 7 in the final hierarchical clustering.

Tier B2 ID	Cluster name	Example endpoints
45	Body Composition Measures	1) Change from Baseline to Week 16 in body composition quantification by dual-energy X-ray absorptiometry scan (at selected sites) 2) Change From Baseline in Body Composition Assessed by DXA (Total Body Water) at Weeks 12 and 36
360	Plasma/Serum Drug Concentration	1)Pharmacokinetics (AZD4831 Plasma Exposure) 2) Cohorts A and B: Plasma Concentration of AZD5462
488	Pharmacokinetic Parameters	1) Maximum observed concentration (Cmax) 2) Pharmacokinetic Profile - Maximum Serum Concentration (Cmax)

Table 5.1: Representative Tier B2 clusters within Tier B1 ID 85 under Tier B0 ID 7, together with example endpoints from each subgroup.

These examples show that Tier B2 captures more specific endpoint concepts within a broader Tier B1 branch, such as body-composition measures, plasma or serum drug concentration, and pharmacokinetic parameters. Figure 5.8 shows the full local hierarchy within Tier B0 ID 7, corresponding to the broad endpoint family *Body Composition and Metabolism*. The background polygons represent Tier B1 clusters, while the colored points represent endpoints assigned to Tier B2 clusters. This provides a compact example of the final hierarchy's coarse-to-fine organization.

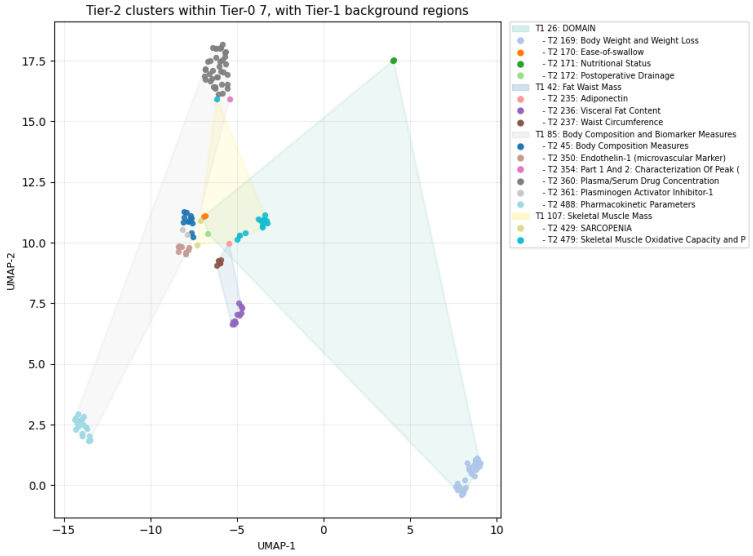


Figure 5.8: Full local hierarchy within Tier B0 ID 7 (*Body Composition and Metabolism*). The background polygons denote Tier B1 clusters within this Tier B0 branch, while the colored points denote Tier B2 clusters of the individual endpoints. The figure illustrates how the final hierarchical clustering organizes endpoints from a broad top-level family into progressively more specific clusters.

5.2 Reliability of the Merge-Only LLM Review

This section reports the repeated-run reliability results for the merge-only LLM review described in Section 4.4.4. The purpose was to assess whether the late-stage merge process produced structurally stable hierarchy outputs across repeated executions.

5.2.1 Run-level stability

Table 5.2 summarizes the five repeated runs. The main immediate observation is that the runs were procedurally consistent: all runs used the same recorded parameters, all retained the same total number of endpoints (3 700), and none produced any parse failures. This indicates that the merge-only review stage was technically robust under repetition and did not behave in a fragile or failure-prone manner.

At the same time, the exact number of accepted merges varied across runs, from 11 to 19 in total. The final hierarchy sizes remained relatively stable despite this variation: Tier B1 varied only between 59 and 60 clusters, while Tier B2 varied between 299 and 304 clusters. Thus, although the exact merge trajectories differed, the resulting hierarchy scale remained narrow and structurally comparable across runs.

Run	Merges	TB2 merges	TB1 merges	Parse fails	Fin. TB1	Fin. TB2
1	13	12	1	0	60	302
2	14	12	2	0	59	302
3	12	11	1	0	60	303
4	11	10	1	0	60	304
5	19	17	2	0	59	299

Table 5.2: Run overview for the five repeated merge-only LLM review runs.

5.2.2 Stable core merges versus variable marginal merges

Although the exact merge sets differed across runs, the repeated-run analysis revealed a clear stable core of recurring merge decisions. Across the five runs, 35 distinct decision-level merges and 42 distinct full merges were observed in total. However, four decision-level merges appeared in *all five runs*, five appeared in at least four runs, and seven appeared in at least three runs. This shows that a subset of merge decisions was repeatedly recognized by the LLM despite run-to-run stochasticity.

Table 5.3 lists the most stable recurring merges. These are particularly informative because they represent cluster relationships that the merge-only review repeatedly judged as valid under identical conditions. The recurring core included one Tier B1 merge and several highly interpretable Tier B2 merges, for example the combination of *Cardiac Adverse Events* with *Cardiovascular Events*, and the combination of *Health-Related Quality of Life Assessment Questionnaires* with *Quality of Life Measures*.

Level	Merge decision	Frequency	Mean confidence
Tier B1	Blood and Plasma Biomarkers + Inflammatory Markers → Blood and Plasma Biomarkers	5/5	0.86
Tier B2	Cardiac Adverse Events + Cardiovascular Events → Cardiac and Cardiovascular Events	5/5	0.90
Tier B2	Cardiac Function and Structure Measurements + Left Ventricular Wall Parameters and Strain Measurements → Cardiac Function and Structure Measurements	5/5	0.92
Tier B2	Health-Related Quality of Life Assessment Questionnaires + Quality of Life Measures → Quality of Life Measures	5/5	0.95
Tier B2	Surgery Beliefs and Perceptions + Surgery Hospital Anxiety and Side Effects → Surgery Beliefs, Perceptions and Anxiety	4/5	0.89

Table 5.3: Most stable recurring decision-level merges across the five repeated runs.

At the same time, many merges appeared in only one or two runs. This indicates that the LLM was not fully deterministic at the exact-merge level. Less obvious or more borderline merge opportunities were handled differently across runs, especially at the margins of the hierarchy. This distinction between a recurring stable core and more variable marginal decisions is important for interpreting the role of the LLM in the pipeline. The results do not support the claim that repeated merge-only review yields an identical patch each time, but they do support the weaker and more defensible claim that the LLM consistently identifies a meaningful subset of the most obvious merges.

5.2.3 Exact merge reproducibility versus final structural agreement

To quantify how similar the merge behavior was across runs, pairwise Jaccard similarity was computed between the merge sets of every run pair. These values were relatively low. Decision-level Jaccard ranged from 0.1818 to 0.4375, with a mean of approximately 0.3040, while full Jaccard ranged from 0.1739 to 0.3529, with a mean of approximately 0.2302. These values show that the exact merge sets overlapped only partially across repeated runs.

However, the final endpoint-level hierarchy assignments were substantially more stable than the merge-set overlap alone would suggest. Pairwise final endpoint-assignment agreement ranged from 0.8225 to 0.9640, with a mean of approximately 0.8824. Thus, even though the exact merges differed, the corrected hierarchies still placed the same endpoints in the same final Tier B1-Tier B2 locations in the large majority of cases. Table 5.4 summarizes the most important aggregate reliability results.

Statistic	Value
Number of repeated runs	5
Unique parameter hashes	1
Total parse failures across runs	0
Unique final hierarchy hashes	5
Distinct decision-level merges across all runs	35
Distinct full merges across all runs	42
Decision-level merges seen in all runs	4
Decision-level merges seen in at least 4 runs	5
Decision-level merges seen in at least 3 runs	7
Pairwise decision-level Jaccard (min-max)	0.1818-0.4375
Pairwise full Jaccard (min-max)	0.1739-0.3529
Pairwise endpoint-assignment agreement (min-max)	0.8225-0.9640
Mean endpoint-assignment agreement	0.8824
Final Tier B1 count range	59-60
Final Tier B2 count range	299-304

Table 5.4: Aggregate summary statistics from the repeated merge-only reliability analysis.

This contrast between low merge-set Jaccard and much higher final endpoint-assignment agreement is one of the most important findings of the reliability analysis. It shows that the merge-only LLM behaves stochastically at the level of individual merge proposals, but that many of these local differences do *not* substantially alter the final structure of the hierarchy. In other words, repeated runs do not follow the same exact path, but they often arrive at broadly similar final endpoint placements.

5.2.4 Interpretation of warnings and rejected merges

The warning outputs also provided useful qualitative evidence about the behavior of the review stage. The warnings did not mainly reflect malformed outputs or parser instability; instead, they reflected *low-confidence merge proposals being rejected*. For example, one repeatedly rejected Tier B1 merge attempted to combine *Fat Waist Mass* with *Scar Mass (as %left Ventricular)* into a generic *Body Composition Measurements* cluster with confidence 0.52. Another repeatedly rejected Tier B2 proposal attempted to merge *Heart Rate* with *Treatment-related TEAEs* at confidence 0.0. These rejected proposals are useful because they show that the confidence-based filtering and validation logic acted conservatively and prevented clearly implausible merges from entering the final hierarchy.

This behavior supports the interpretation that the merge-only review stage is not simply accepting arbitrary LLM suggestions. Instead, the stage appears to combine stochastic proposal generation with deterministic validation and thresholding, so that obviously weak proposals are filtered out before they can affect the final corrected hierarchy.

5.2.5 Overall interpretation

Overall, the merge-only LLM review was not exactly reproducible, since the five runs produced different final hierarchy hashes and only partial overlap between exact merge sets. However, the results support structural reliability: there were no parse failures, no endpoint losses, a stable core of recurring merge decisions, and high final endpoint-assignment agreement across runs.

The reliability analysis therefore supports the use of the reviewed hierarchy in the downstream modeling pipeline, while also showing that the LLM-assisted merge step should not be described as deterministic.

5.3 Role of Standardization in Downstream Prediction

The standardization stage was not evaluated as a standalone prediction experiment. Its effect instead appeared through the downstream Stage 1 and Stage 2 models, where NCIt, CDISC, and LOINC codes were used as structured semantic features, as described in Section 4.5.

The main result was that standardized codes were useful as supporting signals, but not as replacements for hierarchy-aware modeling. In Stage 1, observed secondary-endpoint code bags were most useful in the pairwise candidate-scoring formulation, where the model compared partial endpoint context against candidate clusters. Direct multilabel prediction from high-dimensional code bags remained more limited.

In Stage 2, standardized codes contributed more directly through candidate-code features and overlap-based features between protocol-side and candidate-side representations. Overall, terminology standardization improved semantic comparability and interpretability, but worked best when combined with the reviewed endpoint hierarchy, partial endpoint context, and pairwise ranking.

5.4 Stage 1 Model Performance

Stage 1 evaluated cluster-level recommendation using the formulations described in Section 4.9. The main comparison was between from-scratch multilabel prediction and pairwise leave-one-out cluster scoring.

The key result was that the from-scratch setup provided a useful cold-start baseline, but the pairwise leave-one-out formulation was more useful for the final recommender workflow. It better supported top-k ranking, missing-cluster recovery, and candidate generation for Stage 2. Across the final Stage 1 experiments, RF was competitive in the from-scratch multilabel setting, while XGBoost was selected for the final pairwise leave-one-out configuration.

5.4.1 From-Scratch Model

The from-scratch experiments evaluated how much endpoint-cluster information could be recovered without observed secondary-endpoint context. Two formulations were tested: direct multilabel prediction and row-pair-based binary scoring. Their final performance was broadly comparable, with both reaching approximately 50% test micro-F1 in the final from-scratch setting.

5.4.1.1 Multilabel Approach

The multilabel formulation was evaluated under the three hierarchical support settings defined in Section 4.9.2: **Prev=None**, **Prev=Pred**, and **Prev=True**. These settings compare strict from-scratch prediction, realistic predicted-tier support, and oracle previous-tier support.

Two tree-based multilabel classifiers were evaluated for the from-scratch multilabel setup: RF and XGBoost. The final reported results do not correspond to a single model family across all hierarchical conditioning settings. Instead, the best-performing model differed depending on the source of previous-tier information. For the realistic settings, **Prev=None** and **Prev=Pred**, RF produced the strongest results. In contrast, for the oracle-conditioned setting, **Prev=True**, XGBoost produced the strongest results. The figures and tables in this subsection should therefore be interpreted as the best selected model for each conditioning setting, rather than as the output of one fixed classifier across all settings.

This distinction is important because the oracle setting changes the prediction problem substantially. When true previous-tier labels are supplied, the downstream model receives cleaner hierarchical information than is available in the realistic pipeline. XGBoost benefited more from this clean oracle signal, while RF was more robust in the settings where previous-tier information was absent or predicted.

Conditioning setting	Selected model	Interpretation
Prev=None	RF	Strict from-scratch prediction without previous-tier information.
Prev=Pred	RF	Realistic hierarchical pipeline using predicted previous-tier labels.
Prev=True	XGBoost	Oracle upper-bound setting using true previous-tier labels.

Table 5.5: Selected model family for each hierarchical conditioning setting in the from-scratch multilabel experiments.

Parameter	RF	XGBoost
`n_estimators`	500	350
`max_depth`	12	4
`min_samples_leaf`	4	–
`max_features`	`sqrt`	–
`class_weight`	`balanced`	–
`learning_rate`	–	0.03
`subsample`	–	0.85
`colsample_bytree`	–	0.85
`reg_lambda`	–	2.0
`min_child_weight`	–	4
Objective / fitting criterion	Multilabel classification with impurity-based splits (`gini` default)	`binary:logistic` one-vs-rest multilabel classification
Training evaluation metric	–	`logloss`
Tree method	–	`hist`
Training strategy	`weighted|pow=1.00`	`weighted|pow=1.00`
Random seed	42	42

Table 5.6: RF and XGBoost configurations used in the from-scratch multilabel hierarchical-comparison experiments. The selected model family differed between previous-tier feature settings, as shown in Table 5.5.

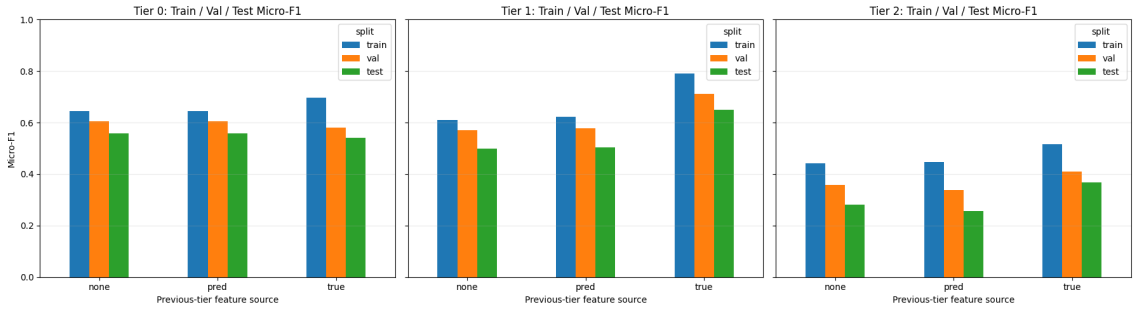


Figure 5.9: Test micro-F1 across Tier B0, Tier B1, and Tier B2 under the three hierarchical support settings.

The exact test micro-F1 values for the selected model under each conditioning setting are shown in Table 5.7.

Tier / Configuration	None	Pred	True
Tier B0	0.558559	0.558559	0.541772
Tier B1	0.497619	0.503113	0.650078
Tier B2	0.281426	0.256314	0.367839

Table 5.7: Exact test micro-F1 values for the selected from-scratch multilabel model under each hierarchical support setting.

Figure 5.9 and Table 5.7 show that **Prev=None** and **Prev=Pred** are relatively similar overall. For Tier B1, predicted hierarchical support gives only a marginal improvement over the strict from-scratch baseline, while for Tier B2 it performs slightly worse. This confirms the limitation discussed in Section 4.9.2: realistic predicted support can suffer from error propagation, which weakens its value for downstream child-tier prediction. By contrast, **Prev=True** improves performance substantially for Tier B1 and Tier B2, indicating that the hierarchy contains useful predictive information when upstream tier assignments are correct.

Importantly, the oracle setting remains far from perfect. In a theoretically ideal hierarchy with well-separated child clusters, correct parent-tier labels would be expected to remove much of the downstream ambiguity. The remaining error therefore suggests that the current cluster structure is still imperfect: even within the correct parent tier, several child clusters may be semantically close, overlapping, or difficult to distinguish from the available protocol features. The **Prev=True** setting should therefore be interpreted both as an upper-bound estimate of correct hierarchical conditioning and as a diagnostic indication of how well the current hierarchy separates endpoint concepts. This also suggests that improved clustering could increase the practical value of hierarchical conditioning in later versions of the pipeline.

Hamming loss was also examined to provide the complementary error view defined in Section 3.10 and used in Section 4.11.

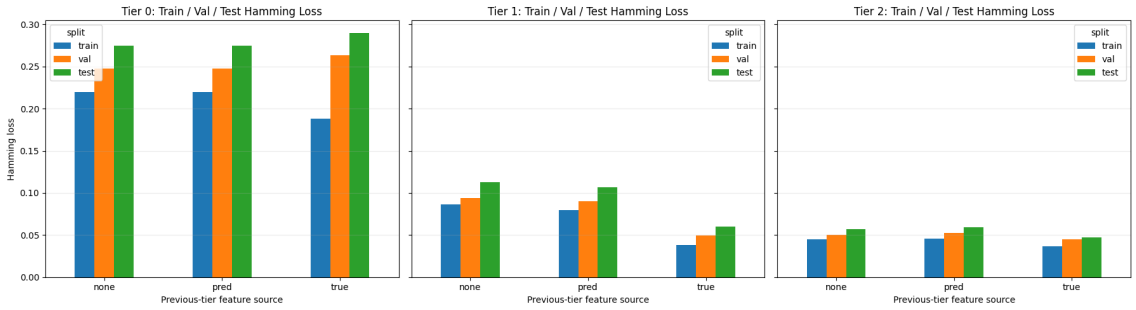


Figure 5.10: Test Hamming loss across Tier B0, Tier B1, and Tier B2 under the three hierarchical conditioning settings.

The Hamming-loss results follow the same general pattern as micro-F1. The difference between **Prev=None** and **Prev=Pred** is small, while the oracle-conditioned setting yields consistently lower loss for the lower tiers. This again suggests that the current predicted hierarchical signals are not yet accurate enough to provide strong practical gains, even though the hierarchy is clearly useful in principle.

The following figures report the top-(k) ranking metrics defined in Section 3.10 and applied as described in Section 4.11. These metrics are reported because the intended Stage 1 output is a ranked set of plausible endpoint clusters, not only a thresholded multilabel prediction.

The Hit@k results show the same broad pattern as the thresholded metrics. Hit@k is

generally higher for Tier B0 and Tier B1, while Tier B2 remains more difficult due to the finer cluster granularity. The difference between **Prev=None** and **Prev=Pred** is small across tiers, indicating that predicted parent-tier information does not substantially improve whether at least one relevant cluster appears near the top of the ranking. In contrast, **Prev=True** improves Hit@k most clearly for Tier B2, suggesting that correct upstream information is particularly useful when the model must choose among more fine-grained endpoint clusters.

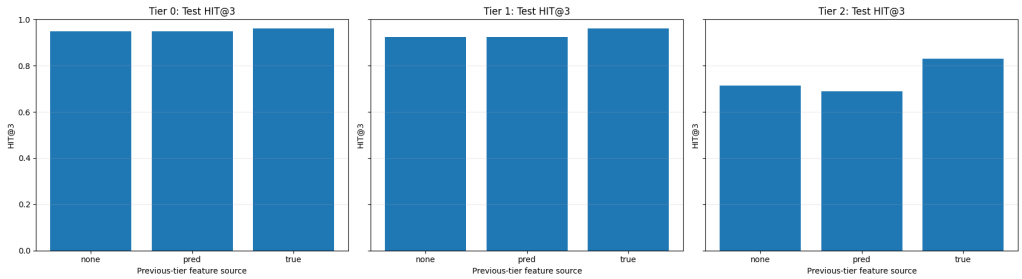


Figure 5.11: Hit@3 across the three tiers under the three hierarchical support settings.

The Recall@k results show that **Prev=Pred** gives a small improvement over **Prev=None** in Tier B1, but not in Tier B2. This again indicates that predicted hierarchical support is not yet reliable enough to improve deeper-tier recovery consistently. The strongest recall values are obtained under **Prev=True**, which confirms that correct parent-tier information can improve recovery of relevant child clusters. However, the improvement is still limited by the separability of the child clusters themselves.

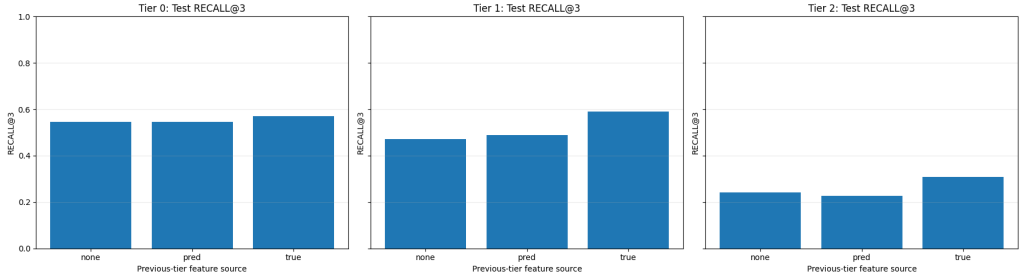


Figure 5.12: Recall@3 across the three tiers under the three hierarchical support settings.

The Precision@k results follow the same interpretation. **Prev=None** and **Prev=Pred** remain close, while **Prev=True** produces cleaner top-ranked suggestions, especially for Tier B2. This suggests that the problem is not only that the model lacks hierarchical information, but that the predicted hierarchy is too noisy to improve the ranking in the realistic setting. When the hierarchy is supplied correctly, the top-ranked predictions become more accurate, but not perfect.

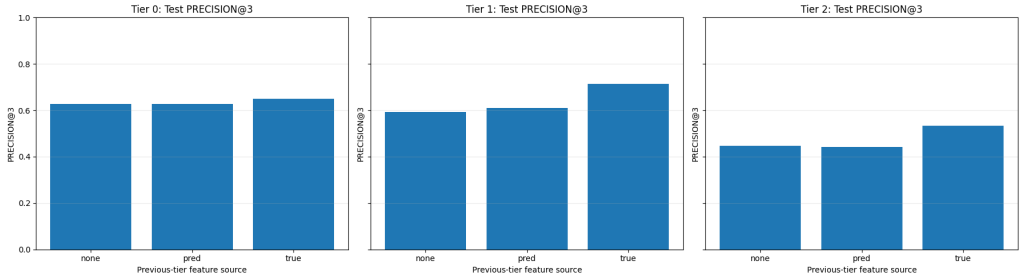


Figure 5.13: Precision@3 across the three tiers under the three hierarchical support settings.

Overall, the multilabel ranking results show that hierarchical support provides useful but moderate additional signal. The **Prev=None** setting already achieves relatively strong top-(k) performance, especially for Tier B0 and Tier B1, which indicates that much of the ranking signal is captured directly from the protocol and endpoint-code features. The oracle **Prev=True** setting improves the results most clearly for Tier B2, but the absolute gains are limited rather than transformative. This suggests that correct parent-tier information helps refine the ranking, particularly at the finest cluster level, but does not by itself solve the prediction task.

The realistic **Prev=Pred** setting remains close to **Prev=None**, and in some Tier B2 cases slightly weaker. This indicates that predicted parent-tier labels are not yet reliable enough to provide consistent ranking gains. Taken together with the oracle results discussed above, the ranking metrics suggest that future improvements should focus on making hierarchical information more useful in practice: both by improving upstream prediction reliability and by refining the Tier B2 cluster structure where child clusters remain hardest to separate.

A brief RF tree diagnostic was inspected to verify that the selected from-scratch model learned clinically interpretable local patterns rather than only arbitrary splits. Since RF was the strongest model in the realistic **Prev=None** and **Prev=Pred** settings, this diagnostic is most relevant as an illustration of model behaviour in the realistic from-scratch pipeline.

Figure 5.14 shows one representative Tier B2 tree from the selected RF model with **Prev=Pred**. The purpose of this figure is not to provide a complete tree-by-tree explanation of the forest, but to illustrate that the model relied on recognizable protocol-level signals when separating endpoint clusters.

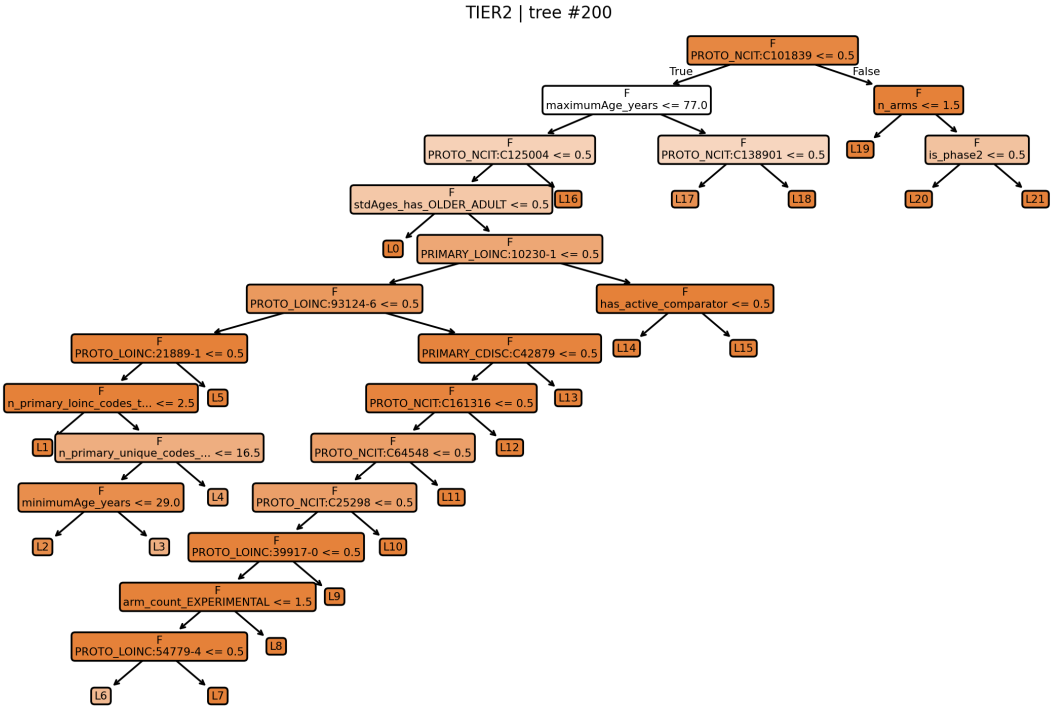


Figure 5.14: Example decision tree from the selected RF model for Tier B2.

A more local example is shown in Figure 5.15, which zooms in on one decision path leading to Leaf 21 (L21). This branch corresponded to protocols with a NYHA-related signal, at least two study arms, and Phase II status, and it was strongly associated with the endpoint cluster *NT-proBNP Measurements*. Other parts of the same tree also used features such as age constraints and primary-endpoint terminology, indicating that the model could combine several different types of structured protocol information.

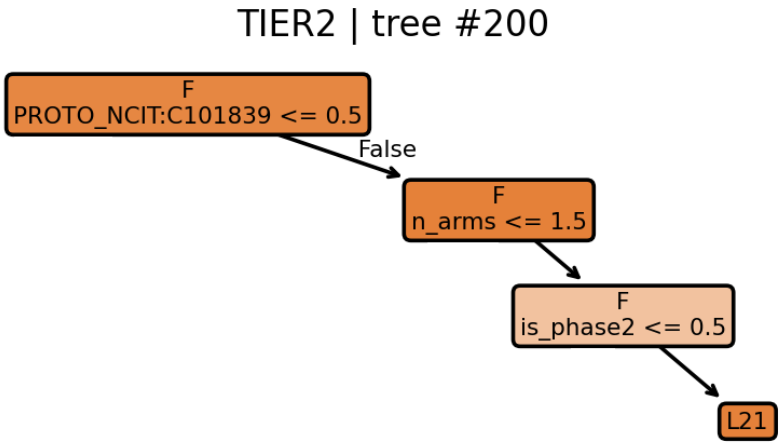


Figure 5.15: Zoomed-in view of the decision path leading to Leaf 21 (L21) in the Tier B2 RF tree.

This diagnostic supports the interpretation that the model captured some clinically meaningful protocol-to-cluster relationships. It also suggested that the same endpoint theme could sometimes be reached through multiple leaves, meaning that the model had learned more than one protocol profile associated with a given cluster. However, because this tree inspection is illustrative rather than a primary evaluation result, the detailed leaf profiles, endpoint lists, and alternative decision paths were omitted here to keep the focus on the main comparative results.

5.4.1.2 Row-Pair-Based Binary Approach

The row-pair-based binary formulation achieved performance broadly similar to the direct multilabel model in the final from-scratch setting. This suggests that the main limitation was not only the prediction formulation, but also the difficulty of learning stable protocol-to-cluster mappings from the available feature space.

5.4.2 Leave-X-Out Stage 1 Results

The leave-X-out Stage 1 experiments evaluated whether partial secondary-endpoint context improved cluster recommendation compared with the from-scratch baseline. The final experiments focused mainly on leave-one-out, as described in Section 4.9.3.

The main result was that partial endpoint information was useful only when the task formulation matched the recommendation objective. Direct single-cluster recovery performed poorly, while pairwise candidate scoring gave the most useful ranking behavior and became the final Stage 1 formulation.

5.4.2.1 Approach 1: Predicting a Single Withheld Cluster

The first leave-X-out formulation directly predicted one withheld Tier B1 cluster from the observed endpoint context. This proved too restrictive for a multilabel endpoint setting: the best test accuracy was only 0.227 for logistic regression, while XGBoost and RF performed worse.

Model	Tier	Val. acc.	Test acc.	Test top-3	Test top-5
LR	Tier B1	0.245	0.227	0.420	0.512
XGBoost	Tier B1	0.159	0.121	0.300	0.413
RF	Tier B1	0.020	0.014	0.024	0.031

Table 5.8: Single-withheld-cluster recovery results for the first leave-X-out Stage 1 formulation.

These results show that missing-cluster recovery should not be framed as a single isolated multiclass decision. The formulation was therefore not used as the final Stage 1 approach.

5.4.2.2 Approach 2: Predicting the Full Cluster Set from Partial Endpoint Code Bags

The second leave-X-out formulation kept the full protocol-level cluster vector as the target, while adding partial secondary-endpoint code bags as input. This produced substantially stronger results than the single-withheld-cluster formulation.

The best representative XGBoost configuration indicated a great increase in performance for **Prev = Pred** and **Prev = True**, just above 90% micro-F1. At the same time, **Prev = None** also had an increase, but much lower, approximately 60% micro-F1. This showed that partial secondary-endpoint evidence can improve cluster prediction when the target remains a full multilabel profile. However, the results were sensitive to code-vocabulary pruning, especially for secondary-endpoint codes.

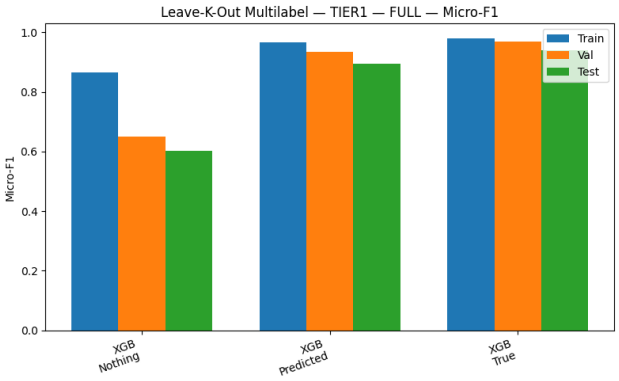


Figure 5.16: Leave 1 out, target mode = Full, Multilabel approach

Overall, this approach showed that partial endpoint context was useful, especially when the target was the full protocol-level cluster profile. However, the stronger performance in the `full` target mode did not fully answer the more important recommendation question in this study: which endpoint cluster is missing from the already observed endpoint context? Since Approach 1 had performed poorly when directly targeting missing-cluster recovery, the results from Approaches 1 and 2 together motivated a third formulation: pairwise candidate scoring, where missing-cluster recommendation could be treated as a ranking problem rather than as a single isolated prediction or full-profile reconstruction task.

5.4.2.3 Approach 3: Pairwise Candidate Scoring with Partial Endpoint Code Bags

The third leave-X-out formulation converted Stage 1 into a pairwise candidate-scoring task, as described in Section 4.9.3. This approach was examined because the earlier experiments left an important gap: Approach 1 directly targeted missing-cluster recovery but performed poorly, while Approach 2 performed much better mainly by predicting the full protocol-level cluster profile. Since the final Stage 1 objective is closer to the `missing` target mode, the next step was to test whether candidate ranking could improve missing-cluster recommendation.

In this formulation, the model no longer had to output one isolated withheld cluster or reconstruct the entire cluster vector in a single step. Instead, each partially observed protocol context was paired with candidate clusters, and the model scored

how plausible each candidate was as missing information. This made the task better aligned with the final recommender workflow, where the system must rank candidate endpoint clusters for downstream Stage 2 endpoint ranking.

Since the pairwise formulation introduced both a new dataset construction procedure and a candidate-ranking objective, an additional grid search was performed over the most important construction and pruning parameters. The search focused primarily on source-aware code pruning thresholds and pairwise negative sampling. In particular, the following parameters were varied:

- minimum document frequency for protocol codes,
- minimum document frequency for primary endpoint codes,
- minimum document frequency for secondary endpoint codes,
- maximum document-frequency ratio used to remove overly common codes,
- number of sampled negative candidate rows per positive row during training.

The initial strongest pairwise configuration used

$$\texttt{proto\_min} = 2, \quad \texttt{primary\_min} = 3, \quad \texttt{secondary\_min} = 2, \quad \texttt{max\_doc\_freq\_ratio} = 0.35,$$

together with

$$\texttt{PAIR\_MAX\_COMBOS\_PER\_PROTOCOL\_PER\_K} = 10,$$

and target-specific negative sampling:

$$\texttt{PAIR\_NEGATIVES\_PER\_POSITIVE\_BY\_TARGET\_MODE} = \{\texttt{full} : 10,\ \texttt{missing} : 3\}.$$

With XGBoost as the pairwise classifier, this initial configuration produced strong aggregate results. In the Tier B1 missing-target setting, the model reached a test AUC of approximately 0.83 and test F1 of approximately 0.67. The ranking metrics were also high with test Hit@3 of approximately 0.992. These results confirmed that the pairwise formulation was a promising direction and that candidate scoring could exploit partial endpoint information more effectively than the previous direct prediction formulations.

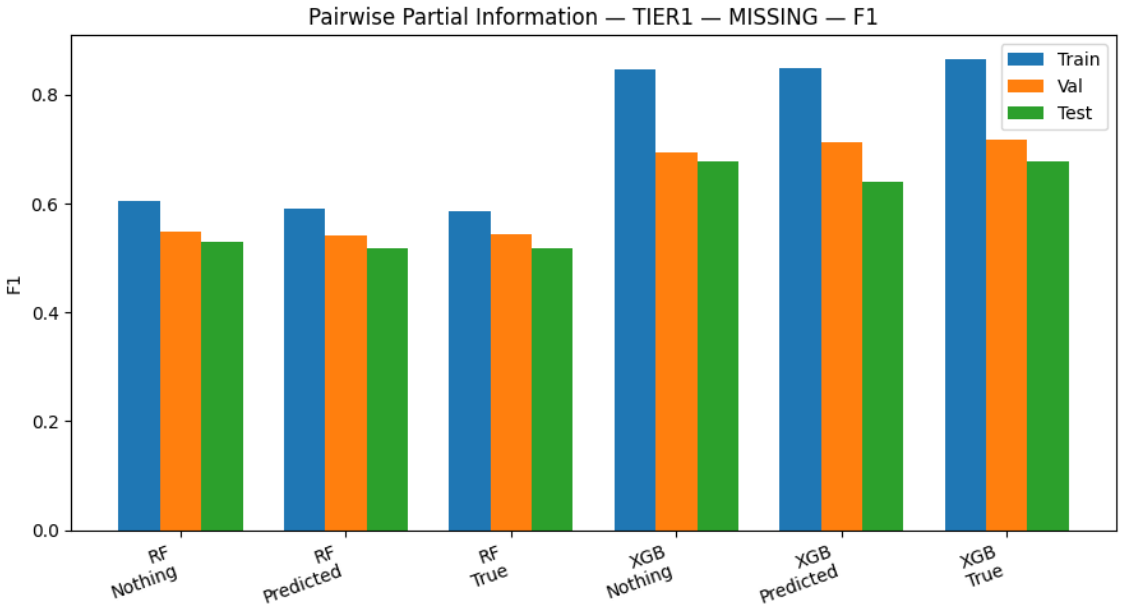


Figure 5.17: Initial Tier B1 pairwise missing-target F1. The missing-target setting is stricter than the full-target setting because it focuses on candidate clusters associated with the withheld endpoint context.

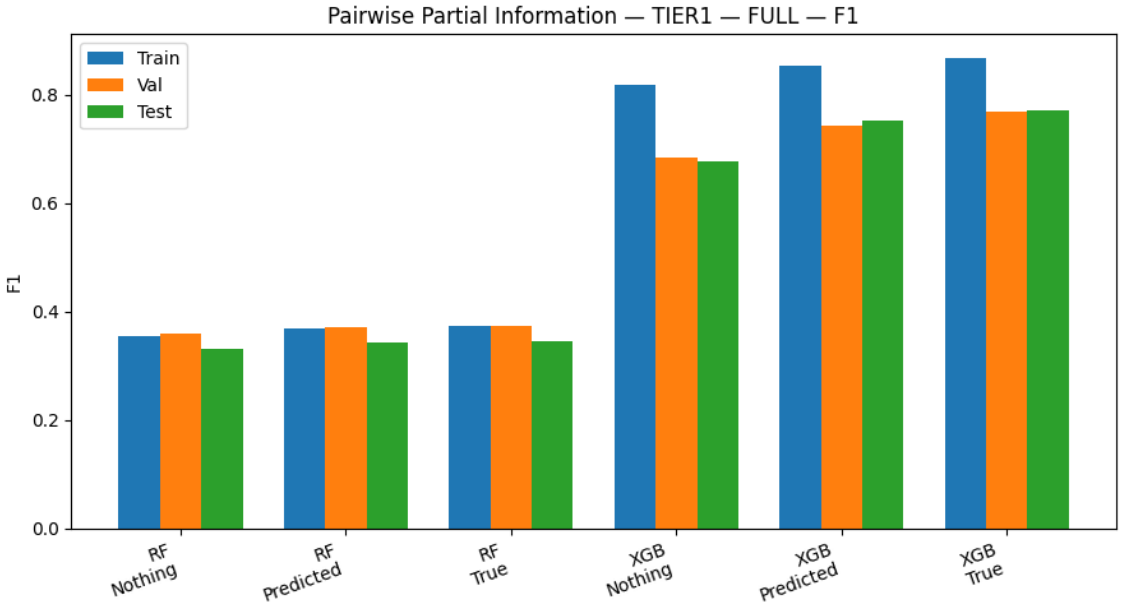


Figure 5.18: Initial Tier B1 pairwise full-target F1. The first pairwise configuration showed strong aggregate performance, especially for XGBoost.

Although the initial pairwise model achieved strong aggregate metrics, manual inspection on unseen protocols revealed an important limitation. The model frequently assigned high scores to large and common endpoint clusters, causing similar cluster predictions to appear across many different protocols. These predictions were

often clinically plausible and domain-relevant, but they were not always the most informative missing endpoint categories for the specific protocol. In several cases, they represented broad endpoint families that were already present in the observed endpoint context, meaning that the model did not provide much genuinely new information.

This behavior was not fully captured by standard row-level metrics such as AUC, average precision, or F1. A model can perform well on these metrics by correctly identifying frequent positive clusters, while still being less useful as a recommendation system if it repeatedly predicts the same high-prevalence clusters. The issue became clearer during qualitative inspection and was also raised during expert review.

Repeated cluster	**Predictions**	**Protocols**	**Avg. rank**
NYHA Functional Classification	10	10	1.1
Kansas City Cardiomyopathy Questionnaire Symptom Scores	9	9	3.2
NT-proBNP Measurements	9	9	4.0
Quality of Life Measures	9	9	5.7
Cardiac Adverse Events	9	9	7.3
Cardiac Function and Structure Measurements	8	8	5.9
Health-Related Quality of Life Assessment Questionnaires	8	8	7.6
All-cause Mortality and Hospitalization Events	7	7	5.9
Heart Failure Events	7	7	7.6
Six Minute Walk Test Distance	5	5	1.8

Table 5.9: Example of repeated cluster predictions in an unseen-protocol inspection run. Several common endpoint clusters appeared across most protocols, indicating that the initial pairwise model was partly biased toward high-prevalence, broadly relevant clusters.

During our first expert review, Niklas Bergh noted that many of these predictions were not random or clinically meaningless. Niklas Bergh is a Senior Medical Director in Early Clinical Development, CVRM, at AstraZeneca, and served as a clinical expert reviewer in this study. Rather, they often captured the correct domain or general clinical theme. However, he also pointed out that several predictions represented "low-hanging fruit": common endpoint families that were broadly relevant, but not necessarily specific enough to the protocol being evaluated. This supported our own manual inspection and motivated a rethinking of the training configuration.

The main conclusion from this analysis was that the pairwise formulation itself remained useful, but the training objective needed to reduce the dominance of large clusters. The original weighting strategy balanced positive and negative pairwise rows, but it did not explicitly balance the candidate clusters within the positive class. Therefore, rare positive clusters could still be under-emphasized during training.

To reduce the tendency toward high-prevalence clusters, an additional inverse-frequency weighting scheme was introduced for positive pairwise rows. The original pairwise weighting only balanced binary classes, meaning positive rows against negative rows. The final configuration kept this binary balancing, but also multiplied positive rows by an inverse-frequency factor based on how common their candidate cluster was among positive training rows.

In practice, positive rows from rare candidate clusters received larger sample weights, while positive rows from very common clusters received smaller relative emphasis. The multiplier was capped to avoid unstable training. The final setting used an inverse-frequency power of 1.25 and a maximum multiplier of 25. This modification did not change the pairwise dataset construction, candidate features, or model architecture. It only changed how strongly different positive candidate clusters contributed during training.

For reproducibility, the final pairwise models were trained using the same XGBoost classifier configuration across the reported final experiments. The same positive-row inverse-frequency weighting strategy was also used throughout the final pairwise candidate-scoring experiments. The complete classifier and weighting configuration is shown in Table 5.10.

Component	**Final setting**
Classifier	XGBoost binary classifier
`n_estimators`	250
`max_depth`	4
`learning_rate`	0.05
`subsample`	0.85
`colsample_bytree`	0.85
`reg_lambda`	2.0
`min_child_weight`	4
Objective	`binary:logistic`
Evaluation metric	`logloss`
Tree method	`hist`
Random seed	42
Inverse-frequency weighting	Enabled
Inverse-frequency power	1.25
Maximum inverse-frequency multiplier	25.0
Inverse-frequency weighting scope	Positive pairwise rows only

Table 5.10: Final XGBoost and inverse-frequency weighting configuration used for the pairwise candidate-scoring experiments.

This adjustment reduced some of the headline aggregate metrics compared with the initial pairwise configuration, but it produced more useful behavior in qualitative inspection. The model became less dominated by the largest endpoint clusters and showed better ability to surface more specific candidate clusters. For this reason,

the inverse-frequency-weighted model was selected as the final Stage 1 pairwise leave-X-out configuration, despite having slightly lower aggregate F1 than the earlier baseline.

Since the `missing` target mode most closely matched the intended endpoint-completion use case, the final analysis focused on this setting. The final detailed Tier B2 analysis was therefore carried out using the no-cluster-info configuration, with the corresponding comparison to predicted and true cluster-support variants reported below.

Final missing-target data representation. The complete hierarchy contained 16 Tier B0 labels, 61 Tier B1 labels, and 314 Tier B2 labels. Before model training, however, labels with fewer than three positive examples were removed from the candidate label space:

$$\texttt{MIN\_LABEL\_SUPPORT} = \{\texttt{tier0} : 3,\ \texttt{tier1} : 3,\ \texttt{tier2} : 3\}.$$

This removed ultra-rare clusters that would be difficult to learn reliably and would otherwise create unstable candidate labels. At the endpoint-instance level, the retained candidate labels still covered 3 322 of 3 700 endpoint assignments, corresponding to 89.8% of the original endpoint data.

Tier	Labels in hierarchy	Minimum support	Retained labels
Tier B0	16	3	16
Tier B1	61	3	48
Tier B2	314	3	172

Table 5.11: Candidate label-space pruning for the Stage 1 models. Labels with fewer than three positive examples were excluded from the model candidate space to reduce instability from ultra-rare clusters.

For the final leave-one-out training split, the model data contained 231 unique training protocols. Table 5.12 summarizes the resulting feature composition after source-aware ontology-code pruning. These counts refer to the final leave-one-out training split used by this model, not to the entire raw database.

Feature/code group	Raw codes	Retained	Role in design matrix
Structured protocol features	–	89	Protocol metadata and structured trial variables
Protocol-level code indicators	1 693	686	Protocol-level ontology evidence
Primary-endpoint code indicators	1 532	324	Primary-endpoint ontology evidence
Observed secondary-endpoint code indicators	3 823	1 769	Partial observed endpoint context
Candidate Tier B2 label indicators	–	172	Candidate cluster representation
Overlap features	–	4	Similarity between observed codes and candidate prototype
Total	**7 048**	**3 044**	Final pairwise feature dimension

Table 5.12: Feature composition of the final Tier B2 missing-target leave-one-out no-cluster-info pairwise design matrix. Raw code counts are shown before source-aware pruning; retained feature counts correspond to the final trained feature matrix.

The corresponding row-expanded pairwise datasets are summarized in Table 5.13.

Split	Rows	Positive rows	Sampled negative rows
Train	2 688	672	2 016
Validation	776	194	582
Test	832	208	624

Table 5.13: Pairwise row counts for the final Tier B2 missing-target leave-one-out no-cluster-info model. Each split contains positive candidate rows and sampled negative candidate rows.

Figure 5.19 summarizes the final pairwise F1 results for both Tier B1 and Tier B2 under the inverse-frequency-weighted no-cluster-info configuration. Across all four settings, XGBoost achieved stronger F1 than RF. The train, validation, and test scores remained relatively aligned, indicating that the final configuration did not show severe overfitting despite the class imbalance in the pairwise candidate-scoring task.

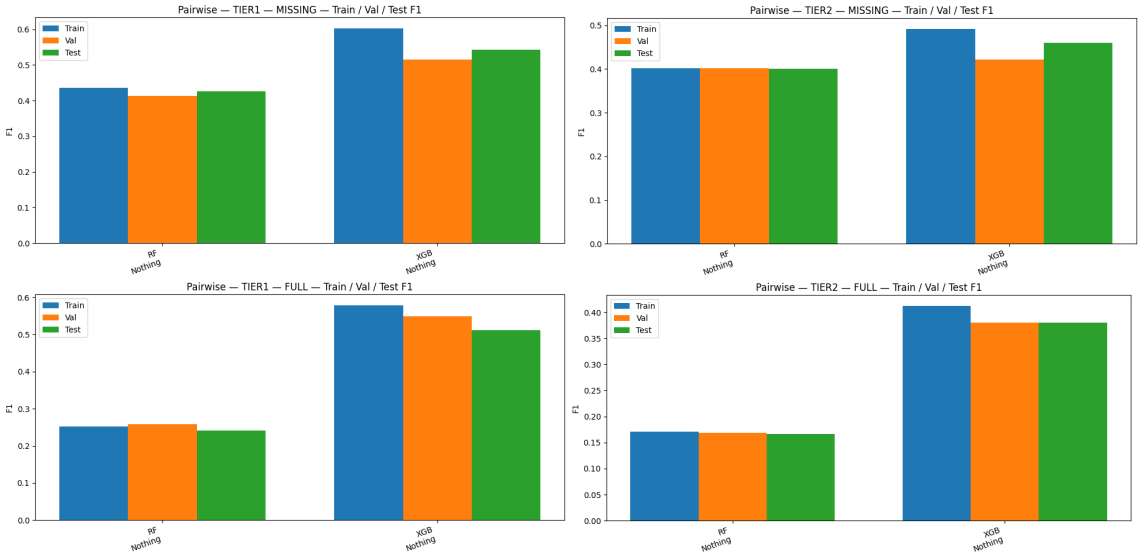


Figure 5.19: Pairwise F1 results for the final inverse-frequency-weighted no-cluster-info Stage 1 configuration. The figure compares RF and XGBoost across Tier B1 and Tier B2, for both missing-target and full-target evaluation. XGBoost consistently achieved higher F1 than RF. Tier B2 was generally harder than Tier B1 because it uses a more fine-grained and sparse cluster label space.

The merged F1 results support the selection of XGBoost for the final pairwise Stage 1 model. The lowest performance appeared for RF in the Tier B2 full-target setting. More importantly for the final recommendation pipeline, XGBoost remained stronger in the missing-target setting, which is the setting most closely aligned with recovering endpoint information from a partially specified protocol.

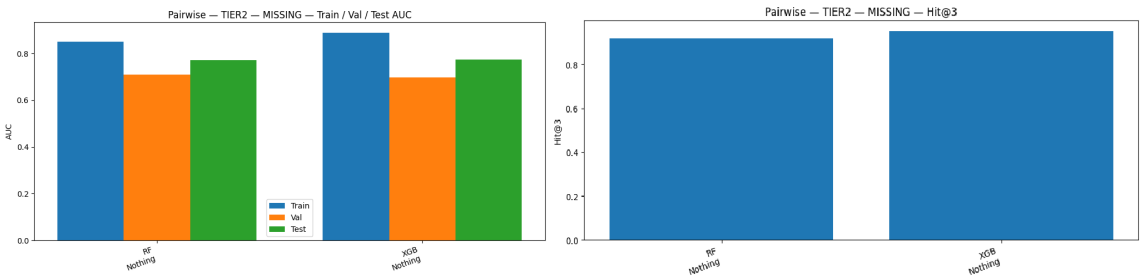


Figure 5.20: Tier B2 missing-target pairwise AUC and Hit@3 for the final inverse-frequency-weighted no-cluster-info configuration. The AUC results show that both models retained useful candidate-discrimination ability in the stricter missing-target setting, while the Hit@3 results show that relevant Tier B2 clusters were often ranked near the top of the candidate list. XGBoost achieved slightly stronger ranking performance than RF, supporting its selection for the final pairwise Stage 1 configuration.

The final model showed a clear distinction between discrimination, thresholded classification, and ranking. AUC remained relatively high, indicating that the model could separate relevant from irrelevant candidates. F1 was more moderate, reflecting

the difficulty of choosing a fixed threshold in an imbalanced pairwise setting. Hit@3 remained high, suggesting that even when the thresholded binary prediction was imperfect, the model usually ranked a relevant candidate near the top. This ranking behavior is particularly important for a recommendation system, where the practical output is a short candidate list rather than a single binary decision.

Interpretation of the ranking metrics. In the pairwise formulation, ranking metrics are evaluated at the *sample* level rather than the candidate-row level. Each leave-one-out sample is expanded into multiple candidate-cluster rows, and a sample is counted as a Hit@k success if a true target cluster appears within the top (k) ranked candidates.

The meaning of this success depends on the target mode. In the `full` setting, Hit@k only indicates that at least one relevant cluster was surfaced; it does not imply recovery of the complete cluster set. In the `missing` setting, however, Hit@k is more directly aligned with the intended use case, since the target is the withheld endpoint cluster. High Hit@k in this setting therefore indicates that the model often ranks the missing cluster among its top recommendations.

Leave-K sensitivity. The final pairwise leave-X-out model was primarily evaluated with $K = 1$, meaning that one secondary endpoint was hidden at a time. To examine whether the model remained stable when more endpoint information was removed, an additional sensitivity experiment was performed for $K \in \{1, 2, 3, 4, 5\}$.

As K increases, each sample has less observed endpoint context and a larger withheld target set. At the same time, the number of possible endpoint combinations grows rapidly. For this reason, the maximum number of sampled withheld-endpoint combinations per protocol was reduced as K increased in order to keep the experiment computationally feasible. Consequently, the leave-K results reported here should be interpreted as a controlled sensitivity analysis rather than a fully exhaustive evaluation over all possible endpoint subsets. This also means that the $K = 1$ results in this subsection are not expected to match exactly the earlier leave-one-out results, since they were produced under a different sampling budget.

K	**Samples**	**Rows**	**Avg. cand.**	**AUC**	**F1**	**Hit@3**
1	286	6006	21.0	0.829	0.398	0.647
2	231	9702	42.0	0.813	0.394	0.732
3	162	10206	63.0	0.829	0.400	0.716
4	98	8232	84.0	0.808	0.363	0.724
5	42	4410	105.0	0.780	0.356	0.690

Table 5.14: Leave-K sensitivity results for the final pairwise Tier B2 missing-target model. The maximum number of sampled withheld-endpoint combinations per protocol was reduced for larger K values to keep the experiment computationally feasible.

The results show that performance decreased gradually rather than collapsing. AUC

and F1 declined as more endpoints were hidden, indicating that the observed endpoint context contributes meaningful signal. However, the ranking metrics remained relatively strong, especially Hit@3, suggesting that the model continued to place relevant candidate clusters near the top of the ranked list even when several endpoints were withheld. This supports the interpretation that the pairwise model is useful as a ranking model, even though exact thresholded classification becomes more difficult as the observed context becomes sparser.

Figure 5.21: Tier B2 missing-target F1 across leave-K settings. Performance decreases gradually as more endpoints are hidden, indicating that observed endpoint context contributes useful signal.

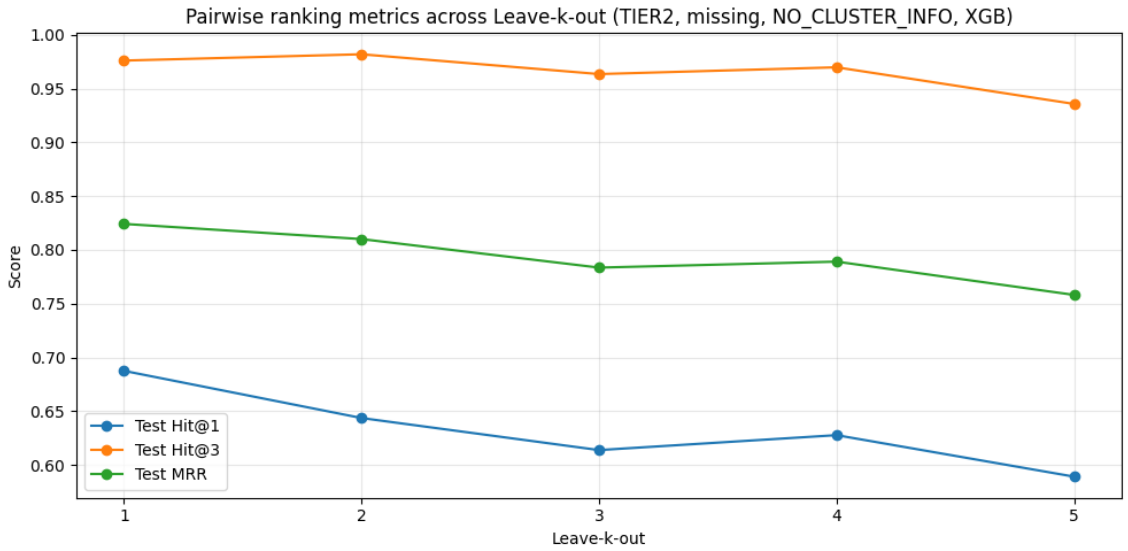


Figure 5.22: Tier B2 missing-target ranking metrics across leave-K settings. Hit@3 remains relatively high even when multiple endpoints are withheld.

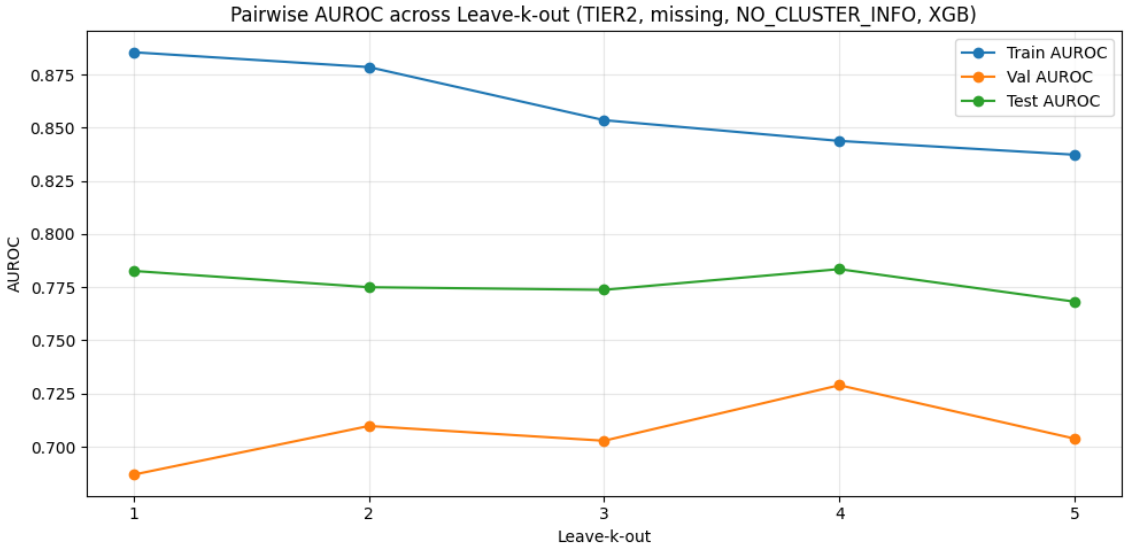


Figure 5.23: Tier B2 missing-target AUROC across leave-K settings. Discrimination performance decreases only gradually as more endpoints are withheld, indicating that the pairwise model retains useful candidate-ranking signal even under increasingly limited observed endpoint context.

The pairwise candidate-scoring formulation was the most useful leave-X-out formulation tested. The initial configuration achieved strong aggregate metrics, but qualitative inspection showed that it was partly biased toward large and frequently occurring endpoint clusters. The final inverse-frequency-weighted configuration addressed this issue by giving rare positive clusters more influence during training. Although this reduced some aggregate metrics, it produced a model that better matched the intended recommendation behavior: ranking clinically relevant and more specific candidate clusters rather than repeatedly selecting only the largest endpoint families.

5.5 Stage 2 Model Performance

Stage 2 evaluated endpoint-level ranking, using the pairwise protocol–endpoint formulation described in Section 4.10.1. This section reports how well the Stage 2 models ranked concrete secondary endpoints under from-scratch and leave-one-out settings.

The endpoint catalogue used for Stage 2 contained 3,700 secondary endpoint candidates from 444 protocols, covering 16 Tier B0 clusters, 61 Tier B1 clusters, and 314 active Tier B2 clusters. The metrics should be interpreted as recorded endpoint recovery rather than complete clinical usefulness, since clinically similar endpoint concepts may differ in wording, timeframe, or source protocol.

Interpretation of endpoint-level metrics. The Stage 2 metrics should be interpreted as measures of recorded endpoint recovery, not as complete measures of

clinical usefulness. A candidate endpoint is counted as correct only when it matches the held-out endpoint from the dataset. However, two endpoint descriptions may be clinically similar while differing in wording, identifier, time frame, or source protocol. Conversely, a candidate labelled as negative may still be clinically plausible if it was not recorded as an endpoint in the historical protocol.

For this reason, false positives (FP) and failed exact hits require some caution when interpreted clinically. The automated metrics are useful for comparing model formulations, but they do not fully capture whether a recommendation would be acceptable after expert review or protocol-specific rewriting.

5.5.1 Feature Representation

The Stage 2 feature construction is described in Section 4.10.1. In the results, the most important point is that the model combined protocol-side context, candidate-endpoint information, hierarchy identifiers, and pairwise code-overlap features. These features allowed the model to rank candidate endpoints not only by their own properties, but also by their compatibility with the target protocol context.

5.5.2 From-Scratch Stage 2

The from-scratch Stage 2 setting is the strict endpoint-level baseline. The model ranks concrete secondary endpoint candidates using protocol, primary-endpoint, and candidate-endpoint information, but without observed secondary-endpoint context. This makes the task more fine-grained than Stage 1 cluster prediction, since many candidate endpoints may be clinically similar while only a small subset are recorded as positives for each protocol.

5.5.2.1 Initial From-Scratch Pairwise Formulation

The initial from-scratch Stage 2 formulation used a small sampled candidate construction, with four negative endpoint candidates for each positive endpoint. Its purpose was to test whether the model could learn any useful protocol-endpoint ranking signal before moving to a larger candidate pool.

This first formulation achieved strong headline metrics on the held-out test split, as shown in Table 5.15. The model reached a pair accuracy of 72.2%, protocol-level micro-F1 of 48.7%, and Hit@3 of 86.5%. These results suggested that the model had learned a meaningful ranking signal under the initial small-candidate construction.

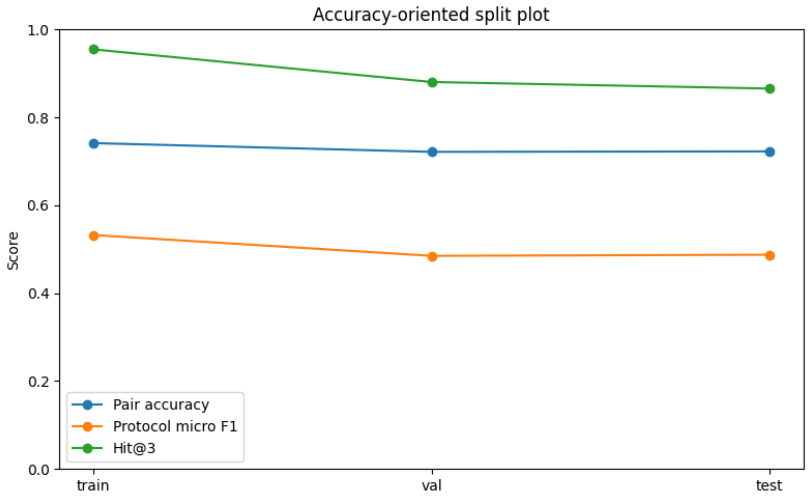


Figure 5.24: Initial from-scratch Stage 2 results using four negative candidates per positive endpoint.

However, qualitative inspection showed that this setup did not fully reflect the intended endpoint-ranking scenario. The model was not expected to choose from only a few endpoint candidates; instead, it would often need to score many candidates drawn from multiple relevant clusters in order to avoid excluding potentially useful endpoints too early. This initial one-positive, four-negative construction therefore made the ranking problem too constrained compared with the intended usage.

The main issue was that the model often behaved as if the task were a small closed-set distinction problem. It could separate the sampled positives from the sampled negatives reasonably well, but the predictions were less reliable when the candidate pool became larger and contained many clinically similar endpoints. In practice, the top-ranked predictions tended to either hit the exact target or miss it, rather than producing a stable ranking of endpoints that were similar in domain, measurement type, timeframe, and clinical meaning. This motivated a revised training construction that kept the same pairwise hard-negative principle, but increased the number of negative candidates per positive endpoint.

5.5.2.2 Revised Deployment-Style Pairwise Construction

The revised from-scratch Stage 2 model used a larger and harder candidate construction, as described in Section 4.10.1. This made exact endpoint recovery more difficult, but better reflected the intended deployment scenario where the model must rank many clinically similar candidates.

The final from-scratch model used XGBoost with the `deployment_hard__n20` construction and the `min2_all` code configuration. The difference between the initial and revised versions was therefore the scale and difficulty of the candidate construction, not the overall modelling formulation. Both versions used pairwise protocol–endpoint classification with hierarchy-aware hard-negative sampling, but the revised version increased the negative pool from 4 to 20 negatives per positive endpoint.

Under this revised construction, the headline metrics changed substantially. Pair accuracy increased to 90.6%, while protocol-level micro-F1 decreased to 21.9% and Hit@3 decreased to 50.0%. The increase in pair accuracy should not be interpreted in isolation, since the revised setting contains many more negative rows and is therefore more affected by class imbalance. The decrease in protocol-level micro-F1 and Hit@3 reflects that exact endpoint recovery became harder when the candidate space was expanded.

Model version	**Candidate construction**	**Pair accuracy**	**Protocol micro-F1**	**Hit@3**
Stage 2 from scratch V1	4 negatives per positive	72.2%	48.7%	86.5%
Stage 2 from scratch V2 / final	20 negatives per positive	90.6%	21.9%	50.0%

Table 5.15: Comparison between the initial and revised from-scratch Stage 2 pairwise formulations.

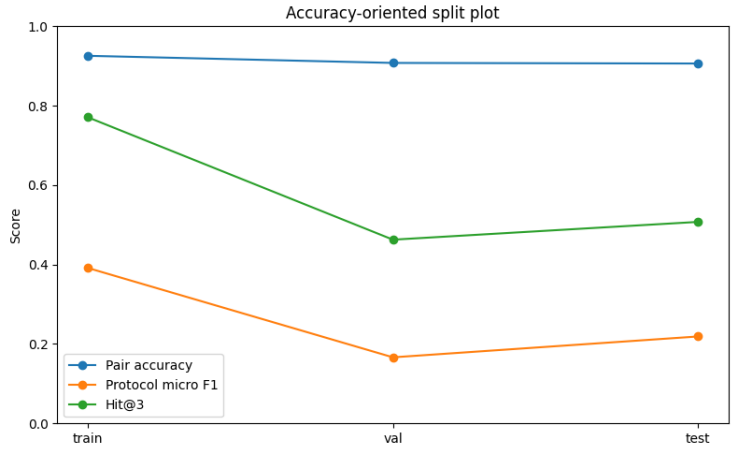


Figure 5.25: Final from-scratch Stage 2 results using 20 negative candidates per positive endpoint.

Although the numerical ranking metrics decreased under the revised setup, manual inspection suggested that the model had still learned useful candidate-level signal. The larger candidate set exposed the model to many more clinically similar alternatives, making exact endpoint recovery harder. However, the highest-ranked predictions were often closer to the target endpoints in clinical domain, measurement type, timeframe, and endpoint description than the raw metrics alone suggested.

Overall, the from-scratch Stage 2 experiments show that endpoint-level recommendation is substantially harder than cluster-level prediction. The initial four-negative setup demonstrated that the model could learn a protocol–endpoint ranking signal, but it also produced an overly constrained evaluation. The revised 20-negative setup was therefore kept as the default Stage 2 construction and was also used in the subsequent leave-K-out experiments.

5.5.3 Leave-K-Out Stage 2

The leave-(K)-out Stage 2 experiments evaluated endpoint ranking under partial secondary-endpoint information, using the formulation described in Section 4.10. The final exported model used leave-one-out as the main setting, while additional leave-(K) sensitivity experiments examined whether performance remained stable when more endpoints were withheld.

Compared with the from-scratch Stage 2 model, the leave-one-out model had access to observed secondary-endpoint context. This made it better aligned with the final intended use case: ranking candidate endpoints for a partially specified endpoint design. The final leave-one-out setting used the `missing` target mode and the same hierarchy-aware hard-negative construction as the final from-scratch model. The resulting row-expanded datasets are summarized in Table 5.16.

Split	**Rows**	**Positive rows**	**Negative rows**
Train	50 736	2 416	48 320
Validation	11 214	534	10 680
Test	10 773	513	10 260

Table 5.16: Pairwise row counts for the final Stage 2 leave-one-out missing-target model.

Table 5.17 compares the final feature composition between the from-scratch and leave-one-out Stage 2 models. The leave-one-out model introduced observed secondary-context features, while the remaining differences reflect that feature vocabularies were fitted on the actual training pair rows and then refined.

Feature block	**From scratch**	**Leave-one-out**
Structured/protocol numeric features	87	88
Protocol code one-hot features	1 053	1 023
Primary-endpoint code one-hot features	972	942
Candidate endpoint code one-hot features	3 492	3 472
Candidate hierarchy one-hot features	1 058	1 053
Pairwise overlap features	36	36
Code-bag size features	9	9
Observed secondary-context features	–	6
Total	**6 707**	**6 629**

Table 5.17: Final feature composition for the from-scratch and leave-one-out Stage 2 models. The leave-one-out setting includes additional observed secondary-context features, while small differences in code and hierarchy feature counts reflect the fitted training-pair vocabulary after pruning and refinement.

5.5.3.1 Two Evaluation Views

The Stage 2 notebook reports the final leave-one-out model using two evaluation views. This distinction is important because the model was trained with sampled hard negatives, but the intended use case requires ranking endpoints from a larger candidate pool.

First, the *sampled hard-negative evaluation* uses the same type of pairwise rows as training and validation. This corresponds to the revised Stage 2 construction described above, where each positive candidate is paired with 20 hierarchy-aware hard negatives. This view measures whether the model has learned a discriminative pairwise signal under the sampled training distribution.

Second, the *expanded candidate-pool evaluation* ranks each leave-one-out sample against the larger available candidate set. This view is closer to the intended endpoint-recommendation scenario, because the model must select the missing endpoint from many plausible alternatives rather than from a small sampled set. In the final evaluation, this corresponded to approximately 128 candidate endpoints per sample. Because the setting was leave-one-out with `target_mode=missing`, each ranked list contained exactly one correct missing endpoint.

5.5.3.2 Sampled Hard-Negative Results

Table 5.18 shows the final sampled hard-negative results. The table reports both candidate-row classification metrics and sample-level ranking metrics, allowing the model to be evaluated both as a pairwise scorer and as a ranked endpoint recommender.

Split	Pair F1	ROC-AUC	PR-AUC	Micro-F1	Hit@1	Hit@3	Hit@5
Train	0.5488	0.9326	0.5911	0.5628	0.6978	0.8777	0.9460
Val	0.4390	0.8391	0.4165	0.4658	0.4677	0.6290	0.7742
Test	0.3879	0.8248	0.3439	0.4186	0.5323	0.6774	0.7742

Table 5.18: Stage 2 leave-one-out performance under the sampled hard-negative pairwise evaluation. The metrics show both row-level classification performance and sample-level ranking performance.

Figure 5.26 summarizes three complementary performance views for the sampled hard-negative evaluation: pairwise accuracy, sample-level Hit@3, and test F1. The high pairwise accuracy should be interpreted cautiously because the pairwise dataset contains many negative candidate rows. Hit@3 is more informative for the ranking use case, since it measures whether a true hidden endpoint appears among the highest-ranked candidates for a sample. The lower F1 reflects the difficulty of converting the ranked probabilities into thresholded binary endpoint decisions under strong class imbalance.

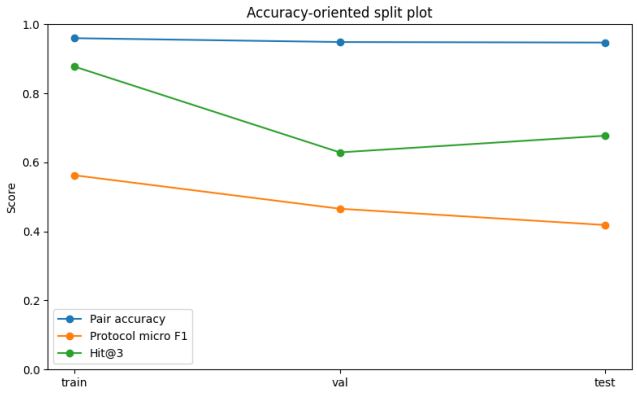


Figure 5.26: Overview of sampled hard-negative Stage 2 performance, showing pairwise accuracy, sample-level Hit@3, and test F1. Pairwise accuracy is high partly because most candidate rows are negative, while Hit@3 better reflects the ranking behavior of the model.

The selected probability threshold is shown in Figure 5.27. The threshold was selected on the validation set to optimize F1, which explains the threshold-dependent F1 value shown in the overview plot.

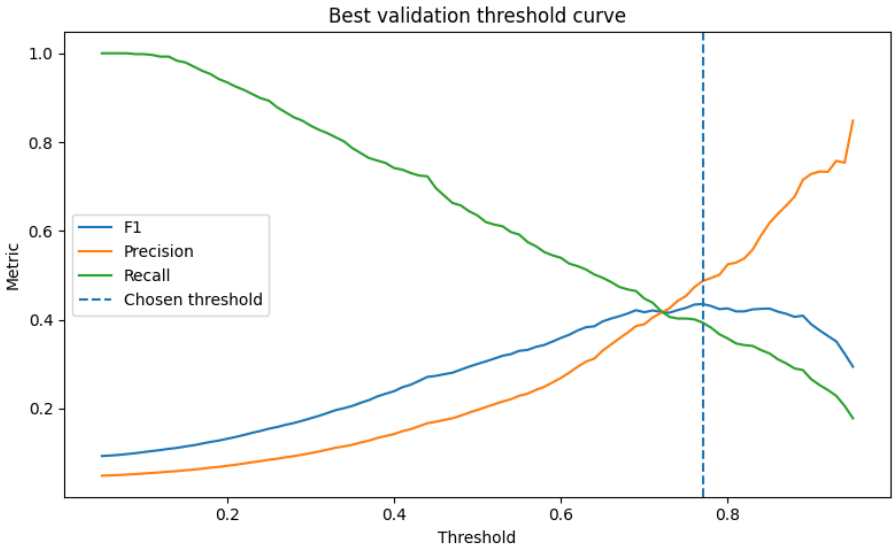


Figure 5.27: Validation threshold sweep for the Stage 2 leave-one-out model. The selected threshold balances precision and recall by maximizing validation F1 under the sampled hard-negative evaluation.

Figure 5.28 further illustrates the model's error profile at the selected threshold. The FP and false negative (FN) distribution shows how thresholded mistakes are distributed after converting predicted probabilities into binary decisions. This diagnostic is useful because the sampled-pair F1 score depends not only on ranking quality, but also on how many candidate endpoints are pushed above or below the selected decision boundary.

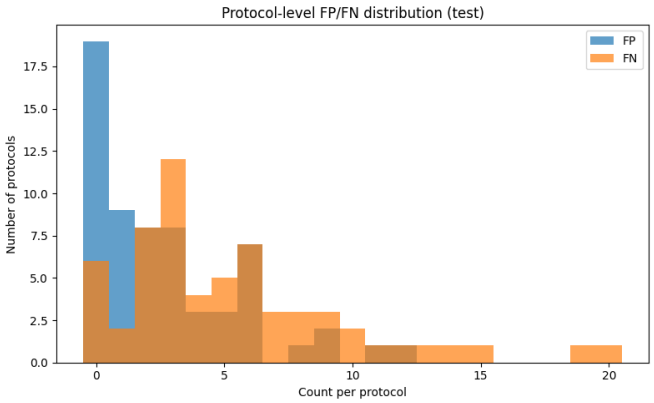


Figure 5.28: FP and FN distribution for the Stage 2 leave-one-out model at the selected threshold. FPs correspond to incorrect candidate endpoints accepted by the threshold, while FNs correspond to hidden endpoints missed by the threshold.

The gap between train and test performance shows that endpoint-level ranking remains difficult, especially under hard negative sampling. However, the validation and test ROC-AUC values are relatively close, suggesting that the model learned a transferable ranking signal rather than only memorizing the training protocols. The lower PR-AUC and F1 reflect the strong class imbalance in the pairwise formulation, where relevant endpoints form only a small fraction of all evaluated protocol-candidate pairs.

To better understand which features influenced the Stage 2 model, two complementary diagnostic views were inspected: standard XGBoost feature importance and mean absolute SHAP importance. The purpose of these plots is not to provide causal explanations, but to indicate which feature groups the fitted model relied on when ranking candidate endpoints.

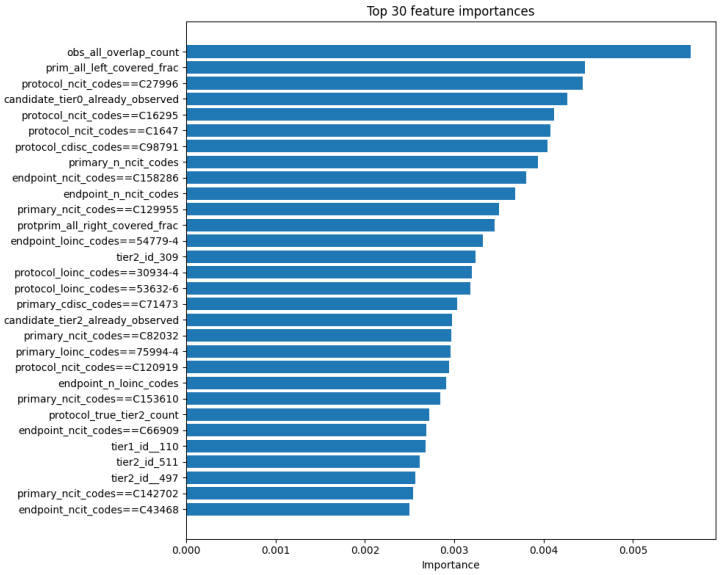


Figure 5.29: Top 30 standard XGBoost feature importances for the Stage 2 leave-one-out model. Selected code translations are provided in Appendix C.1.

The standard XGBoost importance view mainly reflects which features were useful during tree construction. Several high-ranking features were standardized ontology-code indicators, suggesting that the model used clinical code evidence when separating true hidden endpoints from hard negative candidates. However, tree-based importance alone does not show how strongly a feature influenced individual predictions, so it was complemented with the SHAP-based view.

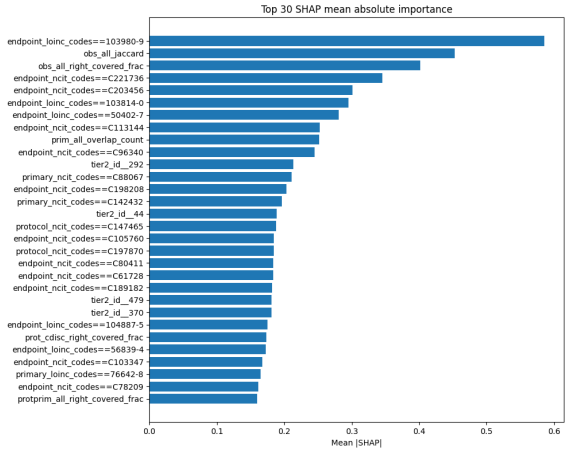


Figure 5.30: Top 30 features by mean absolute SHAP value for the Stage 2 leave-one-out model. Selected code translations are provided in Appendix C.1.1.

The SHAP-based view is more useful for interpreting prediction behavior, because it summarizes which features had the largest average influence on candidate endpoint scores. The strongest features can be grouped into candidate identity and hierarchy features, ontology-code indicators, and overlap features between the observed endpoint context and candidate endpoint. This suggests that the model did not rely only on general protocol metadata, but also used candidate-specific clinical information and pairwise similarity signals when ranking possible hidden endpoints.

Together, the two plots provide a consistency check. Features appearing in both views are more likely to reflect stable signals used by the model, while features appearing mainly in one view should be interpreted more cautiously. The appendix translations are included only for selected interpretable code features and should therefore be treated as examples rather than a complete mapping of all feature names.

5.5.3.3 Expanded Candidate-Pool Evaluation

Table 5.19 reports the second evaluation view of the same trained leave-one-out model. Here, each sample was ranked against the expanded endpoint candidate pool rather than only against the sampled hard negatives used during training.

Split	Samples	Avg. cand.	MRR	Hit@1	Hit@3	Hit@10	Hit@50
Validation	230	127.7	0.3850	0.2870	0.4087	0.5739	0.8435
Test	237	127.7	0.4657	0.3586	0.5148	0.6456	0.8439

Table 5.19: Stage 2 leave-one-out performance under the expanded candidate-pool evaluation.

Candidate-pool recall was 1.0 for both validation and test, meaning that the recorded hidden endpoint was present somewhere in the expanded candidate pool for every evaluated sample. This should be interpreted as a property of the evaluation construction rather than as a prediction result. Since the correct endpoint was always available to rank, the relevant Stage 2 question was how highly the model placed it within the expanded candidate pool.

On the test split, the model achieved an MRR of 0.4657, Hit@1 of 0.3586, Hit@10 of 0.6456, and Hit@50 of 0.8439. This suggests that the model is useful as a prioritization tool: it often places the hidden endpoint within a manageable recommendation list, but it should not yet be interpreted as a fully automatic endpoint selector.

5.5.3.4 Leave-K Sensitivity Experiments

The final pairwise leave-X-out model was primarily evaluated using $K = 1$, meaning that one secondary endpoint was hidden at a time. To examine whether the model remained stable when more endpoint information was removed, an additional leave-K sensitivity experiment was performed for $K \in \{1, 2, 3, 4, 5\}$.

In this setup, larger values of K correspond to increasingly incomplete observed endpoint context. At the same time, the target set also becomes larger, since more withheld endpoints may contribute relevant candidate clusters. The metrics must therefore be interpreted carefully. Pairwise ROC-AUC, PR-AUC, and F1 evaluate row-level candidate discrimination, while Hit@k evaluates whether at least one relevant candidate is ranked within the top-k candidates for each generated sample.

A practical limitation of this experiment is computational complexity. As K increases, the number of possible withheld endpoint combinations grows rapidly. Each generated sample is then expanded into multiple pairwise candidate rows, causing the dataset size to increase substantially. To keep the experiment computationally feasible, the maximum number of sampled combinations per protocol was reduced as K increased. Therefore, the results in Table 5.20 should be interpreted as a controlled sensitivity analysis rather than a fully exhaustive leave-K evaluation over all possible endpoint combinations.

Leave-K	Test samples	Pair rows	Avg. cand.	ROC-AUC	PR-AUC	Pair F1	Hit@1	Hit@3	Hit@10
1	286	6006	21.0	0.829	0.353	0.398	0.427	0.647	0.902
2	231	9702	42.0	0.813	0.353	0.394	0.515	0.732	0.922
3	162	10206	63.0	0.829	0.383	0.400	0.506	0.716	0.914
4	98	8232	84.0	0.808	0.348	0.363	0.571	0.724	0.959
5	42	4410	105.0	0.780	0.307	0.356	0.476	0.690	0.905

Table 5.20: Leave-K sensitivity results for the pairwise candidate-scoring model. As K increases, more endpoints are withheld, which increases the average number of candidate rows per sample. The maximum number of sampled withheld-endpoint combinations per protocol was reduced for larger K values to keep the experiment computationally feasible.

The row-level pairwise discrimination metrics are visualized in Figures 5.31-5.33. These plots show that the model remained relatively stable across the lower leave-K settings. ROC-AUC stayed above 0.80 for $K = 1$ through $K = 4$, while PR-AUC remained in a similar range over the same interval. The largest decrease occurred at $K = 5$, where ROC-AUC dropped to 0.780 and PR-AUC to 0.307. However, this setting also contained the smallest number of test samples, partly because fewer endpoint-combination samples were retained at higher K values. The $K = 5$ result should therefore be interpreted with more caution than the lower-K settings.

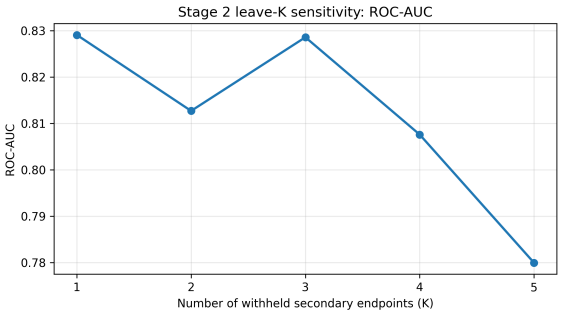


Figure 5.31: Pairwise ROC-AUC across leave-K settings. ROC-AUC remains relatively stable for $K = 1$ through $K = 4$, indicating that the model preserved its ability to discriminate relevant from non-relevant candidate clusters under increasingly incomplete endpoint context.

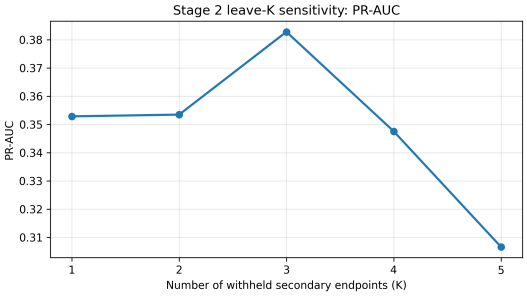


Figure 5.32: Pairwise PR-AUC across leave-K settings. PR-AUC is more sensitive to class imbalance than ROC-AUC, but remains broadly stable for the lower leave-K settings before decreasing at $K = 5$.

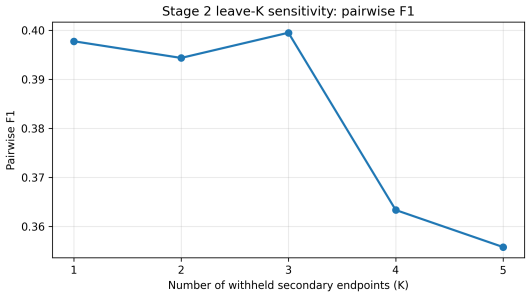


Figure 5.33: Pairwise F1 score across leave-K settings. The F1 score shows only moderate variation across the sensitivity experiment, suggesting that the classification threshold retained reasonable precision-recall balance as additional endpoints were hidden.

The ranking metrics are shown in Figure 5.34. Hit@10 stayed above 0.90 for every value of K, indicating that the model usually ranked at least one relevant candidate among the top ten predictions even when several endpoints were hidden. Hit@1 and Hit@3 did not decrease monotonically as K increased. This is expected, since larger K values produce larger target sets: when more endpoints are withheld, there are more relevant candidates that can count as a hit. Therefore, Hit@k should not be interpreted as becoming easier or harder solely from the value of K; it reflects both model ranking quality and the number of true candidates available per sample.

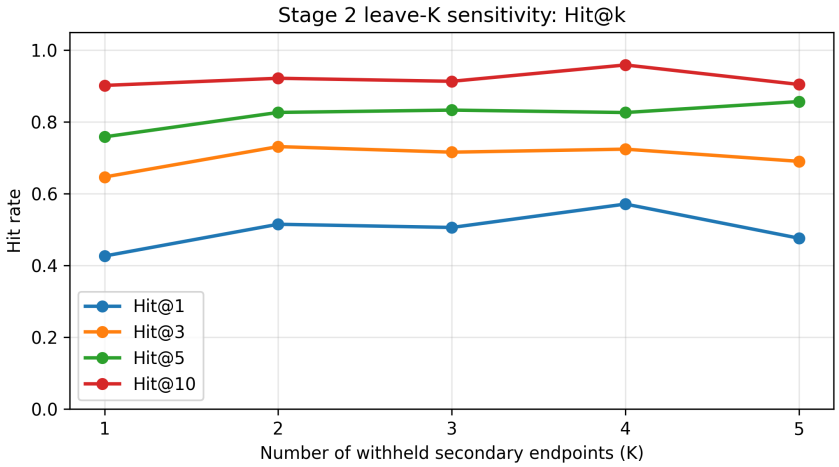


Figure 5.34: Hit@k ranking performance across leave-K settings. Hit@10 remains high across all settings, showing that relevant candidate clusters are usually retained within the top-ranked predictions. The non-monotonic behavior of Hit@1 and Hit@3 should be interpreted in light of the increasing number of withheld endpoints and therefore the increasing number of relevant candidates per sample.

Overall, the sensitivity experiment suggests that the pairwise model does not depend exclusively on having almost complete endpoint context. Even when several endpoints were hidden, the model retained useful candidate-discrimination ability and continued to rank relevant candidates highly. At the same time, the experiment was not a full combinatorial evaluation of all possible leave-K subsets. The results should therefore be viewed as evidence of robustness under increasingly incomplete endpoint context, rather than as definitive performance estimates for every possible leave-K configuration.

5.6 Full Pipeline Evaluation

The full-pipeline evaluation applied the exported Stage 1 and Stage 2 models end to end on previously unseen Tier C protocols, using the inference workflow described in Section 4.10.3. The purpose was not strict exact-string recovery, but qualitative inspection of whether the system could surface clinically meaningful endpoint concepts outside the Tier B model-development cohort.

The final evaluation used 40 unseen Tier C protocols. For each protocol, the first secondary endpoint was hidden, the remaining secondary endpoints were retained as observed context, and the system generated ranked candidate endpoint recommendations. In total, the run produced 61,022 scored candidate endpoint pairs and 986 top-ranked endpoint rows.

Because the hidden Tier C endpoints were not exact candidates from the trained endpoint catalogue, the output was assessed by manual inspection rather than by strict exact-match metrics. The main question was whether the ranked recommendations

were clinically plausible, redundant with observed endpoints, misaligned with the protocol, or potentially useful as missing endpoint concepts.

The full-pipeline evaluation should therefore be viewed as an early end-to-end diagnostic. The internal Stage 2 experiments showed that the endpoint ranker learned useful candidate-level signal when relevant candidates were available in the candidate pool. The Tier C run tested a harder and more realistic situation: applying the complete system to unseen protocols and inspecting whether the generated recommendations remained meaningful at the endpoint-concept level. The main limitation is that this type of evaluation cannot be reduced to exact-match metrics without a curated expert-labelled relevance set.

For this reason, the full-pipeline results are complemented with a qualitative case study from one unseen Tier C protocol.

5.6.1 Example Prediction from One Unseen Protocol: NCT05768230

To illustrate the full-pipeline behavior at the endpoint-concept level, one protocol was selected at random from the 40 unseen Tier C protocols described in Section 5.6. The selected protocol, `NCT05768230`, was used as a qualitative case study of how the full pipeline behaves when applied to a previously unseen protocol and evaluated by manual inspection rather than exact-string recovery.

The protocol investigates whether levosimendan can improve right ventricular function in mechanically ventilated patients with acute respiratory distress syndrome (ARDS) complicated by right ventricular dysfunction. The intervention is motivated by the hypothesis that levosimendan may improve right ventriculo-pulmonary artery coupling by increasing right-heart contractility and reducing pulmonary vascular resistance. The study uses transesophageal echocardiography (TEE) to assess right-heart function, pulmonary circulation, hemodynamic status, organ failure, and mortality.

In the leave-one-out setup, the first secondary endpoint was hidden from the protocol and the remaining endpoints were retained as observed context. The observed endpoint context was highly focused on right ventricular function and acute hemodynamic status, including tricuspid annular plane systolic excursion (TAPSE), tricuspid annular systolic velocity, right/left ventricular end-diastolic area (RVEDA/LVEDA) ratio, pulmonary vascular resistance, cardiac index, $ScvO_2$, systemic vascular resistance, SOFA score, and mortality outcomes. The hidden endpoint in this example was: *Right ventricular area fractional change (RV FAC) 48 hours after randomization.* This endpoint is clinically close to several of the observed measurements, but it is not identical to them. RV FAC measures the fractional change in right ventricular area between diastole and systole and is an echocardiographic marker of right ventricular systolic function. It therefore belongs to the same physiological endpoint family as TAPSE and tricuspid annular systolic velocity, but captures a distinct right ventricular functional measurement. Figure 5.35 summarizes the clinical setting of the selected protocol.

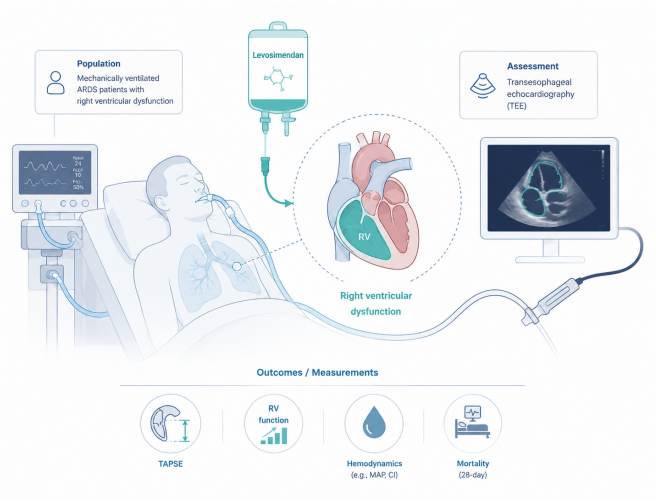


Figure 5.35: Qualitative summary of protocol NCT05768230. The study evaluates levosimendan in mechanically ventilated ARDS patients with right ventricular dysfunction, using transesophageal echocardiography and hemodynamic measurements to assess right-heart function and clinical outcomes.

Rank	Predicted endpoint concept	Manual annotation
1	right ventricular function	**Relevant**
2	Change From Baseline in RV Function as Measured by Tricuspid Annular Plane Systolic Excursion (TAPSE) as Assessed by Echocardiograms at 12 Weeks	**Relevant**
3	Right Ventricular systolic function	**Relevant**
4	RV function	**Relevant**
5	Pulmonary function / Cardiopulmonary Exercise	**Relevant**
6	Effect of Potassium Nitrate (KNO_3) on Left Ventricle (LV) Diastolic Function: Left Atrial Volume Index	**Wrong context**
7	**Change From Baseline in Right Ventricular Fractional Area Change (RVFAC) at Week 24**	**Relevant**
8	Change in left ventricular diastolic function at 6 and 12 months	**Partially relevant**
9	Effect of Potassium Nitrate (KNO_3) on Left Ventricle (LV) Diastolic Function: E/e' Ratio	**Wrong context**
10	Assessment of left ventricular diastolic function in HFpEF	**Partially relevant**

Table 5.21: Manually annotated top-ranked endpoint concepts for protocol NCT05768230. Green entries indicate clinically relevant right-heart or pulmonary-function concepts, yellow entries indicate partially relevant but less directly aligned concepts, and red entries indicate candidates judged to belong to the wrong clinical context. The hidden endpoint concept, RVFAC, appeared among the ranked candidates, but with wording and timeframe inherited from another historical protocol.

Table 5.21 shows the top-ranked endpoint concepts returned by the full pipeline for this protocol. The predictions were manually annotated after model inference to support qualitative interpretation. Green entries indicate clinically relevant or clearly aligned endpoint concepts, yellow entries indicate partially relevant but less directly aligned concepts, and red entries indicate candidates judged to belong to the wrong clinical context. These annotations were not used during model training or evaluation.

The annotated predictions show that the pipeline retrieved several concepts from the correct physiological neighborhood. The highest-ranked candidates were mostly related to right ventricular function, including general right ventricular function, TAPSE-based assessment, right ventricular systolic function, and RV function. These predictions are not exact matches to the hidden endpoint, but they are clinically aligned with the protocol's right-heart dysfunction setting.

The most important prediction was the RVFAC candidate at rank 7. Although its wording and timeframe came from another historical protocol, the underlying endpoint concept matches the hidden endpoint concept. This illustrates the main limitation of exact-string evaluation for the full pipeline: a historical endpoint candidate may capture the correct clinical measurement idea while still differing from the target protocol in timeframe, wording, disease context, or intervention context. In this example, an RVFAC endpoint measured at Week 24 is not directly appropriate for an acute ARDS protocol where the relevant assessment window is 48 hours, but the measurement concept itself is highly relevant.

The table also shows why post-processing is needed. Several candidates were redundant or only partially distinct from the observed endpoint context, such as broad RV function and TAPSE-related suggestions. Other candidates were cardiology-related but clinically misaligned with this specific protocol, especially the left-ventricular diastolic-function endpoints related to left atrial volume index, E/e' ratio, and HFpEF. These predictions indicate that the pipeline identified the broader cardiovascular domain, but could still confuse nearby cardiac subdomains when the candidate pool contained semantically related but physiologically different endpoint types.

Overall, this case study suggests that the full pipeline can retrieve clinically meaningful endpoint concepts on unseen protocols, even when exact endpoint recovery is not an appropriate evaluation criterion. The model was not simply returning arbitrary endpoints; it surfaced several right-heart and pulmonary-function concepts, including the hidden RVFAC concept. At the same time, the raw ranked output contained duplicate, already-covered, partially overlapping, and wrong-context candidates. This motivates a post-processing layer that can distinguish between already-covered endpoint ideas, genuinely useful missing concepts, and clinically misaligned recommendations.

Such a layer would not simply copy endpoint text from historical protocols. Instead, it would interpret the recommended candidates as endpoint concepts and adapt useful concepts to the protocol-specific measurement method, clinical objective, and time frame. Additional qualitative examples are provided in Appendix D.1. These examples are not used as quantitative performance estimates, but illustrate whether

the same qualitative pattern appears across other unseen protocols.

5.6.2 Expert Review #2: Qualitative Assessment of Full-Pipeline Predictions

Following the full-pipeline evaluation, a second expert review was conducted with Niklas Bergh to assess the qualitative behavior of the generated endpoint recommendations. The purpose of this review was not to re-score the model using exact-match metrics, but to understand whether the predicted endpoints were clinically meaningful, whether they captured relevant endpoint domains for unseen protocols, and what kinds of errors remained in the raw ranked outputs.

The review focused on predictions from unseen Tier C protocols. Overall, the expert feedback was positive. A majority of the reviewed endpoint recommendations were considered to be in the correct clinical domain and reflected relevant endpoint categories for the protocols being evaluated. This suggests that, even though the full pipeline did not reproduce exact hidden endpoint strings, the recommendations often contained meaningful clinical signal. Since the Tier C protocols were unseen and not part of the Tier B training set, the more relevant interpretation is whether the model can surface clinically related endpoint concepts, rather than whether it can reproduce the exact hidden endpoint text.

The main limitation identified in the review was not that the predictions were generally irrelevant, but that they sometimes lacked sufficient protocol-specific precision. The model often captured the *domain* or broader clinical concept of the protocol, while the exact endpoint wording, time frame, molecule-specific context, or measurement emphasis was not always fully aligned with the target protocol. In this sense, the predictions were frequently clinically plausible, but not always specific enough to be considered direct replacements for the held-out endpoints.

Niklas emphasized the distinction between *domain relevance*, *conceptual novelty*, and *protocol specificity*. Many predicted endpoints belonged to the right clinical area and were reasonable from a trial-design perspective, but they did not always add new information to the partially observed endpoint design. Some recommendations overlapped conceptually with endpoints already present in the observed context. For example, in protocols where right ventricular function was already represented through TAPSE, tricuspid annular velocity, or related echocardiographic measures, the model could still recommend additional right-ventricular-function endpoints. These recommendations were not clinically wrong; rather, they showed that the model had identified the correct endpoint domain, but had not always distinguished between an already-covered concept and a genuinely missing or additive endpoint idea.

A related issue was that endpoint candidates inherited details from the historical protocols where they originally appeared. A candidate endpoint could include molecule-specific wording, a mismatched time frame, or a measurement focus that was relevant in the source protocol but less central in the target protocol. Similarly, endpoints with similar surface forms could play different roles in different studies

depending on the intervention, population, disease mechanism, and assessment window. This means that endpoint usefulness is not determined only by semantic similarity or terminology-code overlap, but also by how well the endpoint fits the specific clinical objective of the target protocol.

This helps explain the discrepancy between the quantitative full-pipeline evaluation and the qualitative review. Exact recovery of a held-out endpoint is a strict criterion: a prediction can be clinically relevant, belong to the correct endpoint family, and still fail exact-match evaluation if it differs in wording, time frame, or protocol-specific context. The expert review therefore indicated that the full pipeline performed better qualitatively than exact-hit results alone suggested.

Niklas described this with the analogy of a recipe: the task is similar to being given a cake recipe with one ingredient removed and then trying to infer the missing ingredient. Once the title, study description, and other protocol-level fields are already written, the intended endpoint structure is often implicitly encoded in the protocol. This makes the prediction task somewhat artificial in a strict leave-one-out sense, since the surrounding protocol text is usually written in relation to the planned endpoints. At the same time, the analogy also highlights that the model was often able to infer the right type of missing ingredient, even when it did not reproduce the exact one.

Taken together, the expert review suggests that the full pipeline learned meaningful domain-level endpoint structure, but that raw endpoint retrieval is not sufficient on its own. The generated recommendations were often clinically reasonable and aligned with the correct broad objective of the protocol, but many were not directly usable without further interpretation. The remaining gap was therefore not only one of predictive accuracy, but also one of endpoint utility: distinguishing already-covered concepts from genuinely additive endpoint ideas, separating relevant clinical concepts from protocol-specific endpoint wording, and adapting useful endpoint concepts to the target protocol.

This further motivates a post-prediction layer. Such a layer would not simply rerank historical endpoint strings, but would interpret the retrieved candidates as endpoint concepts. Its role would be to filter redundant or already-covered ideas, remove clinically misaligned candidates, and adapt useful concepts to the protocol-specific measurement method, clinical objective, and time frame. In this interpretation, the model output is best understood as a ranked set of candidate endpoint ideas for expert review, rather than as final endpoint text ready to be inserted into a protocol.

5.7 Future Work

The results of this thesis suggest that historical clinical trial data can be structured into meaningful endpoint groups and used for endpoint-oriented recommendation. At the same time, the experiments and expert reviews also showed that the usefulness of such a system depends strongly on the quality of the endpoint hierarchy, the specificity of the clusters, and the degree to which the recommendations are adapted to the clinical context of the target protocol.

A recurring theme throughout the project was that the pipeline is highly cluster-dependent. Both Stage 1 and Stage 2 rely on the assumption that endpoint clusters are coherent, distinguishable, and clinically meaningful. If clusters are too broad, the model tends to produce generic recommendations. If clusters are too small or poorly represented, they become difficult to learn reliably. Future work should therefore focus not only on improving model performance, but also on improving the clinical structure and usefulness of the endpoint space itself.

5.7.1 Clustering

One important direction is to further improve the endpoint clustering strategy. The current hierarchy provides a useful coarse-to-fine structure, but the downstream results showed that cluster quality directly affects prediction quality. Future clustering work should therefore focus on improving cluster homogeneity, reducing ambiguous overlap between clusters, and making better use of noise and outlier endpoints.

Outlier endpoints should not necessarily be treated only as failed clustering cases. In clinical trial design, rare or unusual endpoints may still be highly relevant, especially in specialized studies. A future clustering strategy could therefore separate between true noise, rare but meaningful endpoint types, and endpoints that should be merged into broader nearby groups. This would make the hierarchy more complete while still preserving interpretability.

A further direction is to involve domain experts more directly in the construction or validation of higher-level clusters. Instead of relying entirely on unsupervised clustering, the upper tiers could be curated manually or semi-manually, with embedding-based or LLM-assisted methods used to assign endpoints into the resulting structure. This could produce clusters that are better aligned with clinical reasoning and easier for downstream models to learn.

5.7.2 Modelling

Future modelling work should focus on improving both cluster prediction and endpoint-level ranking. The full-pipeline evaluation showed that Stage 2 can learn useful candidate-level signal, but that end-to-end performance is limited when the correct cluster is not included in the Stage 1 candidate pool. This suggests that future models should place greater emphasis on candidate-pool recall, calibration of cluster probabilities, and more robust handling of rare clusters.

Another important direction is to reduce the model's tendency to favor large or high-support clusters when uncertain. The frequency-corrected pairwise experiments showed that cluster imbalance affects recommendation behavior even when aggregate metrics appear strong. Future models should therefore consider cluster-aware loss functions, label-balanced sampling, or ranking objectives that more directly reward clinically useful recommendations rather than only common-cluster recovery.

Finally, the endpoint ranking stage could be extended with reranking methods that incorporate semantic similarity, protocol specificity, time frame compatibility, and

expert-defined relevance criteria. This would help distinguish between endpoints that are broadly domain-relevant and endpoints that are specifically appropriate for the target protocol.

5.7.3 Data Preparation

The expert review highlighted that endpoint specificity depends on more than endpoint wording alone. Time frame, population, intervention mechanism, disease context, and measurement intent all affect whether an endpoint is appropriate for a protocol. Future data preparation should therefore represent these factors more explicitly.

In particular, time frame information should be standardized more carefully. Endpoints that appear semantically similar may have different clinical meanings depending on whether they measure acute response, medium-term change, or long-term outcome. Similarly, intervention and mechanism information could be represented more directly, since a clinically relevant endpoint often depends on how the intervention is expected to affect the patient population.

Improved data preparation could also include stronger normalization of endpoint wording, separation of molecule-specific text from reusable endpoint concepts, and more detailed mapping between protocol objectives and endpoint measurements. This would make it easier for the model to learn endpoint relevance rather than only surface-level similarity. Cleaner upstream representations would also reduce the burden on the post-prediction layer discussed in Section 5.6.2, since retrieved candidates would already be more consistent with the target protocol's clinical context and require less correction, filtering, or rewriting after prediction.

5.7.4 Pre-Protocol and Post-Protocol Recommendation

The expert reviews also suggested that endpoint recommendation can be framed in two related, but distinct, ways: post-protocol recommendation and pre-protocol recommendation. The system developed in this thesis sits between these two settings. It uses protocol-level information and observed endpoint context, but it is also intended to support endpoint reasoning rather than merely reproduce endpoints that are already implied by a nearly complete protocol.

A shared requirement for both settings is a more clinically grounded endpoint hierarchy. The experiments showed that the pipeline is highly dependent on how endpoints are grouped. If the clusters are too broad, the model tends to return generic domain-level suggestions. If they are too narrow or poorly represented, the model struggles to learn them reliably. Future versions of the system would therefore benefit from clinician-in-the-loop clustering, not only to improve cluster quality in general, but to align the hierarchy with the intended recommendation setting. In a post-protocol setting, clusters may need to distinguish already-covered concepts from genuinely missing endpoint ideas. In a pre-protocol setting, higher-level endpoint categories may instead need to be organized around clinically meaningful design

dimensions such as intervention mechanism, patient population, disease process, measurement purpose, and time frame.

This would allow the model to reason less from surface-level endpoint similarity alone and more from clinically relevant structure. For example, endpoints could be grouped not only by whether they mention cardiac function, renal function, or mortality, but also by why the measurement is relevant in a given protocol: acute hemodynamic response, long-term disease progression, safety monitoring, treatment efficacy, or mechanism-specific pharmacodynamic effect.

5.7.4.1 Post-Protocol Endpoint Recommendation

The post-protocol setting remains useful, but the expert review showed that raw model predictions should not be treated as final endpoint recommendations. Many top-ranked candidates may be clinically related while still being redundant, too generic, or written for another protocol context. Future work should therefore focus on a post-prediction review layer that converts retrieved endpoint candidates into more useful protocol-specific recommendations.

One possible approach is an LLM-based review step after Stage 2. A small exploratory test of this idea was performed on the `NCT05768230` case study discussed in Section 5.6.1. Although this was not a controlled quantitative experiment, the qualitative result was promising: the LLM was able to filter a noisy top-10 prediction list down to the single most useful non-duplicate endpoint concept. Instead of returning the ranked endpoint candidates directly, such a layer could separate predictions into three groups:

1. endpoint concepts that are already covered by the observed protocol endpoints,
2. endpoint concepts that are clinically misaligned or irrelevant,
3. endpoint concepts that are potentially new and clinically useful.

The purpose of this layer would not be to replace clinical judgement, but to make the model output easier to interpret. Stage 2 retrieves endpoint candidates from historical protocols, and these candidates naturally carry their original wording, time frame, intervention context, and disease context. This matches the expert-review observation in Section 5.6.2: a clinically relevant candidate may look superficially wrong because it inherits wording, time frame, molecule-specific context, or measurement emphasis from another protocol. Conversely, a candidate may look plausible but add little new information because the same endpoint concept is already present in the observed endpoint set. The case study in section 5.6.1 using protocol NCT05768230 illustrates this idea. In that example, the hidden endpoint was:

> *Right ventricular area fractional change (RV FAC) 48 hours after randomization.*

The full pipeline returned a top-10 list containing several right-ventricular-function concepts, but many of these were already conceptually covered by the observed endpoint set. For example, the observed protocol already contained endpoints related to TAPSE, tricuspid annular systolic velocity, RVEDA/LVEDA, pulmonary vascular

resistance, and hemodynamic status. Therefore, several of the model's right-heart predictions were clinically relevant but not necessarily additive.

To make the post-processing step more transparent, Table 5.22 shows the LLM-based filtering outcome for the top-10 predicted endpoint concepts for protocol `NCT05768230`. In this step, the true hidden endpoint was not shown to the LLM. The prompt was structured as a two-step review task: first, the LLM was given the protocol context, the observed endpoint set, and the top-10 predicted candidates, and was asked to classify each candidate as redundant, clinically misaligned, or potentially useful. The true hidden endpoint was not shown. Second, the retained candidate was rewritten into a protocol-specific endpoint proposal. Most candidates were filtered out because they were either redundant with the observed endpoint context or less well aligned with the protocol. Only one candidate was retained for further consideration: a historical RVFAC endpoint.

Rank	**Predicted endpoint concept**	**LLM review decision**
1	right ventricular function	**Discarded**
2	Change From Baseline in RV Function as Measured by Tricuspid Annular Plane Systolic Excursion (TAPSE) as Assessed by Echocardiograms at 12 Weeks	**Discarded**
3	Right Ventricular systolic function	**Discarded**
4	RV function	**Discarded**
5	Pulmonary function / Cardiopulmonary Exercise	**Discarded**
6	Effect of Potassium Nitrate (KNO_3) on Left Ventricle (LV) Diastolic Function: Left Atrial Volume Index	**Discarded**
7	**Change From Baseline in Right Ventricular Fractional Area Change (RVFAC) at Week 24**	**Kept**
8	Change in left ventricular diastolic function at 6 and 12 months	**Discarded**
9	Effect of Potassium Nitrate (KNO_3) on Left Ventricle (LV) Diastolic Function: E/e' Ratio	**Discarded**
10	Assessment of left ventricular diastolic function in HFpEF	**Discarded**

Table 5.22: LLM-based filtering outcome for the top-10 predicted endpoint concepts for protocol `NCT05768230`. Red entries were discarded during post-processing because they were judged redundant, insufficiently specific, or less well aligned with the protocol context. The single green entry was retained as the most useful non-duplicate concept for potential rewriting.

As raw text, this candidate was not a direct match to the target protocol because the wording and time frame came from another study. However, the underlying endpoint concept was highly relevant and not already fully represented by the observed endpoints. In a second step, the LLM was therefore asked to rewrite this retained endpoint concept into the style, measurement context, and time frame of

the target protocol. Without seeing the hidden endpoint, it produced the endpoint proposal shown in Table 5.23.

LLM-rewritten endpoint proposal	
Endpoint component	**Protocol-specific rewritten content**
Measure	Right ventricular fractional area change (RV FAC) 48 hours after randomization.
Description	RV FAC is an effective ultrasound method for evaluating right ventricular systolic function. The right ventricular end-diastolic area and right ventricular end-systolic area are measured by echocardiography, and RV FAC is calculated as the percentage change in right ventricular area between diastole and systole.
Time frame	At the time of 48 hours after enrollment.

Table 5.23: Protocol-specific endpoint proposal generated by the LLM after filtering the top-10 predicted candidates. The hidden endpoint was not provided to the LLM.

The rewritten endpoint was extremely close to the actual hidden endpoint shown above, despite the LLM never being given that endpoint. The retained historical candidate originally referred to RVFAC at Week 24, but after filtering and rewriting, the LLM adapted it to the target protocol's acute 48-hour assessment window. The resulting measure differed from the hidden endpoint only in minor wording, using "right ventricular fractional area change" rather than "right ventricular area fractional change", while preserving the same RV FAC concept and time frame. The important result is therefore not only that RV FAC appeared somewhere in the model's top-10 predictions, but that a post-processing layer was able to identify it as the only clearly useful non-redundant concept and adapt it into a protocol-specific endpoint proposal.

This suggests that the most useful post-protocol system may not be one that simply returns historical endpoint strings. Instead, the model could retrieve relevant endpoint concepts, after which a post-processing layer would filter redundant candidates, remove misaligned suggestions, and rewrite the remaining useful concepts into protocol-specific endpoint proposals.

A practical workflow could therefore be:

Step	**Role in the proposed post-protocol workflow**
1	Retrieve historically relevant endpoint concepts from similar protocols.
2	Remove predictions that are redundant, clinically misaligned, or insufficiently specific.
3	Rewrite the retained candidate concepts into protocol-specific endpoint proposals.

Table 5.24: Proposed post-protocol workflow combining candidate retrieval, filtering, and protocol-specific endpoint rewriting.

A further limitation is that the data used in this thesis comes from ClinicalTrials.gov, where the available record generally represents the final uploaded or registered trial information rather than the full evolution of the protocol over time. As a result, the leave-one-out setup used in this thesis simulates missing endpoints by removing information from an otherwise completed protocol. This is useful for evaluation, but it is not the same as observing how endpoint decisions were actually made during protocol development.

A more informative post-protocol setting would require access to protocol version history. For example, if versions V_1, V_2, and V_3 of a protocol were available, it would be possible to study which endpoints were added, removed, or modified over time, and how these changes related to updates in the title, objectives, inclusion criteria, intervention description, time frame, or clinical rationale. This would allow the model to learn from real amendment patterns rather than from artificially hidden endpoints.

This distinction is important. In the current setup, the model is asked to infer an endpoint from a protocol that may already contain indirect textual evidence of that endpoint. In a version-based setup, the model could instead learn when a protocol appears incomplete, which endpoint categories tend to be added later, and what types of protocol changes are associated with endpoint amendments. Such data would better match the practical post-protocol use case, where the goal is to support review of whether the current endpoint strategy is sufficiently complete and aligned with the protocol's objectives.

5.7.4.2 Pre-Protocol Endpoint Recommendation

The pre-protocol setting was identified as a particularly interesting alternative use case. In contrast to the post-protocol setting, the user would not begin with a nearly complete protocol and ask which endpoint might be missing. Instead, the system would support endpoint strategy earlier in the design process, before the endpoint list has been finalized.

This setting changes the role of the recommendation system. Rather than recovering a hidden endpoint from an already written protocol, the goal would be to narrow the design space by suggesting clinically appropriate endpoint families, commonly used measurements, and comparable historical trial strategies. This may be more aligned with practical protocol design, where endpoint decisions are shaped gradually from the study objective, clinical rationale, and intended evidence strategy.

A key difference from the current thesis pipeline is that the endpoint hierarchy would likely need to be more explicitly clinically structured. The hierarchy used in this thesis was derived primarily from endpoint text, ontology codes, and protocol context. This was useful for discovering broad endpoint groupings, but it did not always capture the reasoning that determines endpoint relevance in practice. In a pre-protocol recommendation setting, the hierarchy could instead be organized around clinically meaningful decision dimensions, such as disease process, mechanism of action, measurement purpose, patient population, and time frame. For example, a clinically informed hierarchy could be organized as follows:

T0: Disease or therapeutic area

T1: Intervention mechanism or biological pathway

T2: Endpoint purpose or clinical outcome domain

T3: Specific measurement, instrument, or time-frame-specific endpoint

A simplified example is shown below:

Tier	**Example cluster**	**Clinical meaning**	**Example endpoints**
T0	Stroke / thromboembolic disease	Broad disease or therapeutic area	Stroke recurrence; thromboembolic events; functional recovery after stroke
T1	Anticoagulation / blood thinning mechanism	Intervention mechanism expected to reduce clot formation	Ischemic stroke recurrence; major bleeding; systemic embolism
T2	Safety outcomes	Endpoints measuring treatment risk rather than efficacy	Major bleeding events; clinically relevant non-major bleeding; intracranial hemorrhage
T3	Time-frame-specific bleeding endpoint	Specific safety measurement adapted to follow-up duration	Major bleeding through 30 days; major bleeding through 6 months; time to first major bleeding event

Table 5.25: Illustrative example of a clinically structured endpoint hierarchy for pre-protocol recommendation.

This type of structure would allow the system to make more specific recommendations than a hierarchy based only on endpoint wording or ontology overlap. Two endpoints that appear similar on the surface could belong to different parts of the hierarchy depending on their clinical role. For example, a bleeding endpoint in an anticoagulation trial may be a central safety endpoint, while a superficially similar bleeding measurement in another setting may only be a secondary adverse event. Similarly, a six-minute walk test may be highly relevant in a heart failure or functional-capacity study, but much less central in a trial primarily focused on stroke prevention.

Time frame provides another example of why endpoint similarity cannot be judged from wording alone. An endpoint such as hospitalization, mortality, biomarker change, or functional improvement can have different clinical meaning depending on whether it is measured acutely, after several weeks, or over long-term follow-up. In a clinically guided hierarchy, these distinctions could be represented explicitly, rather than left for the model to infer indirectly from endpoint text.

This would make the system useful as an early design-support tool. Instead of returning a list of historical endpoint strings, it could help identify comparable endpoint strategies and organize them into clinically meaningful recommendation categories. The output would not need to be a final endpoint list. A more realistic

output would be a structured overview of relevant endpoint families, commonly used measurements, and protocol-specific considerations that can support expert decision-making.

In this sense, pre-protocol recommendation changes the question from "which endpoint is missing from this almost complete protocol?" to "which endpoint strategy is appropriate for this planned study?" Compared with the post-protocol setting, the pre-protocol setting would place less emphasis on recovering a hidden endpoint and more emphasis on narrowing the clinical design space.

This distinction helps clarify the role of the present study. The current work should be viewed as an exploratory foundation: it investigates whether endpoint data can be standardized, clustered, and used for recommendation at all. The expert feedback suggests that this direction is meaningful, but that future systems may need to combine data-driven structure with stronger clinical guidance in how endpoint categories are defined, and with post-processing methods that turn retrieved endpoint concepts into protocol-specific recommendations.

6

Conclusion

This study investigated whether historical heart-failure clinical trial data can be transformed into a structured and terminology-aware representation that supports secondary-endpoint recommendation. The study developed an end-to-end proof-of-concept pipeline that starts from raw ClinicalTrials.gov protocol records, reduces them to a heart-failure-focused Phase II-III cohort, separates protocol and endpoint information, standardizes clinical text against biomedical vocabularies, organizes heterogeneous secondary endpoints into a reviewed hierarchy, and uses the resulting representations for two-stage endpoint recommendation.

The first research question concerned how unstructured and heterogeneous endpoint descriptions can be transformed into a representation suitable for computational analysis. The results show that raw endpoint strings were not suitable prediction targets on their own, because similar measurements were often expressed using different wording and similar wording did not always imply the same clinical construct. The reviewed hierarchical endpoint representation addressed this by organizing secondary endpoints into broad and fine-grained clusters. The final hierarchy provided a more interpretable and model-compatible endpoint space, although it should be understood as a reviewed practical representation rather than a definitive clinical taxonomy.

The second research question asked whether protocol characteristics, primary endpoints, and partially observed secondary-endpoint information can be used to recommend relevant endpoint clusters and candidate secondary endpoints. The experiments show that this is possible to a limited but meaningful extent. From-scratch prediction provided a useful baseline, but the strongest practical behavior came from partial-information and pairwise formulations, where observed endpoint context helped guide the recommendation of missing endpoint clusters and candidate endpoints. Stage 1 was most useful as a candidate-generation step, while Stage 2 showed that concrete endpoint ranking is feasible when relevant candidates are available. The full-pipeline evaluation on unseen Tier C protocols further indicated that the system could often retrieve clinically related endpoint concepts, even when exact recovery of the hidden endpoint string was not realistic.

The third research question concerned the effect of different representation choices, clustering strategies, and ML methods. The main conclusion is that representation design mattered at least as much as model choice. Flat clustering and direct missing-cluster prediction were too restrictive or unstable, while hierarchical structuring and

pairwise candidate scoring better matched the endpoint-recommendation problem. RF and XGBoost both showed useful behavior in different settings, but XGBoost was selected for the final pairwise stages because it provided the most useful balance between ranking performance, generalization, and practical recommendation behavior. The results also showed that model selection could not rely only on aggregate validation metrics; expert review was necessary to identify when predictions were clinically plausible but too generic or insufficiently protocol-specific.

The fourth research question addressed the role of standardized clinical terminologies. NCIt, CDISC, and LOINC improved semantic consistency and provided interpretable structured signals for both protocol and endpoint representations. However, terminology matching was not sufficient by itself. The codes were most useful when combined with the reviewed endpoint hierarchy, observed endpoint context, and pairwise ranking models. They should therefore be interpreted as supporting semantic features rather than as perfect ground-truth mappings.

Overall, the study demonstrates that endpoint recommendation for heart-failure trials is not only a model-training task, but a layered clinical-informatics problem. Meaningful recommendation required dataset construction, endpoint hierarchy design, terminology standardization, partial-information modeling, candidate-pool construction, endpoint ranking, and expert interpretation. The final system should therefore be viewed as a proof of concept for endpoint-focused decision support rather than as a production-ready tool.

The main limitations are that the study is retrospective, based on registry-derived data, restricted empirically to heart failure, and evaluated mostly offline. The pipeline is not hardcoded to heart failure, but applying it to another therapeutic area would require new data processing, hierarchy review, model training, and validation. Future work should therefore focus on broader therapeutic-area testing, stronger expert-labelled evaluation sets, improved terminology resources, duplicate-aware or LLM-assisted reranking, and prospective validation with protocol designers. Despite these limitations, the study shows that historical clinical trial data can be reused more systematically to support endpoint-selection discussions and provides a methodological foundation for further work on AI-assisted clinical trial protocol design.

Bibliography

[1] National Center for Advancing Translational Sciences, *Endpoint*, NCATS Toolkit Glossary, Accessed 2026-05-19, 2026. [Online]. Available: https://toolkit.ncats.nih.gov/glossary/endpoint/.

[2] U.S. Food and Drug Administration, *Multiple endpoints in clinical trials: Guidance for industry*, FDA Guidance Snapshot, Accessed 2026-05-19, 2022. [Online]. Available: https://www.fda.gov/media/162427/download.

[3] T. Das, M. Beigi, J. Aptekar, and J. Sun, "AMEND++: Benchmarking Eligibility Criteria Amendments in Clinical Trials," *arXiv preprint arXiv:2601.06300*, 2026. DOI: 10.48550/arXiv.2601.06300. arXiv: 2601.06300 [cs.CL]. [Online]. Available: https://arxiv.org/abs/2601.06300.

[4] M. Pato, M. Barros, and F. M. Couto, "Survey on Recommender Systems for Biomedical Items in Life and Health Sciences," *ACM Computing Surveys*, vol. 56, no. 6, 2024. DOI: 10.1145/3639047. [Online]. Available: https://dl.acm.org/doi/10.1145/3639047.

[5] K. A. Getz et al., "The impact of protocol amendments on clinical trial performance and cost," *Therapeutic Innovation & Regulatory Science*, vol. 50, 2016. DOI: 10.1177/2168479016632271. [Online]. Available: https://doi.org/10.1177/2168479016632271.

[6] E. Botto, Z. Smith, and K. Getz, "New benchmarks on protocol amendment experience in oncology clinical trials," *Therapeutic Innovation & Regulatory Science*, vol. 58, 2024. DOI: 10.1007/s43441-024-00629-2. [Online]. Available: https://doi.org/10.1007/s43441-024-00629-2.

[7] K. Getz, Z. Smith, E. Botto, E. Murphy, and A. Dauchy, "New benchmarks on protocol amendment practices, trends and their impact on clinical trial performance," *Therapeutic Innovation & Regulatory Science*, vol. 58, 2024. DOI: 10.1007/s43441-024-00622-9. [Online]. Available: https://doi.org/10.1007/s43441-024-00622-9.

[8] S. Joshi, "Common clinical trial amendments, why they are submitted and how they can be avoided: A mixed methods study on nhs uk sponsored research (amendments assemble)," *Trials*, vol. 24, 2023. DOI: 10.1186/s13063-022-06989-0. [Online]. Available: https://doi.org/10.1186/s13063-022-06989-0.

[9] National Cancer Institute, *NCI Thesaurus (NCIt)*, https://www.cancer.gov/research/resources/resource/197, Accessed 2026-05-20, 2026.

[10] Clinical Data Interchange Standards Consortium, *CDISC Controlled Terminology*, https://www.cdisc.org/standards/terminology/controlled-terminology, Accessed 2026-05-20, 2026.

[11] Regenstrief Institute, *LOINC: Logical Observation Identifiers Names and Codes*, https://loinc.org/, Accessed 2026-05-20, 2026.

[12] P. Majumdar, *Finding semantically similar clinical trials using sentence embeddings and a transformer model*, Zenodo-hosted paper, 2025. DOI: 10.5281/zenodo.16195845. [Online]. Available: https://doi.org/10.5281/zenodo.16195845.

[13] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence embeddings using siamese BERT-networks," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, Association for Computational Linguistics, 2019. [Online]. Available: https://aclanthology.org/D19-1410/.

[14] scikit-learn developers, *Clustering: Hierarchical clustering*, scikit-learn User Guide, Accessed: 2026-05-29, 2026. [Online]. Available: https://scikit-learn.org/stable/modules/clustering.html.

[15] C. N. J. Silla and A. A. Freitas, "A survey of hierarchical classification across different application domains," *Data Mining and Knowledge Discovery*, vol. 22, 2011. DOI: 10.1007/s10618-010-0175-9. [Online]. Available: https://doi.org/10.1007/s10618-010-0175-9.

[16] Anthropic, *Prompting Best Practices*, Claude API Documentation, Accessed: 2026-05-25, 2026. [Online]. Available: https://platform.claude.com/docs/en/build-with-claude/prompt-engineering/claude-prompting-best-practices.

[17] P. Sahoo, A. K. Singh, S. Saha, V. Jain, S. Mondal, and A. Chadha, "A Systematic Survey of Prompt Engineering in Large Language Models: Techniques and Applications," *arXiv preprint arXiv:2402.07927*, 2024. arXiv: 2402.07927 [cs.CL]. [Online]. Available: https://arxiv.org/abs/2402.07927.

[18] L. Breiman, "Random forests," *Machine Learning*, vol. 45, 2001. DOI: 10.1023/A:1010933404324. [Online]. Available: https://doi.org/10.1023/A:1010933404324.

[19] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Association for Computing Machinery, 2016. DOI: 10.1145/2939672.2939785. [Online]. Available: https://doi.org/10.1145/2939672.2939785.

[20] J. Read, B. Pfahringer, G. Holmes, and E. Frank, "Classifier chains for multi-label classification," *Machine Learning*, vol. 85, 2011. DOI: 10.1007/s10994-011-5256-5. [Online]. Available: https://doi.org/10.1007/s10994-011-5256-5.

[21] T. Fawcett, "An introduction to ROC analysis," *Pattern Recognition Letters*, vol. 27, 2006. DOI: 10.1016/j.patrec.2005.10.010. [Online]. Available: https://doi.org/10.1016/j.patrec.2005.10.010.

[22] T. Saito and M. Rehmsmeier, "The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets,"

PLOS ONE, 2015. DOI: 10.1371/journal.pone.0118432. [Online]. Available: https://doi.org/10.1371/journal.pone.0118432.

[23] G. W. Brier, "Verification of forecasts expressed in terms of probability," *Monthly Weather Review*, vol. 78, no. 1, 1950. DOI: 10.1175/1520-0493(1950)078<0001:VOFEIT>2.0.CO;2. [Online]. Available: https://doi.org/10.1175/1520-0493(1950)078%3C0001:VOFEIT%3E2.0.CO;2.

[24] C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to Information Retrieval*. Cambridge University Press, 2008. DOI: 10.1017/CBO9780511809071. [Online]. Available: https://doi.org/10.1017/CBO9780511809071.

[25] G. Shani and A. Gunawardana, "Evaluating recommendation systems," in *Recommender Systems Handbook*, F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor, Eds., Springer, 2011. DOI: 10.1007/978-0-387-85820-3_8. [Online]. Available: https://doi.org/10.1007/978-0-387-85820-3_8.

[26] Google Cloud, *Agent Development Kit*, Google Cloud Documentation, Accessed: 2026-05-27, 2026. [Online]. Available: https://docs.cloud.google.com/gemini-enterprise-agent-platform/build/adk.

[27] National Library of Medicine (US), *Clinicaltrials.gov [internet]*, Bethesda (MD): National Library of Medicine (US). Available from: https://clinicaltrials.gov/, 2000. [Online]. Available: https://clinicaltrials.gov/.

[28] sentence-transformers, *all-MiniLM-L6-v2*, Hugging Face model card, Accessed: 2026-05-27, 2025. [Online]. Available: https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2.

[29] Qwen, *Qwen3-Embedding-8B*, Hugging Face model card, Accessed: 2026-05-27, 2025. [Online]. Available: https://huggingface.co/Qwen/Qwen3-Embedding-8B.

[30] Anthropic, *Claude 3.7 Sonnet and Claude Code*, Anthropic News, Accessed: 2026-05-25, Feb. 2025. [Online]. Available: https://www.anthropic.com/news/claude-3-7-sonnet.

[31] W. McKinney, "Data structures for statistical computing in python," in *Proceedings of the 9th Python in Science Conference*, 2010. DOI: 10.25080/Majora-92bf1922-00a.

[32] C. R. Harris, K. J. Millman, S. J. van der Walt, et al., "Array programming with NumPy," *Nature*, vol. 585, 2020. DOI: 10.1038/s41586-020-2649-2.

[33] P. Virtanen, R. Gommers, T. E. Oliphant, et al., "SciPy 1.0: Fundamental algorithms for scientific computing in python," *Nature Methods*, vol. 17, no. 3, 2020. DOI: 10.1038/s41592-019-0686-2.

[34] A. Paszke, S. Gross, F. Massa, et al., "PyTorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems*, vol. 32, Curran Associates, Inc., 2019.

[35] F. Pedregosa, G. Varoquaux, A. Gramfort, et al., "Scikit-learn: Machine learning in python," *Journal of Machine Learning Research*, vol. 12, 2011.

[36] RapidFuzz developers, *RapidFuzz: Rapid fuzzy string matching in python*, Software documentation, Accessed: 2026-06-05, 2026. [Online]. Available: https://rapidfuzz.github.io/RapidFuzz/.

[37] joblib developers, *joblib: Running python functions as pipeline jobs*, Software documentation, Accessed: 2026-06-05, 2026. [Online]. Available: `https://joblib.readthedocs.io/`.

[38] J. D. Hunter, "Matplotlib: A 2d graphics environment," *Computing in Science Engineering*, vol. 9, no. 3, 2007. DOI: `10.1109/MCSE.2007.55`.

[39] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in *Advances in Neural Information Processing Systems*, vol. 30, Curran Associates, Inc., 2017.

[40] L. McInnes, J. Healy, N. Saul, and L. Großberger, "UMAP: Uniform manifold approximation and projection," *Journal of Open Source Software*, vol. 3, no. 29, 2018. DOI: `10.21105/joss.00861`. [Online]. Available: `https://doi.org/10.21105/joss.00861`.

A

Abbreviations

Computer Science and Machine Learning

ADK	Agent Development Kit
AI	Artificial Intelligence
CC	Classifier Chains
CSV	Comma-Separated Values
FN	False Negative
FP	False Positive
HDBSCAN	Hierarchical Density-Based Spatial Clustering of Applications with Noise
JSON	JavaScript Object Notation
LLM	Large Language Model
LR	Logistic Regression
ML	Machine Learning
MRR	Mean Reciprocal Rank
NLP	Natural Language Processing
PR-AUC	Area Under the Precision-Recall Curve
RF	Random Forest
ROC-AUC	Area Under the Receiver Operating Characteristic Curve
UMAP	Uniform Manifold Approximation and Projection
XGBoost	eXtreme Gradient Boosting

Clinical and Medical Terms

ARDS	Acute Respiratory Distress Syndrome
CV	Cardiovascular
HF	Heart Failure
HFmrEF	Heart Failure with Mildly Reduced Ejection Fraction
HFpEF	Heart Failure with Preserved Ejection Fraction
HFrEF	Heart Failure with Reduced Ejection Fraction
NYHA	New York Heart Association
RVEDA/LVEDA	Right/Left Ventricular End-Diastolic Area
TAPSE	Tricuspid Annular Plane Systolic Excursion
TEE	Transesophageal Echocardiography

Clinical Trials, Standards, and Ontologies

CDISC	Clinical Data Interchange Standards Consortium
LOINC	Logical Observation Identifiers Names and Codes
MeSH	Medical Subject Headings
NCIt	National Cancer Institute Thesaurus
USDM	Unified Study Definitions Model

B

Data and Representation Examples

This appendix provides concrete examples of the final data artifacts used in the thesis pipeline. The purpose is to make the transformations described in Chapter 4 more transparent by showing how raw ClinicalTrials.gov records were reduced, converted into tabular protocol-level and endpoint-level representations, organized into the reviewed endpoint hierarchy, standardized against biomedical terminologies, and finally transformed into model-ready target indicators.

All examples below are taken from the final project workflow and use real records or shortened excerpts of real records. Long text fields, long code lists, and repeated nested structures are shortened using ellipses where this improves readability without changing the structure being illustrated.

B.1 Overview of final data artifacts

Table B.1 summarizes the main final data artifacts used by the pipeline. The raw and reduced JSON files preserve the original registry structure and the reduced protocol structure, while the CSV and hierarchy files provide the tabular, standardized, and modeling-oriented views used in later stages.

Artifact	Granularity	Role in the pipeline
Full ClinicalTrials.gov JSON records	one file per protocol	Original registry records before reduction. These contain nested protocol, results, derived, and document sections.
Reduced protocol JSON records	one file per retained protocol	Compact protocol representation after field selection. These files preserve the clinically relevant protocol fields used for downstream processing.
standardized_protocols_validated.csv	one row per protocol	Protocol-level standardized table containing structured design variables, eligibility indicators, intervention summaries, and protocol-level NCIt, CDISC, and LOINC code sets.
standardized_endpoints.csv	one row per endpoint	Endpoint-level standardization table for primary and secondary endpoints, including endpoint text fields and terminology-code outputs.
standardized_endpoints_SECONDARY.csv	one row per secondary endpoint	Secondary-endpoint-only standardization table used for endpoint-specific code inspection and downstream aggregation.
UNIQUE_IDS_FIXED_tier0_corrected.json	reviewed hierarchy	Final reviewed three-level endpoint hierarchy with unique Tier 0, Tier 1, and Tier 2 identifiers. This is the authoritative hierarchy used for final target labels.
raw_dataset_secondary_tier0.csv	one row per secondary endpoint	Final secondary-endpoint modeling view containing protocol context, endpoint text, structured protocol variables, and binary Tier 0, Tier 1, and Tier 2 target indicators.

Table B.1: Main final data artifacts used in the thesis pipeline.

B.2 Example of a raw ClinicalTrials.gov JSON record

Listing B.1 shows a shortened excerpt from the full ClinicalTrials.gov JSON record for NCT02625922. This example illustrates the nested source format before reduction. The full record contains additional locations, results tables, adverse-event entries, and metadata that are omitted here for space.

Listing B.1: Shortened excerpt of a full ClinicalTrials.gov JSON record for NCT02625922.


```
{
  "protocolSection": {
    "identificationModule": {
      "nctId": "NCT02625922",
      "orgStudyIdInfo": {
        "id": "CRLX030A2211"
      },
      "organization": {
        "fullName": "Novartis",
        "class": "INDUSTRY"
      },
      "briefTitle": "Study of the Effect of Serelaxin on High-sensitivity Cardiac Troponin I (Hs-
     cTnI) Release in Patients With Chronic Heart Failure",
      "officialTitle": "A Multicenter, Randomized, Double-blind, Crossover Placebo-controlled Phase
     II Study to Assess the Effect of Serelaxin Versus Placebo on High-sensitivity Cardiac Troponin
     I (Hs-cTnI) Release in Patients With Chronic Heart Failure After Exercise When Used in Addition
      to Standard of Care",
      "acronym": "RELAX-Cardio"
    },
    "statusModule": {
      "overallStatus": "TERMINATED",
```

```
    "whyStopped": "In view of outcome of RELAX-AHF-2 trial, the entire RLX030A project was decided
    to be terminated.",
    "primaryCompletionDateStruct": {
      "date": "2017-01-11",
      "type": "ACTUAL"
    },
    "completionDateStruct": {
      "date": "2017-01-11",
      "type": "ACTUAL"
    }
  },
  "descriptionModule": {
    "briefSummary": "This was a multicenter, randomized, double-blind, crossover, placebo-
   controlled, Phase II clinical study that evaluated the effect of serelaxin versus placebo on
   the release of hs-cTnI in patients with CHF after an exercise testing session."
  },
  "conditionsModule": {
    "conditions": [
      "Chronic Heart Failure"
    ],
    "keywords": [
      "Serelaxin",
      "Chronic heart failure",
      "Spiroergometry",
      "Troponin",
      "New York Heart Association (NYHA) functional Class II/III",
      "Left ventricular ejection fraction"
    ]
  },
  "designModule": {
    "studyType": "INTERVENTIONAL",
    "phases": [
      "PHASE2"
    ],
    "designInfo": {
      "allocation": "RANDOMIZED",
      "interventionModel": "CROSSOVER",
      "primaryPurpose": "TREATMENT",
      "maskingInfo": {
        "masking": "TRIPLE",
        "whoMasked": [
          "PARTICIPANT",
          "INVESTIGATOR",
          "OUTCOMES_ASSESSOR"
        ]
      }
    },
    "enrollmentInfo": {
      "count": 26,
      "type": "ACTUAL"
    }
  },
  "armsInterventionsModule": {
    "armGroups": [
      {
        "label": "Serelaxin followed by Placebo",
        "type": "EXPERIMENTAL",
        "interventionNames": [
          "Drug: Serelaxin",
          "Drug: Placebo"
        ]
      },
      {
        "label": "Placebo followed by Serelaxin",
        "type": "EXPERIMENTAL",
        "interventionNames": [
          "Drug: Serelaxin",
          "Drug: Placebo"
        ]
      }
```

```
    ],
    "interventions": [
      {
        "type": "DRUG",
        "name": "Serelaxin",
        "otherNames": [
          "RLX030"
        ]
      },
      {
        "type": "DRUG",
        "name": "Placebo"
      }
    ]
  },
  "outcomesModule": {
    "primaryOutcomes": [
      {
        "measure": "Geometric Mean of High Sensitivity Cardiac Troponin I (Hs-cTnI) Concentration
   After Exercise Compared to Placebo",
        "description": "This cardiac biomarker measurement was obtained to determine plasma
   concentrations following a cardiac stress test.",
        "timeFrame": "Baseline, up to 7 hours after the start of an exercise testing session on
   treatment period 1 and treatment period 2"
      }
    ],
    "secondaryOutcomes": [
      {
        "measure": "Geometric Mean of High Sensitivity Cardiac Troponin I (Hs-cTnI) Concentrations
    After Exercise Compared to Placebo at 4 and 5 Hours",
        "description": "This cardiac biomarker measurement was obtained to determine plasma
   concentrations following a cardiac stress test.",
        "timeFrame": "4 and 5 hours after exercise testing session"
      },
      {
        "measure": "Log-transformed Concentration of N-terminal Pro-B-type Natriuretic Peptide (NT
   -proBNP) Concentrations Compared to Placebo",
        "description": "This cardiac biomarker measurement was obtained to determine plasma
   concentrations following a cardiac stress test.",
        "timeFrame": "Baseline, up to 7 hours after the start of an exercise testing session on
   treatment period 1 and treatment period 2"
      },
      {
        "measure": "Log-transformed Concentration Values of Heart-type Fatty Acid-binding Protein
   (H-FABP) Concentrations Compared to Placebo",
        "description": "This cardiac biomarker measurement was obtained to determine plasma
   concentrations following a cardiac stress test.",
        "timeFrame": "Baseline, up to 7 hours after the start of an exercise testing session on
   treatment period 1 and treatment period 2"
      }
    ]
  },
  "eligibilityModule": {
    "eligibilityCriteria": "Key Inclusion Criteria: Male or female >= 18 years of age; diagnosis
   of stable CHF; NYHA functional Class II/III; left ventricular ejection fraction < 45%; NT-
   proBNP above threshold; ability to exercise. Key Exclusion Criteria: dyspnea primarily due to
   non-cardiac causes; contraindication for exercise testing and spirometry; recent change in CHF
   treatment.",
    "healthyVolunteers": false,
    "sex": "ALL",
    "minimumAge": "18 Years",
    "stdAges": [
      "ADULT",
      "OLDER_ADULT"
    ]
  }
},
"resultsSection": {
  "participantFlowModule": {
    "recruitmentDetails": "There were total 11 centers from 3 countries.",
```

```
    "groups": [
      {
        "id": "FG000",
        "title": "Serelaxin-Placebo"
      },
      {
        "id": "FG001",
        "title": "Placebo-Serelaxin"
      }
    ]
  },
  "outcomeMeasuresModule": {
    "outcomeMeasures": [
      {
        "type": "PRIMARY",
        "title": "Geometric Mean of High Sensitivity Cardiac Troponin I (Hs-cTnI) Concentration
 After Exercise Compared to Placebo",
        "reportingStatus": "POSTED",
        "denoms": [
          {
            "units": "Participants",
            "counts": [
              {
                "groupId": "OG000",
                "value": "0"
              }
            ]
          }
        ]
      },
      {
        "type": "SECONDARY",
        "title": "Log-transformed Concentration of N-terminal Pro-B-type Natriuretic Peptide (NT-
 proBNP) Concentrations Compared to Placebo",
        "paramType": "MEAN",
        "dispersionType": "Standard Deviation",
        "unitOfMeasure": "pg/mL",
        "classes": [
          {
            "title": "Baseline",
            "categories": [
              {
                "measurements": [
                  {
                    "groupId": "OG000",
                    "value": "13.5026",
                    "spread": "0.64893"
                  }
                ]
              }
            ]
          }
        ]
      }
    ]
  },
  "adverseEventsModule": {
    "frequencyThreshold": "4",
    "eventGroups": [
      {
        "id": "EG000",
        "title": "Serelaxin",
        "deathsNumAffected": 0,
        "seriousNumAffected": 0,
        "otherNumAffected": 7
      }
    ]
  }
},
"derivedSection": {
```

```
    "interventionBrowseModule": {
      "meshes": [
        {
          "id": "C577649",
          "term": "serelaxin protein, human"
        }
      ]
    }
  },
  "hasResults": true
}
```


B.3 Example of a reduced protocol JSON record

Listing B.2 shows the reduced version of the same protocol. Compared with the full JSON in Listing B.1, the reduced representation preserves study identification, status, descriptions, design fields, arms, interventions, outcomes, eligibility fields, browse terms, and selected result-related summaries, while removing fields that were not used by the downstream pipeline.

Listing B.2: Shortened excerpt of the reduced protocol JSON record for NCT02625922.


```
{
  "hasResults": true,
  "protocolSection": {
    "identificationModule": {
      "nctId": "NCT02625922",
      "briefTitle": "Study of the Effect of Serelaxin on High-sensitivity Cardiac Troponin I (Hs-
     cTnI) Release in Patients With Chronic Heart Failure",
      "officialTitle": "A Multicenter, Randomized, Double-blind, Crossover Placebo-controlled Phase
     II Study to Assess the Effect of Serelaxin Versus Placebo on High-sensitivity Cardiac Troponin
     I (Hs-cTnI) Release in Patients With Chronic Heart Failure After Exercise When Used in Addition
      to Standard of Care",
      "acronym": "RELAX-Cardio",
      "organization": {
        "fullName": "Novartis",
        "class": "INDUSTRY"
      }
    },
    "sponsorCollaboratorsModule": {
      "leadSponsor": {
        "name": "Novartis Pharmaceuticals",
        "class": "INDUSTRY"
      },
      "collaborators": null
    },
    "statusModule": {
      "overallStatus": "TERMINATED",
      "whyStopped": "In view of outcome of RELAX-AHF-2 trial, the entire RLX030A project was decided
     to be terminated.",
      "startDateStruct": {
        "date": "2016-02-05"
      },
      "primaryCompletionDateStruct": {
        "date": "2017-01-11",
        "type": "ACTUAL"
      },
      "completionDateStruct": {
        "date": "2017-01-11",
        "type": "ACTUAL"
      }
    },
    "descriptionModule": {
```

```
    "briefSummary": "This was a multicenter, randomized, double-blind, crossover, placebo-
   controlled, Phase II clinical study that evaluated the effect of serelaxin versus placebo on
   the release of hs-cTnI in patients with CHF after an exercise testing session.",
    "detailedDescription": null
  },
  "conditionsModule": {
    "conditions": [
      "Chronic Heart Failure"
    ],
    "keywords": [
      "Serelaxin",
      "Chronic heart failure",
      "Spiroergometry",
      "Troponin",
      "New York Heart Association (NYHA) functional Class II/III",
      "Left ventricular ejection fraction"
    ]
  },
  "designModule": {
    "studyType": "INTERVENTIONAL",
    "phases": [
      "PHASE2"
    ],
    "targetDuration": null,
    "designInfo": {
      "allocation": "RANDOMIZED",
      "interventionModel": "CROSSOVER",
      "primaryPurpose": "TREATMENT",
      "maskingInfo": {
        "masking": "TRIPLE",
        "whoMasked": [
          "PARTICIPANT",
          "INVESTIGATOR",
          "OUTCOMES_ASSESSOR"
        ]
      }
    },
    "enrollmentInfo": {
      "count": 26,
      "type": "ACTUAL"
    }
  },
  "armsInterventionsModule": {
    "armGroups": [
      {
        "label": "Serelaxin followed by Placebo",
        "type": "EXPERIMENTAL",
        "description": "On Day 1 of treatment period 1, Serelaxin will be administered as a
   continuous i.v. infusion. In treatment period 2, matching placebo will be administered after
   washout.",
        "interventionNames": [
          "Drug: Serelaxin",
          "Drug: Placebo"
        ]
      },
      {
        "label": "Placebo followed by Serelaxin",
        "type": "EXPERIMENTAL",
        "description": "On Day 1 of treatment period 1, matching placebo will be administered. In
   treatment period 2, Serelaxin will be administered after washout.",
        "interventionNames": [
          "Drug: Serelaxin",
          "Drug: Placebo"
        ]
      }
    ],
    "interventions": [
      {
        "type": "DRUG",
        "name": "Serelaxin",
```

```
        "description": "Serelaxin will be administered as a continuous i.v. infusion according to
    a weight-range adjusted dosing regimen.",
        "otherNames": [
          "RLX030"
        ]
      },
      {
        "type": "DRUG",
        "name": "Placebo",
        "description": "Matching placebo i.v infusion"
      }
    ]
  },
  "outcomesModule": {
    "primaryOutcomes": [
      {
        "measure": "Geometric Mean of High Sensitivity Cardiac Troponin I (Hs-cTnI) Concentration
    After Exercise Compared to Placebo",
        "description": "This cardiac biomarker measurement was obtained to determine plasma
    concentrations following a cardiac stress test.",
        "timeFrame": "Baseline, up to 7 hours after the start of an exercise testing session on
    treatment period 1 and treatment period 2"
      }
    ],
    "secondaryOutcomes": [
      {
        "measure": "Geometric Mean of High Sensitivity Cardiac Troponin I (Hs-cTnI) Concentrations
     After Exercise Compared to Placebo at 4 and 5 Hours",
        "description": "This cardiac biomarker measurement was obtained to determine plasma
    concentrations following a cardiac stress test.",
        "timeFrame": "4 and 5 hours after exercise testing session"
      },
      {
        "measure": "Log-transformed Concentration of N-terminal Pro-B-type Natriuretic Peptide (NT
    -proBNP) Concentrations Compared to Placebo",
        "description": "This cardiac biomarker measurement was obtained to determine plasma
    concentrations following a cardiac stress test.",
        "timeFrame": "Baseline, up to 7 hours after the start of an exercise testing session on
    treatment period 1 and treatment period 2"
      },
      {
        "measure": "Log-transformed Concentration Values of Heart-type Fatty Acid-binding Protein
    (H-FABP) Concentrations Compared to Placebo",
        "description": "This cardiac biomarker measurement was obtained to determine plasma
    concentrations following a cardiac stress test.",
        "timeFrame": "Baseline, up to 7 hours after the start of an exercise testing session on
    treatment period 1 and treatment period 2"
      }
    ],
    "otherOutcomes": null
  },
  "eligibilityModule": {
    "eligibilityCriteria": "Key Inclusion Criteria: Male or female >= 18 years of age; stable CHF;
     NYHA functional Class II/III; left ventricular ejection fraction < 45%; NT-proBNP above
    threshold; ability to exercise. Key Exclusion Criteria: non-cardiac dyspnea; contraindication
    for exercise testing and spirometry; recent change in CHF treatment.",
    "healthyVolunteers": false,
    "sex": "ALL",
    "minimumAge": "18 Years",
    "stdAges": [
      "ADULT",
      "OLDER_ADULT"
    ]
  },
  "referencesModule": {
    "references": null
  }
},
"derivedSection": {
  "conditionBrowseModule": null,
```

```
    "interventionBrowseModule": {
      "meshes": [
        {
          "id": "C577649",
          "term": "serelaxin protein, human"
        }
      ]
    }
  },
  "documentSection": {
    "largeDocumentModule": null
  },
  "resultsStatMethods": null
}
```


B.4 Example of a protocol-level CSV row

The final protocol-level standardized table is represented by `standardized_protocols_validated.cs`
Each row corresponds to one protocol and combines structured study-design variables with terminology-derived code sets. Table B.2 shows selected fields for `NCT02625922`. Long code lists are shortened for readability.

Column	**Value for `NCT02625922`**
nctId	NCT02625922
phase	PHASE2
phase_numeric	2
study_type	INTERVENTIONAL
n_arms	2
n_interventions	2
interv_has_DRUG	1
elig_sex_present	1
sex_is_ALL	1
minimumAge_years	18
stdAges_has_ADULT	1
stdAges_has_OLDER_ADULT	1
n_ncit_codes	16
n_cdisc_codes	2
n_loinc_codes	9
best_system_final	ncit
ncit_codes	C101839; C110935; C111327; C116517; . . . ; C99524
cdisc_codes	C15220; C15228
loinc_codes	102097-3; 10230-1; 103693-8; 18684-1; . . . ; 93124-6

Table B.2: Representative excerpt of a protocol-level standardized CSV row from `standardized_protocols_validated.csv`.

The full row also contains additional binary indicators for arm-group types, intervention types, eligibility-field availability, age-group availability, phase flags, protocol text, intervention-name lists, intervention-type lists, and code-count diagnostics.

B.5 Example of endpoint-level CSV rows

Endpoint-level representations are used in two closely related ways. Primary endpoints are standardized and later aggregated into input features, whereas secondary endpoints are standardized and assigned to the final reviewed hierarchy to form

prediction targets. Table B.3 shows one primary endpoint and the three secondary endpoints from NCT02625922. For secondary endpoints, the final Tier 0–Tier 2 path shown below is taken from the final reviewed hierarchy.

Role	Endpoint measure	Final hierarchy role	Example standardized codes
Primary input	Geometric Mean of High Sensitivity Cardiac Troponin I (Hs-cTnI) Concentration After Exercise Compared to Placebo	Used as primary-endpoint code features, not as a secondary target.	NCIt: C100073; C104220; ... CDISC: C199678 LOINC: 101911-6; 101913-2; ...
Secondary target	Geometric Mean of High Sensitivity Cardiac Troponin I (Hs-cTnI) Concentrations After Exercise Compared to Placebo at 4 and 5 Hours	Tier 0: Cardiac Structure and Function Tier 1: Cardiac Function, Perfusion and Energetics Tier 2: Cardiac Biomarkers Including Troponin	NCIt: C100073; C104220; ... CDISC: C199678; C49648 LOINC: 101911-6; 77516-3; ...
Secondary target	Log-transformed Concentration of N-terminal Pro-B-type Natriuretic Peptide (NT-proBNP) Concentrations Compared to Placebo	Tier 0: Natriuretic Peptides and Cardiac Biomarkers Tier 1: Terminal PRO Natriuretic Tier 2: NT-proBNP Measurements	NCIt: C100068; C82032; C96610; ... CDISC: C199678 LOINC: 106735-4; 30934-4; ...
Secondary target	Log-transformed Concentration Values of Heart-type Fatty Acid-binding Protein (H-FABP) Concentrations Compared to Placebo	Tier 0: Cardiac Structure and Function Tier 1: Cardiac Function, Perfusion and Energetics Tier 2: Cardiac Biomarkers Including Troponin	NCIt: C104927; C106521; C38502; ... CDISC: – LOINC: 100795-4; 106764-4; ...

Table B.3: Representative endpoint-level rows for NCT02625922.

B.6 Example of final binary hierarchy indicators

The final secondary-endpoint modeling table, raw_dataset_secondary_tier0.csv, contains one row per secondary endpoint and includes binary columns for every retained Tier 0, Tier 1, and Tier 2 identifier. These indicator columns are later aggregated to the protocol level by logical maximum when constructing multi-label targets.

Listing B.3 shows a compact view of the active hierarchy indicators for the three secondary endpoints of NCT02625922.

Listing B.3: Compact target-indicator view derived from raw_dataset_secondary_tier0.csv.

```
nctId,index,level,measure,active_tier0,active_tier1,active_tier2
NCT02625922,0,secondary,"Geometric Mean of High Sensitivity Cardiac Troponin I (Hs-cTnI)
    Concentrations After Exercise Compared to Placebo at 4 and 5 Hours",tier0_id_10,tier1_id_16,
    tier2_id_511
NCT02625922,1,secondary,"Log-transformed Concentration of N-terminal Pro-B-type Natriuretic Peptide
    (NT-proBNP) Concentrations Compared to Placebo",tier0_id_15,tier1_id_113,tier2_id_481
NCT02625922,2,secondary,"Log-transformed Concentration Values of Heart-type Fatty Acid-binding
    Protein (H-FABP) Concentrations Compared to Placebo",tier0_id_10,tier1_id_16,tier2_id_511
```

After protocol-level aggregation, this protocol therefore receives the following secondary-endpoint target labels:

```
nctId,tier0_targets,tier1_targets,tier2_targets
NCT02625922,"tier0_id_10; tier0_id_15","tier1_id_16; tier1_id_113","tier2_id_511; tier2_id_481"
```

B.7 Example of the reviewed endpoint hierarchy

The final reviewed hierarchy is stored in `UNIQUE_IDS_FIXED_tier0_corrected.json`. It contains 16 Tier 0 groups, 61 Tier 1 groups, 314 Tier 2 clusters, and 3,700 secondary endpoint records. Table B.4 gives a compact overview of the final Tier 0 layer.

ID	Tier 0 name	Tier 1	Tier 2	Endpoints
1	Medication and Device Interventions	7	10	43
2	Safety and Adverse Events	7	20	110
3	Exercise Capacity and Physical Function	5	13	258
4	Imaging and Tissue Characterization	5	11	31
5	Pharmacokinetics and Drug Concentrations	5	10	45
6	Blood and Plasma Biomarkers	4	30	181
7	Body Composition and Metabolism	4	15	145
8	Disease Response and Progression	4	20	106
9	Miscellaneous Clinical Assessments	4	12	42
10	Cardiac Structure and Function	3	88	1462
11	Patient-Reported Outcomes and Quality of Life	3	23	375
12	Pulmonary Function and Vascular Resistance	3	12	115
13	Blood Pressure and Hemodynamics	2	7	87
14	Mortality and Survival Outcomes	2	18	360
15	Natriuretic Peptides and Cardiac Biomarkers	2	5	176
16	Renal and Urinary Function	1	20	164

Table B.4: Overview of the final Tier 0 layer in the reviewed endpoint hierarchy.

Listing B.4 shows a shortened JSON-like excerpt from the same hierarchy. The excerpt uses endpoint examples from `NCT02625922` and shows how endpoint records are nested under Tier 0, Tier 1, and Tier 2 labels.

Listing B.4: Shortened excerpt of the final reviewed endpoint hierarchy.

```
{
  "tier0_clusters": {
    "Cardiac Structure and Function": {
      "tier0_id": 10,
      "tier1_clusters": {
        "Cardiac Function, Perfusion and Energetics": {
          "tier1_id": 16,
          "subclusters": {
            "Cardiac Biomarkers Including Troponin": {
              "tier2_id": 511,
              "endpoints": [
                {
                  "nct_id": "NCT02625922",
                  "outcome_type": "secondary",
                  "measure": "Geometric Mean of High Sensitivity Cardiac Troponin I (Hs-cTnI)
    Concentrations After Exercise Compared to Placebo at 4 and 5 Hours",
                  "timeframe": "4 and 5 hours after exercise testing session",
                  "ncit_codes": "C100073;C104220;C104226;...",
                  "cdisc_codes": "C199678;C49648",
                  "loinc_codes": "101911-6;101913-2;77516-3;..."
                },
                {
                  "nct_id": "NCT02625922",
                  "outcome_type": "secondary",
```

```
                "measure": "Log-transformed Concentration Values of Heart-type Fatty Acid-binding
    Protein (H-FABP) Concentrations Compared to Placebo",
                "timeframe": "Baseline, up to 7 hours after the start of an exercise testing
    session on treatment period 1 and treatment period 2",
                "ncit_codes": "C104927;C106521;C38502;...",
                "cdisc_codes": "",
                "loinc_codes": "100795-4;106764-4;107216-4;..."
              }
            ]
          }
        }
      }
    }
   },
   "Natriuretic Peptides and Cardiac Biomarkers": {
     "tier0_id": 15,
     "tier1_clusters": {
       "Terminal PRO Natriuretic": {
         "tier1_id": 113,
         "subclusters": {
           "NT-proBNP Measurements": {
             "tier2_id": 481,
             "endpoints": [
               {
                 "nct_id": "NCT02625922",
                 "outcome_type": "secondary",
                 "measure": "Log-transformed Concentration of N-terminal Pro-B-type Natriuretic
    Peptide (NT-proBNP) Concentrations Compared to Placebo",
                 "timeframe": "Baseline, up to 7 hours after the start of an exercise testing
    session on treatment period 1 and treatment period 2",
                 "ncit_codes": "C100068;C82032;C96610;...",
                 "cdisc_codes": "C199678",
                 "loinc_codes": "106735-4;30934-4;47255-5;..."
               }
             ]
           }
         }
       }
     }
   }
}
```


B.8 Example of terminology standardization outputs

Terminology standardization maps protocol and endpoint text to NCIt, CDISC, and LOINC code sets. Table B.5 shows compact examples from the same protocol. These mappings should be interpreted as structured semantic signals rather than as manually verified clinical ground truth.

Record type	Raw text	Standardized terminology examples
Protocol	Serelaxin study in chronic heart failure with NYHA II/III, reduced LVEF, exercise testing, and troponin/NT-proBNP context.	**NCIt:** C101839; C110935; C111327; C116517; ... **CDISC:** C15220; C15228 **LOINC:** 102097-3; 10230-1; 103693-8; ...
Primary endpoint	Geometric Mean of High Sensitivity Cardiac Troponin I Concentration After Exercise Compared to Placebo.	**NCIt:** C100073; C104220; C104226; C94906; ... **CDISC:** C199678 **LOINC:** 101911-6; 101913-2; 76645-1; ...
Secondary endpoint	Log-transformed Concentration of N-terminal Pro-B-type Natriuretic Peptide (NT-proBNP) Concentrations Compared to Placebo.	**NCIt:** C100068; C100279; C82032; C96610; ... **CDISC:** C199678 **LOINC:** 106735-4; 30934-4; 47255-5; ...
Secondary endpoint	Log-transformed Concentration Values of Heart-type Fatty Acid-binding Protein (H-FABP) Concentrations Compared to Placebo.	**NCIt:** C104927; C106521; C38502; C41109; ... **CDISC:** – **LOINC:** 100795-4; 106764-4; 107216-4; ...

Table B.5: Examples of terminology standardization outputs.

The protocol-level standardizer uses protocol context such as titles, summaries, eligibility criteria, intervention text, conditions, keywords, and browse terms. The endpoint-level standardizer uses endpoint-specific fields such as measure, concept, normalized concept, description, and timeframe. This separation is important because protocol-level codes become input features, while secondary-endpoint hierarchy labels become prediction targets.

B.9 Example of partial endpoint-information construction

The leave-X-out evaluation described in Chapter 4 creates derived samples by treating some secondary endpoints as observed and others as withheld. The protocol `NCT02625922` has three secondary endpoints in the final secondary-endpoint hierarchy:

- **Endpoint 0:** hs-cTnI concentration after exercise, assigned to `tier2_id_511`.
- **Endpoint 1:** NT-proBNP concentration, assigned to `tier2_id_481`.
- **Endpoint 2:** H-FABP concentration, assigned to `tier2_id_511`.

A leave-one-out sample that withholds Endpoint 1 and treats Endpoints 0 and 2 as observed can be represented compactly as follows:

Listing B.5: Example leave-one-out construction for `NCT02625922`.

```
protocol: NCT02625922

observed secondary endpoints:
  - index 0: hs-cTnI concentration after exercise
    active target path: tier0_id_10 -> tier1_id_16 -> tier2_id_511
  - index 2: H-FABP concentration after exercise
    active target path: tier0_id_10 -> tier1_id_16 -> tier2_id_511

withheld secondary endpoint:
  - index 1: NT-proBNP concentration
    withheld target path: tier0_id_15 -> tier1_id_113 -> tier2_id_481

observed secondary target set:
  tier0: {tier0_id_10}
  tier1: {tier1_id_16}
  tier2: {tier2_id_511}

full protocol target set:
  tier0: {tier0_id_10, tier0_id_15}
  tier1: {tier1_id_16, tier1_id_113}
  tier2: {tier2_id_511, tier2_id_481}

missing-target mode:
  positive missing candidate: tier2_id_481

full-target mode:
  positive protocol-relevant candidates: tier2_id_511 and tier2_id_481
```

This example illustrates the difference between the strict missing-cluster formulation and the broader full-target relevance-ranking formulation. In missing-target mode, `tier2_id_481` is positive because it corresponds to the withheld NT-proBNP endpoint. In full-target mode, both `tier2_id_511` and `tier2_id_481` are positive because both are part of the full true secondary-endpoint profile of the protocol.

C

Standardized Codes Translation

C.1 Example of codes from Stage 2 feature importance.

Several of the high-ranking XGBoost feature-importance entries were standardized code indicators. These features indicate whether a specific ontology code was present in the protocol, primary endpoint, or candidate endpoint representation. Table C.1 summarizes the translated codes that appeared among the feature-importance outputs. The table is not intended as a complete clinical interpretation of the model, but as a diagnostic view of which standardized medical concepts were useful during tree construction.

Feature / code	Source	Short name	Meaning / use
`protocol_ncit C27996`	NCIt	Myocardial infarction	Cardiovascular event concept referring to myocardial infarction or heart attack. Its presence suggests that protocol-level cardiovascular event terminology was useful for separating candidate endpoints.
`protocol_ncit C16725`	NCIt	Antibody	Immunologic/protein concept referring to antibodies. This type of code can indicate biological or intervention-related protocol context.
`protocol_ncit C1647`	NCIt	Trastuzumab	Drug/intervention concept for a HER2-targeting monoclonal antibody. This shows that molecule- or intervention-specific codes can influence endpoint ranking.
`protocol_cdisc C98791`	CDISC	Tolerability study	Trial-type concept describing a study that assesses whether adverse effects can be tolerated by participants. This is relevant for separating safety-oriented endpoint candidates.
`endpoint_ncit C158286`	NCIt / CDISC	Drug–drug interaction study	Study-design concept describing evaluation of interactions between drugs, including possible effects on disposition, efficacy, or safety.
`primary_ncit C129955`	NCIt	Pulmonary artery wedge pressure	Hemodynamic measurement concept. It is clinically relevant for cardiovascular, pulmonary-pressure, and right-heart-function contexts.
`endpoint_loinc 54779-4`	LOINC	Heart failure assessment	LOINC assessment concept for heart failure during an assessment period. This represents endpoint-level heart-failure context.
`protocol_loinc 30934-4`	LOINC	BNP in serum/plasma	Biomarker concept for B-type natriuretic peptide concentration in serum or plasma, commonly relevant in heart failure and cardiovascular protocols.
`protocol_loinc 53632-6`	LOINC	Breast cancer antigen 225	Tissue immune-stain concept for Breast Cancer Antigen 225. This illustrates that some high-ranking code features may reflect domain-specific protocol context outside the cardiovascular examples.

Table C.1: Examples of standardized code features appearing in the XGBoost feature-importance output for the Stage 2 leave-one-out model. These are code-indicator features used during tree construction, not SHAP explanations.

C.1.1 SHAP mean absolute Importance

Several of the high-ranking mean absolute SHAP features were also standardized code indicators. In contrast to the XGBoost feature-importance table above, these features indicate which code indicators had the largest average influence on individual candidate endpoint scores. Table C.2 summarizes selected translated code features from the SHAP ranking.

Feature / code	Source	Short name	Meaning / use
endpoint_loinc 103980-9	LOINC	Pharmacy preference	Endpoint-side concept describing a reported pharmacy preference. This type of feature reflects candidate endpoint context rather than protocol-level disease information.
endpoint_ncit C221736	NCIt	Disease-specific use consent	Functional concept indicating that study-generated data may only be used for research on a specific disease or related condition.
endpoint_ncit C203456	NCIt	Two-dimensional imaging	Diagnostic-procedure concept for imaging techniques that produce two-dimensional images. This is relevant for candidate endpoints involving imaging-based measurements.
endpoint_loinc 103814-0	LOINC	CT study observation	Endpoint-side observation concept for narrative CT study observations. This indicates imaging or procedure-related endpoint content.
endpoint_loinc 50402-7	LOINC	Blood pressure after transfusion	Quantitative blood-pressure concept for systolic and diastolic blood pressure after transfusion. This represents candidate endpoint information related to hemodynamic measurement.
endpoint_ncit C113114	NCIt	GPNMB gene	Gene or genome concept for the GPNMB gene, which may be related to regulation of cell proliferation. This reflects candidate endpoint or biomarker-specific information.
primary_ncit C88067	NCIt	Health assessment before exposure	Health-care activity concept describing an assessment performed before the exposure of interest. This can represent primary-endpoint timing or baseline-assessment context.
primary_ncit C198208	NCIt	Year of last follow-up	Temporal concept indicating the four-digit year associated with the last clinical follow-up. This reflects time-related primary-endpoint information.
primary_ncit C142432	NCIt	Clinical laboratory	Organization concept referring to a laboratory that analyzes patient or subject samples in healthcare or clinical research. This can indicate laboratory-based primary-endpoint context.
protocol_ncit C147465	NCIt / CDISC	Contrast transthoracic echocardiography	Diagnostic-procedure concept for contrast-enhanced transthoracic echocardiography. This is relevant for protocols involving cardiac imaging and functional assessment.
protocol_ncit C105760	NCIt	Clinical event pattern	General clinical-event concept describing the pattern of a clinical event. This can contribute to protocol-level event or outcome context.
protocol_ncit C197870	NCIt	Chronic systolic heart failure	Disease concept describing chronic congestive heart failure due to reduced left-ventricular contractility. This provides protocol-level cardiovascular disease context.

Table C.2: Examples of standardized code features appearing in the mean absolute SHAP importance output for the Stage 2 leave-one-out model. These code indicators describe protocol, primary-endpoint, and candidate-endpoint concepts that influenced individual prediction scores.

D

Full Pipeline Predictions

D.1 Selected Predictions on 40 Unseen Protocols

This appendix provides selected qualitative examples from the full-pipeline evaluation on 40 previously unseen Tier C protocols. These examples are not used as additional quantitative performance estimates. Instead, they illustrate how the raw ranked predictions behaved beyond strict exact-match evaluation.

All examples shown here use the same leave-one-out convention as the final full-pipeline evaluation. Specifically, the hidden endpoint corresponds to secondary endpoint index 0 for each protocol, i.e., the first secondary endpoint was hidden and the remaining secondary endpoints were retained as observed context. For each selected protocol, all 10 ranked predictions from the raw prediction output are shown.

Protocol NCT05993559

Hidden endpoint: Overall survival.

Rank	Predicted endpoint
1	Percentage of Participants With the Indicated Period of Distant Recurrence-free Survival (Time to Distant Recurrence)
2	Overall Survival (OS)
3	Percent of Participants With Disease-free Survival (DFS)
4	Disease-free Survival at Month 12
5	Progression-free Survival (PFS)
6	Number of Participants with Ventricular Arrhythmias
7	Time to Symptom Progression
8	Percentage of Participants With Total Pathological Complete Response (tpCR), According to Local Pathologist Assessment
9	Percentage of Participants With Total Pathological Complete Response (tpCR), According to Local Pathologist Assessment
10	Number of Participants with Adverse Events

Table D.1: Top-10 full-pipeline predictions for protocol NCT05993559.

This example shows strong concept-level recovery. Although strict exact-match evaluation did not count the prediction as a hit, the model ranked overall survival and related survival endpoints near the top.

Protocol NCT05881382

Hidden endpoint: Time to cardiovascular death.

Rank	Predicted endpoint
1	Key secondary endpoint: Time to first event of cardiovascular death or hospitalisation for heart failure
2	Subjects Included in the Composite Endpoint of CV Death or Hospitalization Due to Heart Failure.
3	Events Included in the Composite Endpoint of Recurrent Hospitalizations Due to Heart Failure and CV Death.
4	Change From Baseline in the Composite Autonomic Symptom Score (COMPASS 31) Total Score at Week 52
5	Minnesota Living with Heart Failure Score
6	Change in Kansas City Cardiomyopathy Questionnaire Total Symptom Score (KCCQ-TSS) from baseline to Month 6.
7	The Hierarchical Composite Endpoint: Percentage of Wins of Participant Pairs
8	KCCQ
9	Incidence of Major Adverse Cardiovascular Events (MACE)
10	Change From Baseline in Total Symptom Score (TSS) of the Kansas City Cardiomyopathy Questionnaire (KCCQ)

Table D.2: Top-10 full-pipeline predictions for protocol NCT05881382.

This example shows that the pipeline retrieved the cardiovascular event family. The highest-ranked candidates broadened the hidden endpoint into composite cardiovascular death and heart-failure hospitalization outcomes.

Protocol NCT05732727

Hidden endpoint: Time to end-stage kidney disease.

Rank	Predicted endpoint
1	Time to First Event in Composite Renal Endpoint: Chronic Dialysis, Renal Transplant or Sustained Reduction of eGFR(CKD-EPI)cr
2	Change from baseline in diastolic blood pressure at each study visit
3	Change From Baseline in Sitting SBP, Sitting DBP and Sitting Pulse Pressure (PP)
4	Time to first occurrence of death from kidney failure, chronic dialysis* or renal transplant or onset of sustained reduction of $\geq 50\%$ estimated glomerular filtration rate (eGFR) or onset of sustained eGFR (CKD-EPI)cr < 10 mL/min/1.73 m^2
5	Total Number of Confirmed Incidences of a Composite Endpoint of Worsening Renal Function
6	Time to the First Event in the Composite Renal Endpoint: Chronic Dialysis, Renal Transplant, or Sustained Reduction in eGFR (CKD-EPI)cr
7	DBP at baseline compared to Day 4 or hospital discharge if earlier
8	eGFR (CKD-EPI) cr Slope of Change From Baseline
9	eGFR (CKD-EPI) cr Slope of Change From Baseline
10	Change in eGFR (CKD-EPI)

Table D.3: Top-10 full-pipeline predictions for protocol NCT05732727.

This example shows a coherent renal-progression neighborhood. The model retrieved endpoints related to dialysis, renal transplant, kidney failure, eGFR reduction, worsening renal function, and renal-function change.

Protocol NCT05768230

Hidden endpoint: Right ventricular area fractional change (RV FAC) 48 hours after randomization.

Rank	Predicted endpoint
1	right ventricular function
2	Change From Baseline in RV Function as Measured by Tricuspid Annular Plane Systolic Excursion (TAPSE) as Assessed by Echocardiograms at 12 Weeks
3	Right Ventricular systolic function
4	RV function
5	Pulmonary function
6	Effect of Potassium Nitrate (KNO3) on Left Ventricle (LV) Diastolic Function: Left Atrial Volume Index
7	Change From Baseline in Right Ventricular Fractional Area Change (RVFAC) at Week 24
8	Change in left ventricular diastolic function at 6 and 12 months.
9	Effect of Potassium Nitrate (KNO3) on Left Ventricle (LV) Diastolic Function: E/e' Ratio
10	2.3 Assessment of left ventricular diastolic function in HFpEF

Table D.4: Top-10 full-pipeline predictions for protocol NCT05768230.

This example corresponds to the detailed case study in Section 5.6.1. The model retrieved several right-ventricular function concepts, including the hidden RVFAC concept, but with a timeframe inherited from another historical protocol.

Protocol NCT06469645

Hidden endpoint: Change in left ventricular ejection fraction.

Rank	Predicted endpoint
1	Wall stress by cardiac MRI
2	Change in LV wall thickening
3	Change in LV end-diastolic volume index (EDVI)
4	Diastolic function (E/e')
5	Change From Baseline in the Cardiac Biomarker: N-terminal Prohormone of B-type Natriuretic Peptide (NT-proBNP)
6	Quality of life at 6 and 12 months assessed using the five-dimensional EQ-5D questionnaire.
7	Quality of Life Questionnaire (Kansas City Cardiomyopathy Questionnaire)
8	Quality of life measured by Minnesota Questionaire and SF12 Questionaire
9	Change From Baseline in N-Terminal Pro B-Type Natriuretic Peptide (NT-ProBNP) Level at Month 6
10	Change in LV end-diastolic wall thickness

Table D.5: Top-10 full-pipeline predictions for protocol NCT06469645.

This example shows retrieval of related left-ventricular imaging and remodeling concepts. However, the model did not recover the specific ejection-fraction endpoint, illustrating the difference between clinical-neighborhood retrieval and exact endpoint recovery.

Protocol NCT05966415

Hidden endpoint: Change in 24 hour blood pressure, daytime blood pressure, nighttime blood pressure in Endocalyx treated patients when compared with subject receiving Placebo.

Rank	Predicted endpoint
1	Safety: peripheral oedema, deterioration of heart failure, increase in body weight during 26 week treatment period, adverse events (AEs), laboratory parameters.
2	Change in Body Weight
3	Change in body weight.
4	Incidence of serious adverse events as events per patient month including infection, stroke, VAD failure, and all serious adverse events. Quality of Life measures - MLHFQ, NYHA class, Neurocognitive assessments.
5	Change in quality of life as assessed by Minnesota Living with Heart Failure Questionnaire scores
6	Change From Baseline in Sitting SBP, Sitting DBP and Sitting Pulse Pressure (PP)
7	Change From Baseline in Supine Systolic Blood Pressure (SBP) and Diastolic Blood Pressure (DBP) Assessed at Weeks 4, 8, 12, and 16/Early Withdrawal
8	Change from baseline in diastolic blood pressure at each study visit
9	Change in body weight (%)
10	Part A: Change in Systolic Blood Pressure (SBP), Diastolic Blood Pressure (DBP) and Mean Arterial Pressure (MAP) at 0.5, 1, 2, 3, 4, 5 and 6 Hours

Table D.6: Top-10 full-pipeline predictions for protocol NCT05966415.

This example shows that relevant blood-pressure candidates were present in the ranked output, but not at the very top. It therefore illustrates a ranking limitation rather than a complete candidate-pool failure.

Overall, these examples support the interpretation from Section 5.6: the full pipeline often retrieved clinically related endpoint concepts, but the raw output still required review and adaptation. In particular, the predictions show recurring issues with endpoint specificity, duplicate or overlapping endpoint ideas, and historical endpoint wording that may not match the target protocol's required measurement window or clinical context.