

UNIVERSITÀ
DI PAVIA

FACOLTA' DI INGEGNERIA
DIPARTIMENTO DI INGEGNERIA INDUSTRIALE E DELL'INFORMAZIONE

CORSO DI LAUREA MAGISTRALE IN INGEGNERIA BIOMEDICA

TESI DI LAUREA

IMPLEMENTAZIONE DI UN FRAMEWORK EPISTEMOLOGICO PER IL RAGIONAMENTO CLINICO BASATO SU LLM: DALL'ARCHITETTURA DI BASE AL SUPPORTO ALLA DIAGNOSI DIFFERENZIALE

Candidato Lorenzo Stafforini

Relatore: Prof. Enea Parimbelli
Correlatori: Prof. Giovanna Nicora, Dott.ssa Laura Bergomi, Dott. Simone Rancati

A.A. 2025/2026

INDICE GENERALE

Abstract

CAPITOLO 1: Introduzione

- **1.1** I CDSS e l'evoluzione dell'intelligenza artificiale in medicina
- **1.2** L'avvento dei *Large Language Models* (LLM) nella sanità digitale
    - *1.2.1 Natural Language Processing* (NLP) e architettura transformer
    - *1.2.2* Funzionamento, validazione e impiego clinico degli LLM
- **1.3** Criticità degli LLM in ambito medico
    - *1.3.1* Allucinazioni, incertezza e *bias*
    - *1.3.2* Il problema della "scatola nera" e la necessità di tracciabilità clinica
    - *1.3.3* Errori di omissione e rischi di sicurezza
- **1.4** Il contributo originale della tesi

CAPITOLO 2: Stato dell'Arte e fondamenti teorici

- **2.1** Il ragionamento clinico
- **2.2** Il *Select-and-Test Model* (ST-Model)
- **2.3** *Benchmarking* degli LLM in ambito clinico
    - *2.3.1 Script Concordance Testing* (SCT)
    - *2.3.2* SCT-Bench: dataset di *Script Concordance Testing*
    - *2.3.3* Caso di studio: il *benchmark* CPC-Bench
- **2.4** Il paradigma *Human-in-the-Loop* (HITL) per l'allineamento uomo-macchina

CAPITOLO 3: L'Architettura di base del framework epistemologico

- **3.1** Implementazione dell'ST-Model come grafo a stati
- **3.2** Struttura del grafo
    - *3.2.1* Nodo di astrazione
    - *3.2.2* Nodi di abduzione
    - *3.2.3* Nodo di ranking
    - *3.2.4* Nodo di deduzione
    - *3.2.5* Nodo di induzione
- **3.3** Output strutturato

- **3.4** Limiti dell'architettura base: staticità e assenza del ciclo di retroazione

CAPITOLO 4: Ottimizzazione dell'architettura ed integrazioni

- **4.1** Ottimizzazione del nodo di astrazione
    - *4.1.1* Motivazione e ruolo del nodo
    - *4.1.2* Transizione a *few-shot prompting*, vincoli strutturali e il compromesso della *lossy abstraction*
    - *4.1.3* Consolidamento semantico dello schema dati (soluzione ibrida)
    - *4.1.4* Disaccoppiamento dello stato del grafo e gestione del carico informativo
- **4.2** Integrazione delle nuove evidenze cliniche
    - *4.2.1* Definizione del *benchmark* e pre-processing dei dati
    - *4.2.2* Approccio I: valutazione del punteggio SCT (*zero-shot vs. framework-aware*)
    - *4.2.3* Approccio II: aggiornamento induttivo e riordinamento delle ipotesi
    - *4.2.4* Approccio III: riesecuzione completa del ciclo cognitivo
- **4.3** Progettazione e sviluppo dell'interfaccia grafica
    - *4.3.1* Analisi dei requisiti e scelte tecnologiche
    - *4.3.2* Flusso dati multi-pagina e gestione dei processi (*threading e polling*)
    - *4.3.3* Integrazione del meccanismo *Human-in-the-Loop*
    - *4.3.4* Rappresentazione strutturata dei dati e tracciabilità
    - *4.3.5* Gestione dello stato globale, persistenza e logging strutturato

CAPITOLO 5: Valutazione sperimentale e risultati

- **5.1** Setup sperimentale e protocollo di valutazione
    - *5.1.1* Configurazione dell'ambiente di esecuzione e selezione degli LLM
    - *5.1.2* Metriche di valutazione
- **5.2** Valutazione del nodo di astrazione
    - *5.2.1* Confronto delle prestazioni diagnostiche
    - *5.2.2* Analisi del carico informativo
- **5.3** Valutazione dell'integrazione di nuove evidenze cliniche
    - *5.3.1* Analisi dei risultati del framework SCT (Approccio I)
    - *5.3.2* Confronto delle prestazioni: re-ranking (Approccio II) vs. re-abduzione (Approccio III) nel recupero delle omissioni diagnostiche

- **5.4** Dimostrazione funzionale e tracciabilità dell'interfaccia grafica

CAPITOLO 6: Discussioni e prospettive future

- **6.1** Bilanciamento tra vincoli di contesto e l*ossy abstraction*
- **6.2** Limiti dell'implementazione attuale e natura preliminare del ciclo di retroazione post-induzione
- **6.3** L'interfaccia grafica come strumento di allineamento uomo-macchina
- **6.4** Sviluppi futuri
    - *6.4.1* Integrazione del ciclo non-monotonico nell'interfaccia: inserimento di nuove evidenze cliniche
    - *6.4.2* Potenziamento epistemologico tramite moduli RAG (*Retrieval-Augmented Generation*)
    - *6.4.3* Infrastruttura di inferenza modello-indipendente tramite SGLang
    - *6.4.4* Standardizzazione delle diagnosi
    - *6.4.5* Validazione prospettica in ambienti di simulazione clinica
    - *6.4.6* Estensione del framework ai compiti di pianificazione terapeutica e di follow-up

CAPITOLO 7: Conclusioni

Bibliografia

Abstract

Alcuni modelli linguistici di grandi dimensioni (Large Language Models, LLM) hanno raggiunto prestazioni diagnostiche paragonabili a quelle dei medici nella valutazione di casi complessi, ma il loro impiego clinico resta limitato da allucinazioni, eccessiva sicurezza sulle decisioni, limitata interpretabilità del processo decisionale ed errori di omissione. Questa tesi parte da un framework di Rancati, Bergomi et al. presentato all'AIME 2026, che implementa il *Select-and-Test Model* di Ramoni et al. come grafo a stati orchestrato con LangGraph. Il modello scompone il ragionamento diagnostico in cinque fasi successive: la sintesi dei dati clinici (astrazione), la generazione di ipotesi diagnostiche (abduzione), il loro ordinamento (ranking), la previsione dei reperti attesi per ciascuna ipotesi (deduzione) e il confronto con il caso osservato (induzione). Il lavoro estende il framework lungo tre direttrici: l'ottimizzazione del nodo di astrazione, l'integrazione di nuove evidenze cliniche e lo sviluppo di un'interfaccia grafica con supervisione umana.

Nella prima fase il sistema trasforma il testo del caso clinico in un elenco di reperti distinti (sintomi, esami, farmaci, malattie pregresse). Questa fase è stata modificata fornendo al modello un esempio da imitare (*few-shot prompting*) e regole esplicite con lo scopo di ridurre il volume testuale fornito alle fasi successive ed evitare il raggruppamento in una sola voce di elementi diversi. Le fasi successive lavorano poi solo su questo elenco, senza rileggere il testo originale. Per gestire l'arrivo di nuove informazioni cliniche sono stati confrontati tre approcci. Nel primo, il modello valuta quanto un nuovo dato renda più o meno probabile una diagnosi, con (approccio *framework-aware)* e senza (approccio *zero-shot*) l'aiuto del ragionamento già prodotto dal sistema, usando come metodo di valutazione lo *Script Concordance Test*. Nel secondo, le ipotesi già generate vengono riordinate alla luce del nuovo dato. Nel terzo, il ciclo di ragionamento viene rieseguito, in modo da poter generare anche diagnosi nuove. L'interfaccia, sviluppata in Dash, integra un meccanismo *Human-in-the-Loop* configurabile con sospensione controllata del ragionamento, modifica del ranking, tracciabilità delle variazioni e persistenza dello stato.

La valutazione del nodo di astrazione, condotta con Gemini 2.5 Pro su 10 casi del CPC-Bench, mostra che la configurazione ottimizzata riduce del 30.3% il numero medio di caratteri dell'output fornito ai nodi successivi. La riduzione del volume testuale si accompagna però a un calo dell'accuratezza Top-1 dal 47% al 36%, mentre la Top-10 aumenta dal 72% al 74%.

Nella valutazione SCT, condotta su 42 quesiti con Gemini 2.5 Pro e Gemini 2.5 Flash, fornire al modello il ragionamento già prodotto dal framework aumenta la concordanza media con le risposte degli esperti di 2.28 e 3.15 punti percentuali, rispettivamente, rispetto all'approccio zero-shot. Le differenze non sono statisticamente significative. Gli Approcci II e III, basati rispettivamente sul riordinamento delle ipotesi esistenti e sulla riesecuzione del ciclo completo, mostrano un miglioramento medio della posizione della diagnosi corretta di 0.42 e 0.54 posizioni nelle esecuzioni incluse nell'analisi del rango.

Il contributo principale della tesi è la guida del ragionamento dell'LLM attraverso un processo esplicito, ispezionabile e supervisionabile, più che un guadagno prestazionale dimostrato. I limiti principali sono la numerosità ridotta dei campioni, il rischio di contaminazione dei *benchmark* pubblici e il carattere ancora preliminare del ciclo di retroazione.

Abstract

Some Large Language Models (LLMs) have achieved diagnostic performance comparable to that of physicians in the evaluation of complex cases, but their clinical use remains limited by hallucinations, overconfidence in decisions, limited interpretability of the decision-making process, and omission errors. This thesis builds on a framework by Rancati, Bergomi et al., presented at AIME 2026, which implements Ramoni et al.'s Select-and-Test Model as a state graph orchestrated with LangGraph. The model decomposes diagnostic reasoning into five sequential phases: synthesis of clinical data (abstraction), generation of diagnostic hypotheses (abduction), ordering of hypotheses (ranking), prediction of the findings expected for each hypothesis (deduction), and comparison with the observed case (induction). This work extends the framework in three directions: optimization of the abstraction node, integration of new clinical evidence, and development of a graphical interface with human supervision.

In the first phase, the system transforms the clinical case text into a list of distinct findings (symptoms, tests, medications, and past medical conditions). This phase was modified by providing the model with an example to imitate (few-shot prompting) and explicit rules, with the aim of reducing the volume of text passed to subsequent phases and preventing different elements from being grouped into a single entry. Subsequent phases then operate only on this list, without re-reading the original text. To handle the arrival of new clinical information, three approaches were compared. In the first, the model evaluates how much a new piece of information makes a diagnosis more or less likely, with (the framework-aware approach) and without (the zero-shot approach) the support of the reasoning already produced by the system, using the Script Concordance Test as the evaluation method. In the second, previously generated hypotheses are re-ranked considering the new information. In the third, the reasoning cycle is repeated, allowing new diagnoses to be generated. The interface, developed in Dash, integrates a configurable Human-in-the-Loop mechanism with controlled suspension of reasoning, modification of the ranking, traceability of changes, and state persistence.

The evaluation of the abstraction node, conducted with Gemini 2.5 Pro on 10 CPC-Bench cases, shows that the optimized configuration reduces the mean character count of the output passed to subsequent nodes by 30.3%. However, this reduction in text volume is accompanied by a decrease in Top-1 accuracy from 47% to 36%, while Top-10 accuracy increases from 72% to 74%.

In the SCT evaluation, conducted on 42 items with Gemini 2.5 Pro and Gemini 2.5 Flash, providing the model with the reasoning already produced by the framework increases mean concordance with expert responses by 2.28 and 3.15 percentage points, respectively, compared with the zero-shot approach. The differences are not statistically significant. Approaches II and III, based on re-ranking existing hypotheses and repeating the complete reasoning cycle, respectively, show a mean improvement in the rank of the correct diagnosis of 0.42 and 0.54 positions in the runs included in the rank analysis.

The main contribution of this thesis is the guidance of LLM reasoning through an explicit, inspectable, and supervisable process, rather than a demonstrated performance gain. The main limitations are the small sample sizes, the risk of contamination from public benchmarks, and the still preliminary nature of the feedback loop.

CAPITOLO 1: Introduzione

1.1 I CDSS e l'evoluzione dell'intelligenza artificiale in medicina

I sistemi di supporto alle decisioni cliniche (*Clinical Decision Support Systems*, CDSS) nascono dall'esigenza di assistere il medico nel processo diagnostico e terapeutico attraverso strumenti computazionali capaci di elaborare conoscenza medica in modo sistematico. Le prime applicazioni dell'intelligenza artificiale (IA) in questo ambito risalgono agli anni '70, quando furono sviluppati i primi sistemi esperti basati su regole, tra cui MYCIN [1], progettato per il supporto alla diagnosi e alla terapia delle infezioni batteriche, e INTERNIST-I [2], orientato alla diagnosi differenziale in medicina interna. Questi sistemi rappresentavano la conoscenza clinica attraverso associazioni empiriche tra manifestazioni e patologie, codificate sotto forma di schemi operativi derivati dall'esperienza pratica di esperti, piuttosto che da autentiche teorie fisiopatologiche.

Nonostante il loro valore innovativo, questi sistemi mostrarono limiti strutturali significativi. Erano definiti fragili, poiché le loro prestazioni degradavano bruscamente quando il problema presentato eccedeva i confini della conoscenza codificata nelle regole di base [3]. Inoltre, non erano in grado di generare spiegazioni soddisfacenti delle proprie conclusioni, potendo esporre soltanto gli schemi operativi che componevano la propria base di conoscenza, né di affrontare problemi "nuovi", non anticipati dai progettisti del sistema [3]. A questi limiti si aggiungeva la difficoltà di acquisire e mantenere aggiornate le basi di conoscenza.

Per superare queste criticità, la ricerca in IA in ambito medico si è orientata verso un'analisi a due livelli dei sistemi basati sulla conoscenza (*Knowledge Based Systems*, KBS): un livello epistemologico, dedicato alla definizione dell'ontologia del dominio e del modello inferenziale impiegato per elaborarla, e un livello computazionale, relativo ai metodi e ai formalismi scelti per l'implementazione [3]. Questa distinzione ha permesso di isolare gli aspetti concettuali del ragionamento clinico, indipendenti dalla specifica tecnologia di implementazione, da quelli propriamente implementativi, ponendo le basi per modelli epistemologici generali del ragionamento medico, tra cui il Select-and-Test Model di Ramoni et al. [3], approfondito nel Capitolo 2.

Parallelamente, gli studi di cognizione medica hanno arricchito questo quadro teorico investigando i processi cognitivi effettivamente impiegati dai clinici esperti, evidenziando l'importanza di strategie di ragionamento sia analitiche sia non analitiche, nonché il ruolo della struttura della conoscenza medica negli errori diagnostici [4]. Il contributo congiunto dell’ingegneria della conoscenza e delle scienze cognitive ha caratterizzato l'evoluzione dei CDSS, prima dell'avvento dei sistemi basati sull'apprendimento automatico e, più recentemente, dei modelli linguistici di grandi dimensioni, che hanno radicalmente trasformato le modalità di rappresentazione ed elaborazione della conoscenza clinica.

1.2 L'avvento dei *Large Language Models* (LLM) nella sanità digitale

1.2.1 *Natural Language Processing* (NLP) e architettura transformer

Nell'ambito dei CDSS, l'elaborazione del linguaggio naturale (*Natural Language Processing*, NLP) ha risposto alla necessità di convertire l'informazione non strutturata dei testi clinici in rappresentazioni trattabili dai sistemi informatici [5]. I compiti primari del NLP biomedico sono stati formalizzati attraverso modelli dedicati a task specifici:

- Estrazione delle informazioni (*Information Extraction*, IE) e riconoscimento di entità nominate (*Named Entity Recognition*, NER): identificazione ed estrazione automatica di entità cliniche pertinenti (es. sintomi, analisi di laboratorio) all'interno di referti o altre documentazioni mediche;
- Classificazione del testo: assegnazione automatica di etichette diagnostiche, codici nosologici (come la classificazione ICD [20]) o classi di rischio ai pazienti sulla base della cartella clinica;
- Risposta a quesiti medici (*Medical Question Answering,* QA): produzione di risposte a domande cliniche o biomediche, eventualmente con il supporto di fonti esterne;
- Riepilogo clinico: sintesi e compressione di note cliniche complesse, referti specialistici o lettere di dimissione ospedaliera, al fine di ridurre il carico della documentazione del personale sanitario [5].

L'efficacia nell'esecuzione di questi compiti è mutata con l'introduzione dell'architettura Transformer [6]. Fondata sui meccanismi di attenzione (*self-attention*), tale architettura consente di calcolare in parallelo le relazioni semantiche e sintattiche tra i token del testo, le unità elementari in cui un LLM scompone una query e su cui basa la generazione delle risposte, lungo l'intera sequenza di input. Ciò elimina i vincoli di elaborazione sequenziale tipici delle reti ricorrenti (come RNN, reti neurali ricorrenti) e migliora la modellazione delle dipendenze a lungo raggio, ovvero la capacità di estrapolare correlazioni tra elementi del testo indipendentemente dalla loro distanza [6].

A partire dall'architettura Transformer originaria si sono sviluppate tre classi di modelli:

- Architetture e*ncoder-only*: elaborano la sequenza di input in modo bidirezionale e vengono addestrati utilizzando compiti di mascheramento (*Masked Language Modeling*, MLM) [7], una tecnica che permette di nascondere determinate parole in un testo lasciando al modello il compito di predirle correttamente. La loro capacità di catturare il contesto completo del testo rende queste architetture particolarmente adatte a compiti di comprensione, classificazione e NER;
- Architetture e*ncoder-decoder*: sono progettate per generare una sequenza in uscita a partire da quella in ingresso. Vengono tipicamente utilizzate in compiti di traduzione e riformulazione [6];
- Architetture d*ecoder-only*: modelli autoregressivi addestrati sul compito di predire il token successivo basandosi sui token precedenti, compresi quelli del testo fornito in ingresso. Tale sistema costituisce la base della maggioranza degli LLM oggi utilizzati [8].

1.2.2 Funzionamento, validazione e impiego clinico degli LLM

Il vantaggio dei moderni LLM risiede anche nella capacità di svolgere compiti diversi, sfruttando il meccanismo *In-Context Learning* (apprendimento contestualizzato) [9]: il modello utilizza le istruzioni e gli eventuali esempi presenti nel prompt per svolgere il compito, senza aggiornare i propri parametri durante l'uso. In questo contesto è possibile definire due modalità di immissione del prompt (input fornito contenente le istruzioni): *zero-shot,* in cui all'LLM viene fornita la sola descrizione del problema, senza alcun esempio di supporto, e *few-shot,* il cui prompt contiene più esempi dimostrativi per guidare il ragionamento [9].

ChatGPT (OpenAI), basato inizialmente sui modelli GPT-3.5 e GPT-4, ha rappresentato il punto di svolta nella percezione pubblica e clinica del potenziale degli LLM, grazie alla sua interattività conversazionale e a prestazioni prossime o pari al livello umano in compiti cognitivi trasversali, inclusa la medicina [5]. In particolare, i modelli sopra citati hanno raggiunto un livello di prestazione sufficiente al superamento degli *United States Medical Licensing Examinations* (USMLE), esame nazionale di abilitazione alla pratica della medicina negli Stati Uniti, alimentando l'ipotesi che le applicazioni basate su LLM fossero ormai pronte per un impiego in ambito clinico, educativo o di ricerca [5].

Studi successivi hanno confermato questa traiettoria, riscontrando prestazioni diagnostiche di modelli LLM di ultima generazione (in particolare della serie o1 di OpenAI) elevate e paragonabili a quelle di centinaia di medici esperti su casi clinici complessi tratti dalle *Clinicopathological Conferences* (CPC) del *New England Journal of Medicine* (NEJM)*,* sia in condizioni di casi clinici standardizzati (utilizzati a scopo didattico) sia in scenari reali di pronto soccorso [10]. L'adozione clinica degli LLM ha conosciuto una crescita altrettanto rapida sul piano pratico: si stima che due terzi dei medici statunitensi utilizzino oggi questi strumenti nella pratica quotidiana, e che circa un medico su cinque li consulti regolarmente per ottenere una seconda opinione diagnostica [11]. Tale diffusione, spinta anche dall'integrazione di funzionalità basate su LLM nelle cartelle cliniche elettroniche e negli strumenti di trascrizione automatizzata delle visite, pone le basi per l'urgenza di una valutazione rigorosa e standardizzata delle capacità di ragionamento clinico di questi sistemi, oggetto delle sezioni successive.

1.3 Criticità degli LLM in ambito medico

Nonostante le prestazioni ormai comparabili o superiori a quelle umane su numerosi compiti diagnostici, l'impiego degli LLM in ambito clinico solleva criticità specifiche che ne condizionano l'adozione sicura e responsabile. Tali criticità possono essere ricondotte a tre categorie principali, esaminate nelle sezioni seguenti: i fenomeni di allucinazione e *bias* cognitivo, la limitata interpretabilità del processo decisionale (il cosiddetto problema della "scatola nera"), e gli errori di omissione con le relative implicazioni per la sicurezza del paziente.

1.3.1 Allucinazioni, incertezza e *bias*

Uno dei fenomeni più critici associati agli LLM è rappresentato dalle cosiddette allucinazioni, ossia la generazione di contenuti inesatti e che si discostano dalla fonte originaria dell'informazione (tipicamente i dati di addestramento), pur mantenendo una plausibilità superficiale che ne rende difficile l'identificazione [12]. Nel quadro della teoria della generalizzazione, le allucinazioni possono essere interpretate come un'applicazione eccessivamente ampia di schemi appresi durante l'addestramento. Questo fenomeno si affianca a un'ulteriore criticità, evidenziata da studi specifici sulla capacità degli LLM di aggiornare il proprio giudizio in risposta a nuove evidenze cliniche: l'analisi delle risposte fornite dai modelli nei compiti di *Script Concordance Testing* (SCT), test ideato per misurare il grado di concordanza tra le risposte dei candidati e quelle di un panel di esperti nell'interpretare nuovi dati relativamente ad un caso in esame [13], ha mostrato una tendenza a sottoutilizzare la risposta neutra e a scegliere più frequentemente le categorie estreme della scala Likert (-2/+2), suggerendo difficoltà nel riconoscere quando una nuova informazione non dovrebbe modificare il giudizio, indicando una forma di fiducia eccessiva (*overconfidence*) che si accentua paradossalmente nei modelli esplicitamente ottimizzati per il ragionamento [14]. Tale evidenza suggerisce che le tecniche di *chain-of-thought,* ovvero un metodo di scomposizione (adottato dagli LLM) di un problema in una sequenza di passaggi intermedi, non garantiscano, da sole, una migliore gestione dell'incertezza [14].

Questi fenomeni presentano interessanti analogie con i *bias* cognitivi documentati nel ragionamento clinico umano. Ad esempio, il *confirmation bias*, ovvero la tendenza ad interpretare e cercare le informazioni per convalidare le proprie convinzioni e sminuire qualsiasi prova contraria, porta i clinici a ricercare attivamente evidenze coerenti con l'ipotesi diagnostica iniziale, trascurando dati potenzialmente indicativi di diagnosi alternative [4]. Se un simile pattern di ancoraggio alle ipotesi iniziali dovesse manifestarsi anche negli LLM impiegati come strumenti di supporto decisionale, l'assenza di un meccanismo esplicito di revisione delle ipotesi rischierebbe di amplificare, anziché correggere, gli errori diagnostici tipici del ragionamento umano.

1.3.2 Il problema della "scatola nera" e la necessità di tracciabilità clinica

Un secondo ordine di criticità riguarda la natura intrinsecamente opaca dei modelli linguistici di grandi dimensioni. A differenza dei sistemi esperti di prima generazione, la cui base di conoscenza era esplicitamente rappresentata da regole curate da esperti e ispezionabile, gli LLM fondano il proprio comportamento su rappresentazioni distribuite apprese statisticamente, non permettendo di anticipare come applicheranno le conoscenze a nuovi scenari [12]. Le tecniche di spiegazione *post-hoc* offerte dalla letteratura sull'*Explainable AI* (XAI) forniscono soluzioni parziali, applicabili caso per caso, ma non restituiscono una comprensione sistematica del processo inferenziale sottostante [12].

In ambito clinico, questa opacità pone criticità specifiche di natura etico-legale e di affidabilità diagnostica: i clinici e le istituzioni sanitarie necessitano di comprendere le motivazioni di una raccomandazione algoritmica, di identificarne i potenziali errori e di verificarne la coerenza con la conoscenza e gli standard di pratica medica, prima di poterla impiegare in decisioni cliniche responsabili [15]. Il "modello di ragionamento" impiegato dagli LLM diagnostici rimane tuttavia

spesso implicito, incorporato nel modello, piuttosto che espresso come un sistema esplicito e controllabile. Ciò rende difficile separare la conoscenza clinica, la strategia inferenziale e la generazione linguistica, e fornire ai clinici una traccia di ragionamento trasparente che ne supporti la validazione.

Questa problematica ripropone, in una nuova veste tecnologica, la dicotomia tra sistemi "profondi" (*deep*) e "superficiali" (*shallow*) già discussa in relazione ai primi CDSS: mentre un sistema *deep* separa esplicitamente ontologia e modello inferenziale rendendo la conoscenza di dominio accessibile e ispezionabile, un sistema *shallow* compila implicitamente entrambe le componenti in un'unica rappresentazione non trasparente [3]. Il contributo originale di questa tesi, descritto nella Sezione 1.4 e sviluppato nei Capitoli 3 e 4, si colloca esattamente in risposta a questa esigenza di esplicitazione del processo inferenziale.

1.3.3 Errori di omissione e rischi di sicurezza

Un terzo ordine di criticità, di natura più direttamente clinica, riguarda la sicurezza delle raccomandazioni prodotte dagli LLM quando impiegati come strumenti di supporto decisionale nella pratica reale. Lo studio NOHARM (*Numerous Options Harm Assessment for Risk in Medicine*) ha valutato 31 LLM su 100 casi clinici reali di consulto elettronico tra medico di base e specialista, coprendo 10 specialità e raccogliendo 12747 annotazioni esperte su 4249 opzioni di gestione clinica [11]. Nei modelli valutati, raccomandazioni con potenziale di danno grave compaiono fino al 22.2% dei casi con gli errori di omissione, ossia la mancata raccomandazione di un'azione clinicamente appropriata, quale la richiesta di un esame diagnostico critico, responsabili del 76.6% degli errori gravi complessivi [11]. Questo dato è particolarmente rilevante poiché la letteratura sugli errori diagnostici documenta che le omissioni costituiscono una delle principali cause di danno grave al paziente anche nel ragionamento clinico umano [11].

Un ulteriore elemento critico emerso da questo studio riguarda la correlazione limitata tra le prestazioni di sicurezza clinica e i *benchmark* convenzionali di conoscenza medica o generale: le prestazioni migliori sul piano della sicurezza non sono predette in modo affidabile da *proxy* comuni della capacità del modello, quali la dimensione dei parametri o la presenza di modalità di ragionamento esplicito [11]. Questo risultato mette in discussione l'assunto per cui un miglioramento delle prestazioni sui *benchmark* di conoscenza medica si traduca automaticamente in una gestione clinica più sicura, e sottolinea la necessità di misurare esplicitamente la sicurezza come dimensione prestazionale distinta dall'accuratezza diagnostica.

Va inoltre osservato che l'affidamento eccessivo dei clinici su output generati dall'IA rischia di rendere gli errori residui degli LLM più difficili da rilevare proprio nei sistemi più performanti, un rischio che si acuisce nella transizione, già in corso in molti contesti sanitari, da contesto "*human-in-the-loop*" (in cui il clinico rivede ogni singolo output) a "*human-on-the-loop*" (in cui il clinico supervisiona e sottopone a verifica il sistema, senza rivedere sistematicamente ogni raccomandazione) [11]. Questo scenario rafforza ulteriormente la necessità, discussa nella Sezione 2.4, di meccanismi espliciti di supervisione umana (*Human-in-the-Loop*, HITL) integrati nell'architettura dei sistemi di supporto decisionale.

1.4 Il contributo originale della tesi

Alla luce delle criticità delineate nella Sezione 1.3, allucinazioni e *overconfidence*, opacità del processo inferenziale, ed errori di omissione con conseguenze potenzialmente gravi per la sicurezza del paziente, il presente lavoro di tesi estende un framework computazionale già implementato, progettato per organizzare il ragionamento degli LLM in fasi esplicite e ispezionabili.

Il punto di partenza concettuale è il *Select-and-Test Model* (ST-Model) proposto da Ramoni, Stefanelli, Magnani e Barosi [3], che formalizza il ragionamento diagnostico come un ciclo iterativo di generazione e verifica di ipotesi, scomposto nei quattro tipi di inferenza mutuati dalla filosofia di Peirce [16], astrazione, abduzione, deduzione e induzione. Tale modello, sviluppato originariamente per i sistemi esperti a base di conoscenza degli anni '90, viene qui reinterpretato e implementato come un grafo a stati orchestrato tramite LangGraph, in cui ciascuna fase epistemologica corrisponde a un nodo computazionale con output strutturato e validato, secondo l'approccio proposto in un lavoro di Rancati, Bergomi et al. presentato all'AIME 2026 [15].

La tesi estende il framework lungo tre direttrici. La prima riguarda il nodo di astrazione, modificato per rappresentare separatamente le informazioni cliniche e ridurre il volume testuale fornito alle fasi successive. La seconda riguarda l'integrazione di nuove evidenze cliniche, mediante tre approcci di valutazione e aggiornamento delle ipotesi. La terza riguarda lo sviluppo di un'interfaccia grafica che permette di ispezionare gli output e intervenire sul ranking attraverso un meccanismo *Human-in-the-Loop*. Questi contributi sono descritti nel Capitolo 4 e valutati nel Capitolo 5.

Il contributo originale della tesi si articola pertanto su due livelli complementari: da un lato, un livello sperimentale, che quantifica, attraverso il *benchmark* CPC-Bench [17] e il dataset SCT pre-elaborato [14], l'effetto di specifiche scelte architetturali (l'ottimizzazione del nodo di astrazione e i tre approcci di integrazione di nuove evidenze) sulle prestazioni diagnostiche e sulla stabilità del framework; dall'altro, un livello ingegneristico e di interazione uomo-macchina, che traduce i principi epistemologici dell'ST-Model in un'infrastruttura software concreta, dotata di *logging* strutturato (file log di output), tracciabilità algoritmica e meccanismi di validazione umana, ponendo le basi per un impiego responsabile e verificabile di sistemi di supporto decisionale basati su LLM in ambito clinico.

CAPITOLO 2: Stato dell'Arte e fondamenti teorici

2.1 Il ragionamento clinico

Il ragionamento clinico costituisce il nucleo cognitivo della pratica medica e rappresenta il processo attraverso il quale i clinici raccolgono, interpretano e integrano informazioni per formulare diagnosi, pianificare terapie e prendere decisioni in condizioni di incertezza [18]. Lo studio di tali processi, noto come *medical cognition*, ha avuto come obiettivo quello di comprendere come gli esperti prendano decisioni e di tradurre tali modelli in sistemi computazionali di supporto [4].

I primi studi sul ragionamento medico hanno proposto modelli a due stadi composti da una fase di generazione di ipotesi seguita da una fase di valutazione [4]. Successivamente, è stato consolidato il modello ipotetico-deduttivo, secondo il quale il medico formula precocemente un numero limitato di ipotesi sulla base dei primi dati disponibili e raccoglie informazioni selettive per confermarle o confutarle [4]. Il processo comprende acquisizione di dati, generazione delle ipotesi e successiva interpretazione dei reperti come positivi, negativi o non contribuenti rispetto a ciascuna delle ipotesi.

Tuttavia, il modello ipotetico-deduttivo non descrive in modo completo il ragionamento degli esperti, poiché non spiega le migliori prestazioni degli esperti rispetto ai medici specializzandi [18]. In situazioni familiari, i clinici esperti possono riconoscere rapidamente segni e sintomi noti e formulare un'ipotesi iniziale senza passare attraverso una lunga ricerca di alternative. Questo approccio, guidato dai dati e dall'esperienza precedente, è spesso definito *forward reasoning* o riconoscimento di pattern. Al contrario, quando il caso è poco familiare, complesso o ambiguo, il clinico può procedere in modo più analitico e guidato dalle ipotesi, ricercando informazioni utili aggiuntive [4].

Questa apparente contraddizione può essere interpretata considerando che il ragionamento clinico non è riducibile a un'unica strategia. Il ricorso alle diverse metodologie dipende dal livello di esperienza, dalla familiarità con il problema e dal grado di incertezza del caso e l'intero processo decisionale deve essere concepito come un processo flessibile che alterna strategie diverse a seconda della situazione [18].

Un quadro più generale può essere ottenuto integrando il modello ipotetico-deduttivo con la nozione di abduzione. L'abduzione è un tipo di inferenza che genera spiegazioni provvisorie a partire da dati osservati, senza garantire la verità logica della conclusione [4]. Nel contesto diagnostico, l'abduzione permette di identificare un insieme di ipotesi capaci di spiegare il quadro clinico, per poi sottoporle a test deduttivi e induttivi. Questo approccio ciclico, generazione abduttiva, deduzione delle conseguenze attese, induzione tramite confronto con i dati osservati, costituisce la base teorica del Select-and-Test Model (ST-Model), descritto nella Sezione 2.2.

Patel et al. [4] descrivono come la conoscenza medica sia organizzata nella memoria degli esperti e come tale organizzazione influenzi il ragionamento. È stato osservato che la conoscenza biomedica di base (biochimica, fisiologia, anatomia) e la conoscenza clinica (schemi di malattia, associazioni sintomo-diagnosi) non sono pienamente integrate dagli esperti durante il ragionamento di routine. Gli esperti tendono a produrre spiegazioni più generali, utilizzando meno concetti biomedici espliciti rispetto agli studenti, pur mantenendo accuratezza diagnostica. Questo fenomeno è stato interpretato attraverso il meccanismo di "incapsulamento della conoscenza": durante la formazione clinica, le nozioni biomediche vengono aggregate all'interno di concetti clinici più ampi, mantenendo la stessa

capacità esplicativa ma con meno dettagli espliciti [4]. Inoltre, l'acquisizione dell'esperienza non comporta necessariamente un miglioramento lineare in tutti i compiti: alcuni studi descrivono un effetto intermedio [4], con prestazioni temporaneamente inferiori in determinate fasi della formazione.

Oltre ai modelli basati su regole e prototipi, è stato evidenziato il ruolo del ragionamento per similarità [18]. È stato osservato come gli esperti utilizzino non solo schemi astratti (prototipi) ma anche ricordi di casi specifici precedentemente incontrati (esempi), confrontando il caso corrente con istanze concrete archiviate in memoria [4][18]. Gli studi sperimentali indicano che la diagnosi si basa almeno in parte sulla somiglianza con un esempio specifico già memorizzato, e che questo processo è rapido e non accessibile all'introspezione [4]. Studi su diagnosi visive (radiografie, immagini dermatologiche, ECG) hanno mostrato che la similarità con casi precedenti contribuisce in modo rilevante alle diagnosi formulate, e che errori di interpretazione possono essere influenzati dalla storia clinica nota del paziente, anche quando le informazioni di contesto non sono rilevanti per l'interpretazione [18].

Questa evidenza suggerisce che il ragionamento clinico non sia esclusivamente analitico (basato su decomposizione logica dei sintomi e applicazione di regole), ma includa anche componenti non-analitiche basate sul riconoscimento di pattern familiari. In questa prospettiva, l'esperienza clinica emerge dalla combinazione di strategie analitiche (ipotetico-deduttive) e non-analitiche (riconoscimento di pattern, recupero di casi simili), la cui prevalenza dipende dal contesto, dalla complessità e dall'esperienza pregressa [4].

Il ragionamento clinico non è esente da errori. Sono state proposte classificazioni degli errori medici, distinguendo tra *slips* (sviste, esecuzione errata di un'azione corretta) e *mistakes* (errori, esecuzione corretta di un'azione errata), ulteriormente articolati in errori di intenzione specifica dell'azione, esecuzione, percezione, interpretazione e valutazione [4]. Nel contesto del ragionamento diagnostico, la maggior parte degli errori rientra nella categoria dei *mistakes*, dovuti a conoscenza incompleta o scorretta e *bias* cognitivi (ad esempio, *confirmation bias*). È ragionevole dunque dedurre che il ragionamento non analitico, per la sua rapidità, sia più esposto all'errore [18]; per questo è stato suggerito di affiancare al riconoscimento di regolarità strategie analitiche di riduzione dell'incertezza [4].

Quando i pazienti non rispondono come atteso, l'équipe clinica può riunirsi per rivedere il caso: rivaluta le ipotesi iniziali e ne considera di alternative, ragionando in termini di probabilità, di diagnosi differenziale e di cause biologiche [4]. Questi risultati sottolineano la necessità di modelli di ragionamento che tengano conto non solo della cognizione individuale, ma anche delle dinamiche collaborative, della comunicazione e dell'uso di tecnologie esterne (cartelle cliniche elettroniche, sistemi di supporto decisionale) che mediano e trasformano i processi cognitivi.

La comprensione dei meccanismi del ragionamento clinico umano ha implicazioni dirette per la progettazione di sistemi di intelligenza artificiale in medicina. I primi sistemi esperti (ad esempio, MYCIN [1], INTERNIST-I [2]) erano basati su regole empiriche e associazioni sintomo-malattia, senza incorporare modelli causali profondi della fisiopatologia ("sistemi superficiali") [4][1][2]. Sistemi successivi hanno tentato di integrare modelli causali "profondi" del funzionamento e malfunzionamento dell'organismo, analoghi ai modelli usati nella fisica qualitativa.

Tuttavia, l'evidenza empirica suggerisce che gli esperti umani non utilizzano sistematicamente conoscenza biomedica "profonda" durante il ragionamento di routine, ma fanno affidamento su conoscenza clinica compilata, derivata dall'esperienza e dall'incapsulamento di relazioni causa-effetto in regole dirette recuperabili in contesti specifici [4]. Questo solleva interrogativi sul compromesso tra profondità esplicativa ed efficienza operativa nei sistemi di supporto decisionale: un sistema che ragiona in modo troppo profondo può risultare costoso in termini di calcolo computazionale e poco allineato con i processi cognitivi umani, mentre un sistema puramente empirico può mancare di flessibilità e capacità esplicativa in casi atipici.

2.2 Il *Select-and-Test Model* (ST-Model)

Il Select-and-Test Model (ST-Model), proposto da Ramoni, Stefanelli, Magnani e Barosi alla fine degli anni '80 e consolidato nei primi anni '90, costituisce un modello epistemologico sviluppato per la formalizzazione del ragionamento clinico dei CDSS [3]. Il modello nasce dall'esigenza di fornire un quadro concettuale unificato per il ragionamento medico, capace di descrivere in modo formale e astratto le fasi inferenziali coinvolte nei compiti di diagnosi, pianificazione terapeutica e monitoraggio, indipendentemente dalle scelte implementative specifiche.

L'ST-Model concepisce il ragionamento medico come un ciclo iterativo di selezione e verifica di ipotesi. Il ciclo si articola in due macro-fasi:

- Fase di selezione delle ipotesi: a partire dai dati osservati nel paziente, il sistema genera un insieme di ipotesi plausibili capaci di spiegare il quadro clinico;
- Fase di verifica delle ipotesi: ciascuna ipotesi viene testata, derivando le conseguenze attese e confrontandole con lo stato reale del paziente, per confermarla, falsificarla o richiederne un ulteriore approfondimento.

Questo schema ciclico riflette la natura non monotona del ragionamento clinico: nuove informazioni possono portare al ritiro di ipotesi precedentemente accettate e all'adozione di nuove spiegazioni [3]. Il modello articola il ragionamento in fasi di astrazione, abduzione, deduzione e induzione, alcune di queste derivate dalla filosofia di Peirce [16], integrate da un ordinamento delle ipotesi secondo criteri di rilevanza clinica.

- Astrazione: i dati grezzi del caso clinico (sintomi, segni, esami) vengono trasformati in una rappresentazione compatta e strutturata di caratteristiche clinicamente rilevanti. Questa fase corrisponde al riconoscimento preliminare di pattern e alla categorizzazione dei reperti in classi semantiche;
- Abduzione: a partire dai reperti astratti, il sistema genera ipotesi diagnostiche plausibili. L'abduzione è suddivisa in due sottofasi:
    - *Abduzione non focalizzata*: genera categorie generali di ipotesi (famiglie diagnostiche, sindromi) per delimitare lo spazio delle soluzioni possibili;
    - *Abduzione focalizzata*: raffina le ipotesi generali in diagnosi più specifiche, in relazione reciproca di dettaglio, competizione o complementarità;

- Ranking: le ipotesi generate vengono ordinate secondo criteri di preferenza clinica, al fine di prioritizzare quelle da testare per prime. Tali criteri includono:
    - Parsimonia: preferenza per le ipotesi che richiedono il minor intervento terapeutico, a parità di esito atteso;
    - Costo: descrive il costo economico complessivo delle risorse richieste per la gestione dell'ipotesi;
    - Curabilità: indica il grado di disponibilità ed efficacia di un trattamento terapeutico per la gestione di una patologia;
    - Pericolosità: entità del rischio clinico relativo ad un ritardo nella diagnosi o nell'attuazione della terapia;
- Deduzione: per ciascuna ipotesi vengono derivate le conseguenze cliniche attese, come manifestazioni, risultati di esami e andamento della condizione;
- Induzione: le conseguenze attese vengono confrontate con le osservazioni disponibili per valutare le ipotesi e stabilire se siano necessarie ulteriori informazioni.

La struttura ciclica permette di modellare esplicitamente il carattere non monotono del ragionamento: nuove informazioni possono innescare nuovi cicli di abduzione, deduzione e induzione [3].

Il modello è stato progettato per essere generico e applicabile a tre compiti fondamentali del ragionamento medico:

- Diagnosi: il ciclo parte dai dati osservati e mira a produrre la migliore spiegazione dello stato del paziente;
- Pianificazione terapeutica: il ciclo parte dalla diagnosi e dai dati osservati, astrae problemi terapeutici, abduce terapie candidate, le ordina per priorità e ne testa le conseguenze attese sul decorso clinico;
- Monitoraggio: il ciclo verifica se la terapia in corso sta producendo gli effetti attesi; in caso contrario, può innescare nuovi cicli di diagnosi o pianificazione.

Questa flessibilità rende l'ST-Model un quadro epistemologico unificato, capace di descrivere in modo coerente diversi aspetti del ragionamento clinico. La Figura 2.1 rappresenta il ciclo di selezione e verifica delle ipotesi su cui si basa il modello.

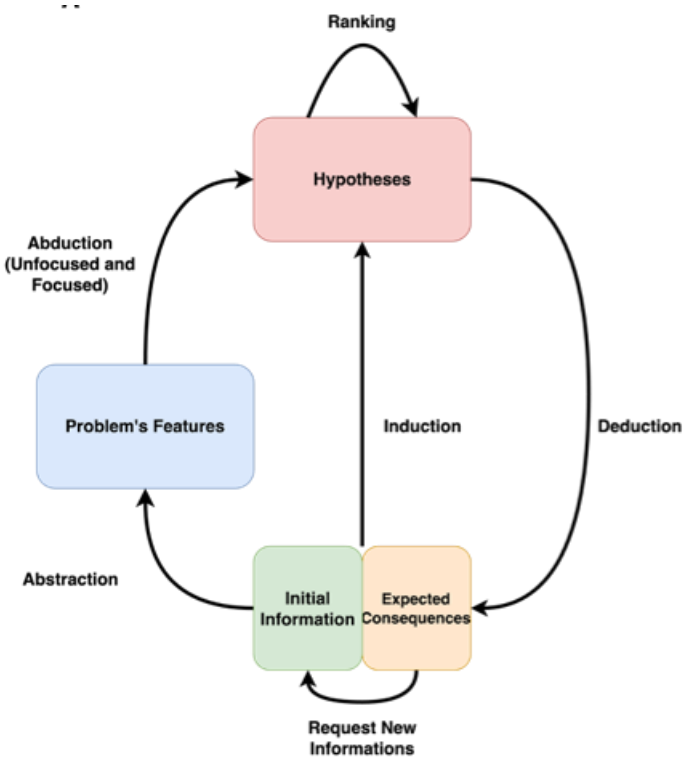


Figura 2.1 - Schema del Select-and-Test Model [15].

2.3 *Benchmarking* degli LLM in ambito clinico

La valutazione delle capacità di ragionamento clinico degli LLM richiede *benchmark* specifici, capaci di andare oltre la semplice accuratezza su domande a risposta multipla e di catturare aspetti quali la generazione di diagnosi differenziali, la pianificazione di test, l'interpretazione di immagini e l'aggiornamento del giudizio in risposta a nuove informazioni. Due riferimenti utilizzati all'interno del presente lavoro di tesi sono l'SCT-Bench [21] e il CPC-Bench [17].

2.3.1 *Script Concordance Testing* (SCT)

Lo *Script Concordance Testing* (SCT) è uno strumento consolidato in ambito di formazione medica, progettato per misurare la competenza decisionale in condizioni di incertezza. Ogni item SCT presenta una vignetta clinica iniziale, un'ipotesi diagnostica o gestionale, e una nuova informazione; il rispondente deve valutare come questa nuova informazione modifichi la probabilità dell'ipotesi, su una scala Likert a 5 punti (da "molto meno probabile" a "molto più probabile", [-2; +2]). Le risposte sono valutate rispetto a un panel di esperti, con credito pieno per la risposta più frequente e credito parziale per le risposte minoritarie, riflettendo il realistico disaccordo tra esperti in casi ambigui [14].

Lo SCT è quindi particolarmente pertinente per la presente tesi, in quanto misura esplicitamente la capacità di aggiornare il ragionamento in risposta a nuove evidenze: una competenza che il framework proposto mira a supportare attraverso i tre approcci descritti nella Sezione 4.2.

2.3.2 SCT-Bench: dataset di *Script Concordance Testing*

SCT-Bench è un *benchmark* pubblico progettato per valutare il ragionamento clinico degli LLM mediante dati di SCT a confronto con il ragionamento umano. I dati sulle risposte fornite sono stati raccolti intervistando medici specialisti esperti e interrogando i modelli sullo stesso quesito. Il *benchmark* è stato presentato da McCoy et al. [14] ed è stato reso disponibile attraverso il *repository* GitHub associato [21][22]. Il progetto raccoglie complessivamente 750 quesiti SCT provenienti da 10 dataset internazionali, relativi a differenti discipline cliniche e livelli di formazione medica [22]. Alcuni risultati basati su tale progetto mostrano che gli LLM ottengono prestazioni inferiori rispetto agli esperti umani, pur eguagliando o superando gli studenti in alcuni test. In particolare, i modelli tendono a sottoutilizzare le risposte neutre (nessun cambiamento di probabilità) e a sovrastimare gli spostamenti estremi (+2/-2), suggerendo una forma di *overconfidence* e una difficoltà nel riconoscere quando una nuova informazione non dovrebbe alterare il giudizio [14].

Il set completo di 750 quesiti comprende dati aggiuntivi che non sono distribuiti pubblicamente dal repository [21], il quale comprende solo quelli provenienti dai dataset *Open Medical SCT* e *Adelaide SCT.* Di conseguenza, il presente lavoro utilizza esclusivamente i dati disponibili nel sottoinsieme pubblico, con le trasformazioni adottate e descritte nella Sezione 4.2.1.

2.3.3 Caso di studio: il *benchmark* CPC-Bench

Il CPC-Bench è un *benchmark* basato su oltre un secolo di *Clinicopathological Conferences* (CPC), discussioni tenute da specialisti in cui vengono analizzati casi clinici complessi confrontando la diagnosi proposta con l'esito reale derivante da esami di biopsia o autopsia, pubblicate dal *New England Journal of Medicine* (NEJM). Il *benchmark* include 7102 casi CPC (1923-2025) e definisce 10 compiti distinti che coprono aspetti multipli del ragionamento clinico [17]:

- Generazione di diagnosi differenziale
- Pianificazione di test e gestione
- Ricerca bibliografica per affermazioni cliniche
- Diagnosi differenziale a step sequenziali
- Domande a risposta multipla (*Question Answering*, QA)
- Ragionamento clinico (evidenze a supporto o a sfavore per ciascuna diagnosi)
- Diagnosi differenziale con omissione di reperti normali
- *Visual Question-Answering* (VQA) su immagini cliniche
- Diagnosi differenziale da sole immagini e tabelle
- NEJM *Image Challenge* (quiz a risposta multipla su immagini)

Nella valutazione su 377 casi CPC, il modello o3 di OpenAI ha raggiunto un'accuratezza Top-1 (diagnosi corretta che occupa la prima posizione tra le ipotesi) del 60% e Top-10 (diagnosi corretta che rientra nelle prime 10 ipotesi generate) dell'84%, superando un riferimento costituito da 20 medici internisti [17]. Tuttavia, le prestazioni risultano inferiori nei compiti di ricerca bibliografica e

interpretazione di immagini, evidenziando limiti attuali nell'integrazione multimodale e nel recupero di letteratura [17].

Il *benchmark* include anche "Dr. CaBot", un agente IA in grado di generare diagnosi differenziali e presentazioni video nello stile di un relatore esperto di CPC. In confronti in cieco tra testi prodotti da CaBot e da esperti umani, i medici hanno attribuito erroneamente l'origine del testo in 46 prove su 62, pari al 74%, e hanno valutato CaBot più favorevolmente su qualità complessiva, giustificazione diagnostica e qualità delle citazioni [17].

Un sottoinsieme di 10 casi clinici distribuiti pubblicamente dal *benchmark* CPC-Bench è stato utilizzato per la valutazione dei risultati di questo lavoro di tesi.

2.4 Il paradigma *Human-in-the-Loop* (HITL) per l'allineamento uomo-macchina

L'integrazione di sistemi di IA nei processi decisionali clinici richiede meccanismi di allineamento tra il ragionamento automatico e il giudizio umano. Il paradigma *Human-in-the-Loop* (HITL) prevede che l'esperto umano mantenga un ruolo attivo nel ciclo inferenziale, validando, correggendo o integrando gli output del sistema prima che questi vengano utilizzati per decisioni successive [12].

Ilievski et al. analizzano l'allineamento tra esseri umani e sistemi di IA lungo tre dimensioni: il significato attribuito alla generalizzazione, i metodi usati per ottenerla e i criteri impiegati per valutarla [12]. Gli autori descrivono inoltre differenze tra i modi in cui esseri umani e sistemi statistici generalizzano: gli esseri umani tendono a ricorrere all'astrazione e all'apprendimento di concetti, mentre i sistemi statistici apprendono regolarità dai dati e possono trasferirle a nuove osservazioni. Queste differenze evidenziano la necessità di valutare non solo le risposte finali, ma anche come i sistemi di IA affrontano casi nuovi e differenti da quelli già incontrati in fase di addestramento [12].

Il framework presentato in questa tesi integra esplicitamente un meccanismo HITL configurabile, descritto nel Capitolo 4.3.

L'integrazione dell'HITL risponde a esigenze di sicurezza e affidabilità in ambito clinico. In contesti ad alto rischio (ad esempio, triage in pronto soccorso, decisioni terapeutiche critiche), la supervisione umana può intercettare errori del modello (ad esempio, ipotesi mancanti, ranking inappropriati) e correggerli prima che influenzino decisioni successive. La tracciabilità degli interventi permette inoltre di distinguere le modifiche introdotte dall'operatore da quelle prodotte dal sistema e di analizzarle successivamente [12].

CAPITOLO 3: L'Architettura di base del framework epistemologico

3.1 Implementazione dell'ST-Model come grafo a stati

Il presente capitolo descrive l'architettura di base del framework sviluppato da Rancati, Bergomi et al. [15], assunta in questo lavoro come punto di partenza. I contributi implementativi e sperimentali introdotti nella tesi (cioè l'ottimizzazione del nodo di astrazione, l'integrazione di nuove evidenze e l'interfaccia HITL) sono presentati nel Capitolo 4 e valutati nel Capitolo 5. La struttura concettuale deriva dall'ST-Model, descritto nel Capitolo 2, nel quale il ragionamento diagnostico è rappresentato come un processo iterativo di generazione, selezione e verifica di ipotesi.

Nel sistema implementato, ciascuna fase è rappresentata da un nodo del grafo. I nodi ricevono lo stato corrente dell'elaborazione, eseguono una specifica trasformazione e restituiscono uno o più aggiornamenti dello stato. L'orchestrazione del flusso è realizzata tramite LangGraph, una libreria Python che permette di organizzare un processo computazionale come un grafo con nodi e transizioni [23].

Lo stato condiviso del grafo è rappresentato da una struttura denominata `EpistState`. Nella configurazione base, tale struttura contiene:

- il testo clinico del caso (`case_text`);
- il modello linguistico da utilizzare (`model`);
- il numero massimo di ipotesi da generare (`top_k`);
- l'output dell'astrazione (`abstraction`);
- gli output delle due fasi di abduzione (`abduction_unfocused`, `abduction_focused`);
- il ranking diagnostico (`ranking`)
- il piano deduttivo (`deduction`)
- il risultato dell'induzione (`induction`)
- variabili di controllo del ciclo, tra cui `reranked`, `iter` e `stop`

Ogni nodo utilizza i campi dello stato necessari al proprio compito. Questa organizzazione mantiene distinti i dati clinici, i risultati intermedi e le variabili di controllo del flusso.

L'architettura base segue il percorso illustrato in Figura 3.1.

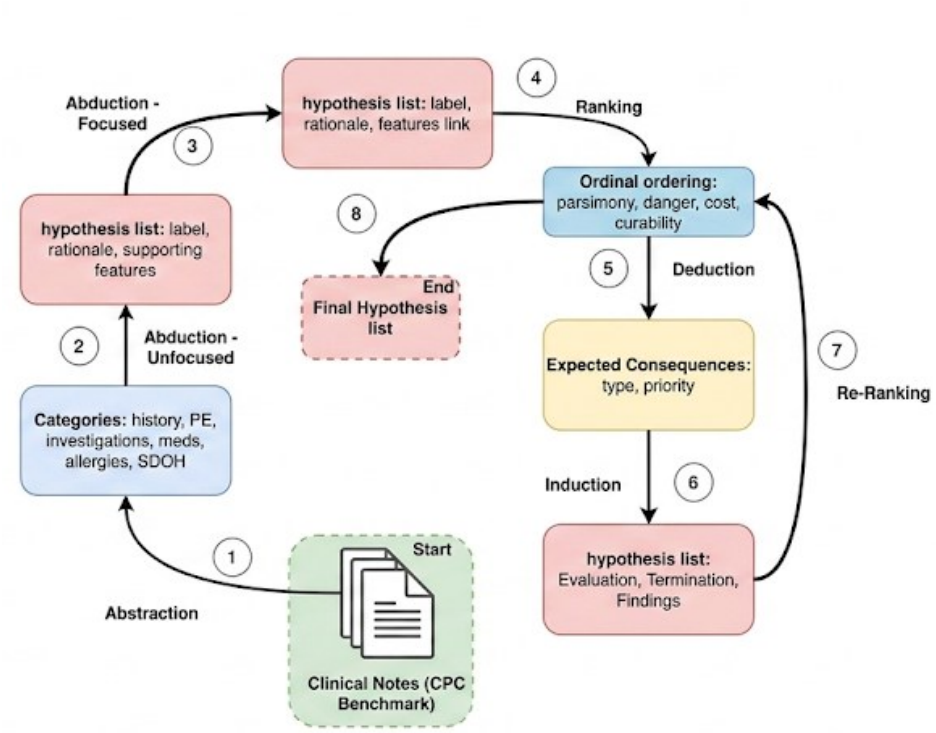


Figura 3.1 – Struttura dell'architettura [15].

Dopo aver ricevuto in input il caso clinico, il processo inizia estraendo le caratteristiche rilevanti (astrazione) ed elaborando una lista di ipotesi in due passaggi: il primo individua famiglie diagnostiche o sindromi (abduzione non focalizzata), mentre il secondo propone diagnosi più specifiche (abduzione focalizzata). Il primo ranking viene costruito a partire dalle ipotesi prodotte dall'abduzione focalizzata. Successivamente, il sistema esegue deduzione e induzione. Al termine dell'induzione, il ranking viene richiamato una seconda volta, questa volta utilizzando le ipotesi annotate con i relativi esiti induttivi. Il secondo ranking rappresenta quindi il riordinamento finale delle ipotesi sulla base del confronto tra conseguenze attese e osservazioni disponibili.

Nel grafo, la decisione sul percorso successivo al nodo di ranking è affidata a una funzione di instradamento condizionale. Se il primo ranking è stato eseguito, il grafo prosegue verso deduzione e induzione; se invece il ranking è già stato aggiornato dopo l'induzione, l'esecuzione termina. Questa logica consente di eseguire il ranking prima e dopo la verifica delle ipotesi e di terminare il processo dopo il secondo ordinamento. Non costituisce però un ciclo completo di acquisizione di nuove informazioni, limite discusso nella Sezione 3.4.

3.2 Struttura del grafo

3.2.1 Nodo di astrazione

Il nodo di astrazione costituisce il punto di ingresso del caso clinico. Il suo compito è trasformare la narrativa in un insieme di reperti strutturati e organizzati per categoria.

Nella versione base, il nodo riceve il testo clinico completo e lo sottopone ad un LLM. Le categorie previste comprendono:

- storia della malattia presente
- storia clinica remota
- farmaci correnti
- allergie
- storia familiare
- esame obiettivo
- indagini diagnostiche
- determinanti sociali di salute

Nella versione base, ogni reperto è costituito dai campi `finding` (informazione estratta) ed `explanation` (descrizione ritrovata nel testo originale associata all'informazione). La gestione esplicita dello stato dell'evidenza tramite `present`, `absent` e `uncertain` (presente, assente e incerto), così come la sintesi temporale e la deduplicazione rafforzata, sono state introdotte nelle modifiche descritte nella Sezione 4.1.

È importante distinguere questa versione dall'implementazione ottimizzata: nel sistema base, l'astrazione non rappresenta ancora un vincolo assoluto per tutti i nodi successivi, poiché il testo clinico completo viene passato anche alle funzioni di abduzione e ranking. Il successivo disaccoppiamento tra narrativa grezza e rappresentazione astratta costituisce quindi una modifica architetturale autonoma.

3.2.2 Nodi di abduzione

La fase abduttiva è suddivisa in due nodi distinti: abduzione non focalizzata e abduzione focalizzata. Questa separazione riflette la progressiva restrizione dello spazio delle ipotesi prevista dall'ST-Model.

Abduzione non focalizzata

L'abduzione non focalizzata riceve il caso clinico e la lista dei reperti astratti. Il suo obiettivo è costruire uno spazio iniziale di ipotesi ampio, composto da famiglie diagnostiche, sindromi o categorie di cause.

Il prompt richiede esplicitamente al modello di:

- proporre ipotesi di livello generale;
- evitare diagnosi eccessivamente specifiche;
- coprire più reperti clinici con ogni famiglia diagnostica;
- associare a ogni ipotesi una motivazione;
- indicare i reperti astratti che la supportano;

- evitare ipotesi chiaramente ridondanti o irrilevanti.

L'output è vincolato allo schema `AbductionHypothesesList`, costituito da una lista di oggetti `DiagnosticHypothesis`. Ogni oggetto contiene:

- il nome della diagnosi o della famiglia diagnostica;
- una spiegazione del collegamento con il quadro clinico;
- l'elenco dei reperti di supporto.

Abduzione focalizzata

Il secondo nodo abduttivo riceve la rappresentazione astratta e le ipotesi generali prodotte dal nodo precedente. Il suo compito è trasformare le famiglie diagnostiche in ipotesi più specifiche.

Il prompt distingue tre possibili relazioni tra ipotesi focalizzate e ipotesi generali:

- Perfezionamento `(refinement)`, quando una famiglia di ipotesi viene specializzata in una diagnosi più precisa;
- Competizione `(competition)`, quando vengono proposte diagnosi alternative all'interno della stessa famiglia;
- Complementarità `(complement)`, quando viene introdotta una diagnosi concomitante capace di spiegare altri reperti del caso.

Anche in questa fase, ogni ipotesi deve essere collegata a uno o più reperti astratti. Il modello non deve ancora stabilire il ranking delle ipotesi: il compito di ordinamento viene riservato al nodo successivo.

L'output della fase focalizzata mantiene la struttura `AbductionHypothesesList`, ma il contenuto è costituito da entità diagnostiche più specifiche rispetto a quelle generate durante l'abduzione non focalizzata.

3.2.3 Nodo di ranking

Il nodo di ranking ordina le ipotesi prodotte dall'abduzione focalizzata. Dopo l'induzione viene richiamato sulle stesse ipotesi, arricchite dalle valutazioni ottenute. L'ordinamento è espresso tramite un rango intero, nel quale il valore 1 identifica l'ipotesi prioritaria.

Il ranking non rappresenta esclusivamente una stima della probabilità diagnostica. Nel framework, la priorità è determinata considerando quattro criteri:

- Parsimonia `(parsimony)`: preferenza per le ipotesi che richiedono il minor intervento terapeutico, a parità di esito atteso;
- Pericolosità `(danger)`: priorità alle condizioni per le quali un ritardo diagnostico o terapeutico potrebbe causare un danno maggiore;
- Costo `(cost)`: considerazione delle risorse necessarie per confermare e gestire l'ipotesi;

- Curabilità (`curability`): preferenza, a parità degli altri elementi, per le condizioni per le quali è possibile ottenere una cura o un miglioramento significativo.

Lo schema prodotto in output (Rank) contiene il rango e una motivazione testuale per ciascuno dei quattro criteri. Gli oggetti `RankedDiagnosticHypothesis` associano quindi il nome di ciascuna diagnosi alla relativa struttura Rank.

Per garantire la coerenza dell'output, il modello viene vincolato a includere tutte le ipotesi ricevute senza aggiungerne di nuove, modificarne il nome o ometterne alcune.

Il nodo di ranking viene utilizzato due volte:

1. dopo l'abduzione focalizzata, per ottenere il ranking preliminare delle ipotesi;
2. dopo l'induzione, per riordinare le ipotesi sulla base degli esiti del test deduttivo-induttivo.

Nel secondo ranking, le valutazioni induttive vengono considerate insieme ai quattro criteri clinici. L'ordinamento può quindi cambiare in base alle evidenze favorevoli o contrarie a ciascuna ipotesi.

3.2.4 Nodo di deduzione

Il nodo di deduzione implementa la fase predittiva del modello. A partire dal ranking diagnostico, il sistema produce, per ogni ipotesi, un insieme di conseguenze cliniche attese nel caso in cui l'ipotesi fosse corretta.

Lo schema `ExpectedFinding` classifica ogni conseguenza secondo tre dimensioni:

- Descrizione (`description`): illustrazione della conseguenza attesa;
- Tipologia (`kind`): classificazione della conseguenza. Può assumere i valori:
  - `manifestation`, per sintomi o segni clinici;
  - `test_result`, per risultati di laboratorio, imaging, ECG o altri esami;
  - `clinical_course`, per elementi relativi all'evoluzione temporale della condizione;
- Priorità (`priority`): importanza clinica o informativa della sua verifica, associata ai possibili valori `high`, `medium` o `low` (alta, media o bassa). Una conseguenza ad alta priorità è particolarmente informativa per confermare o confutare l'ipotesi, oppure è rilevante dal punto di vista della sicurezza clinica.

Il prompt richiede al modello di produrre un insieme compatto di conseguenze, privilegiando:

- reperti altamente discriminativi;
- elementi utili ad orientare ulteriori accertamenti;
- risultati frequentemente associati all'ipotesi;
- informazioni relative alla progressione temporale;
- eventuali risposte attese a trattamenti o interventi.

Il risultato complessivo è rappresentato dallo schema `DeductionPlan`, che contiene una lista di oggetti `HypothesisDeduction`, ciascuno associato a una singola ipotesi diagnostica e alle relative conseguenze previste.

Il nodo non deve modificare il ranking ricevuto né introdurre nuove ipotesi. La sua funzione è esclusivamente quella di esplicitare quali osservazioni dovrebbero essere ricercate per testare ciascuna spiegazione.

3.2.5 Nodo di induzione

Il nodo di induzione confronta le conseguenze attese generate dalla deduzione con il caso clinico osservato. Esso rappresenta la fase di verifica delle ipotesi e produce un giudizio sia a livello dei singoli reperti sia a livello dell'ipotesi complessiva.

Ogni conseguenza dedotta viene classificata attraverso uno dei seguenti stati:

- Confermato (`confirmed`), quando il reperto atteso è supportato da una o più evidenze osservate;
- Non osservato (`not_observed`), quando il reperto non è menzionato o non è stato valutato;
- Contraddetto (`contradicted`), quando il caso contiene informazioni incompatibili con la conseguenza attesa.

Per ogni classificazione, il modello produce inoltre un commento esplicativo. Successivamente, per ciascuna ipotesi, viene assegnato un giudizio globale:

- Plausibile (`plausible`), quando le aspettative principali sono confermate e non sono presenti contraddizioni rilevanti;
- Confutato (`refuted`), quando una o più aspettative fondamentali sono contraddette oppure il quadro complessivo è incompatibile con l'ipotesi.

Il nodo produce anche una raccomandazione relativa alla prosecuzione del processo:

- `sufficient_explanation_reached`, quando l'ipotesi fornisce una spiegazione sufficientemente soddisfacente;
- `continue_testing`, quando l'ipotesi rimane plausibile ma richiede ulteriori dati;
- `reopen_abduction`, quando l'ipotesi appare debole e sarebbe opportuno ampliare o riaprire lo spazio abduttivo.

Nella versione base, queste raccomandazioni sono registrate nell'output, ma non attivano automaticamente l'acquisizione di nuovi dati o la riapertura dell'abduzione. Dopo l'induzione il grafo esegue il ranking finale e termina.

L'output principale è un oggetto `InductionResult` contenente la valutazione completa di tutte le ipotesi. Parallelamente, il codice costruisce una struttura filtrata contenente esclusivamente le ipotesi con valutazione `plausible`. Questa seconda rappresentazione non sostituisce il risultato completo, ma costituisce una vista derivata utile per eventuali elaborazioni successive.

3.3 Output strutturato

Una caratteristica centrale dell'architettura è l'utilizzo di output strutturati, in formato JSON, descritti e validati mediante schemi Pydantic, una libreria Python che ne permette la definizione implementativa. Ogni schema definisce i campi attesi e il tipo di dato ammesso per ognuno (es. una stringa, un intero) [24]. Questo permette ai vari nodi di scambiarsi dati in un formato uniforme.

Questa scelta risponde a diverse esigenze:

- rendere esplicita la struttura dell'informazione prodotta;
- controllare la struttura e i tipi dei dati scambiati;
- facilitare il passaggio dell'output tra nodi diversi;
- rendere gli aggiornamenti serializzabili;
- semplificare il logging e la successiva analisi dei risultati;
- consentire all'interfaccia grafica di costruire componenti visivi a partire da strutture dati uniformi.

Gli schemi principali utilizzati nell'architettura sono illustrati nella Tabella 3.3.1:

Fase	Schema principale	Informazione prodotta
Astrazione	`AbstractionFeaturesList`	Reperti clinici categorizzati
Abduzione	`AbductionHypothesesList`	Ipotesi diagnostiche e reperti di supporto
Ranking	`RankedHypothesesList`	Ordinamento e criteri di priorità
Deduzione	`DeductionPlan`	Conseguenze cliniche attese
Induzione	`InductionResult`	Confronto tra atteso e osservato

Tabella 3.3.1 – Descrizione degli schemi utilizzati nell'architettura.

Gli schemi relativi al ranking, alla deduzione e all'induzione utilizzano enumerazioni dinamiche per limitare i nomi delle diagnosi a quelli effettivamente presenti nell'input. Inoltre, alcuni schemi impongono che siano presenti esattamente 10 ipotesi, coerentemente con l'impostazione sperimentale Top-10 del framework.

La funzione comune `call_llm` riceve, oltre ai messaggi di sistema e utente, lo schema atteso tramite il parametro `response_format`. Il modello viene quindi istruito a produrre un risultato compatibile con la struttura prevista. Questo meccanismo separa la generazione linguistica dalla validazione formale e permette ai nodi successivi di operare su dati omogenei.

È importante sottolineare che l'output strutturato non garantisce automaticamente la correttezza clinica del contenuto. Esso consente di validare la conformità ai campi e ai tipi previsti dallo schema e facilita l'interoperabilità tra i moduli. Inoltre, la conformità dell'output dipende dalle capacità del modello e dell'interfaccia di generazione usata. La validità clinica delle ipotesi e delle relazioni inferenziali deve essere valutata tramite il protocollo sperimentale e l'ispezione dei risultati.

3.4 Limiti dell'architettura base: staticità e assenza del ciclo di retroazione

L'architettura base implementa in modo esplicito il ciclo di selezione e verifica delle ipotesi, ma presenta alcuni limiti rispetto al modello ST completo e al ragionamento clinico interattivo.

Innanzitutto, il caso clinico viene fornito come testo già disponibile e rimane invariato durante l'esecuzione. Il sistema non dispone di una modalità per richiedere e ricevere nuove osservazioni. Il secondo ranking aggiorna quindi l'ordine delle ipotesi sulla base degli esiti induttivi, ma non corrisponde ad un nuovo ciclo di acquisizione delle informazioni.

Un ulteriore limite riguarda la possibilità che la fase abduttiva iniziale non includa una diagnosi successivamente resa rilevante da una nuova informazione. In assenza di una riapertura dell'abduzione, il sistema può soltanto modificare la priorità delle ipotesi già presenti, senza introdurre una diagnosi assente dall'insieme iniziale delle ipotesi. La staticità dei dati e i limiti nella revisione delle ipotesi motivano i tre approcci della Sezione 4.2, che affrontano aspetti diversi dell'integrazione di nuove evidenze:

- Valutazione puntuale dell'impatto di una nuova informazione tramite SCT;
- Aggiornamento induttivo con riordinamento delle ipotesi esistenti;
- Riesecuzione del ciclo completo, compresa la generazione di nuove ipotesi.

L'architettura base rappresenta dunque il punto di partenza necessario per studiare l'integrazione di nuove evidenze cliniche. Essa rende esplicite le fasi epistemologiche fondamentali, ma non dispone ancora della flessibilità necessaria per modellare pienamente il ragionamento diagnostico come processo interattivo e non monotono.

CAPITOLO 4: Ottimizzazione dell'architettura ed integrazioni

4.1 Ottimizzazione del nodo di astrazione

4.1.1 Motivazione e ruolo del nodo

All'interno dell'architettura epistemologica, il nodo di astrazione costituisce il primo stadio del framework, incaricato di trasformare la narrativa clinica grezza in una rappresentazione strutturata di caratteristiche clinicamente rilevanti, organizzate per categorie (anamnesi prossima, storia clinica remota, esame obiettivo, indagini strumentali/laboratoristiche, farmaci, allergie, e determinanti sociali di salute) selezionate per coprire le principali fonti di informazione utilizzate nella valutazione clinica. Poiché nella versione ottimizzata tutte le fasi successive del grafo operativo (abduzione, ranking, deduzione, induzione) operano esclusivamente sull'output di questo nodo, la qualità, la granularità e la coerenza dell'astrazione condizionano l'intera catena inferenziale a valle. La revisione si è articolata in due passaggi: l'introduzione di un esempio di output (*few-shot prompting*) e del campo di stato dell'evidenza, seguita dall'aggiunta di regole per organizzare e sintetizzare i reperti.

4.1.2 Transizione a *few-shot prompting,* vincoli strutturali e il compromesso della *lossy abstraction*

La versione preliminare del nodo impiegava un prompt di sistema puramente discorsivo, organizzato in linee guida estese senza fornire alcun esempio concreto di output atteso. Il campo `explanation` era vincolato a restituire "un estratto puntuale o una formulazione testuale riportata nel caso clinico originale", non una sintesi, e lo schema non prevedeva ancora un meccanismo esplicito per distinguere i reperti osservati da quelli esclusi (*pertinent negatives,* reperti negativi rilevanti) [25].

Nella prima iterazione di ottimizzazione, il prompt è stato ristrutturato adottando tecniche di *few-shot prompting*: oltre ai vincoli di output espliciti, è stato inserito un esempio completo, e lo schema dati è stato esteso con il campo `status` (stato, che può assumere uno dei seguenti valori: presente [“`present`”] quando l'osservazione è documentata chiaramente nel testo; assente [“`absent`”] quando l'assenza è esplicitamente indicata; incerto [“`uncertain`”] quando l'informazione è incompleta e non è possibile stabilirne con certezza la presenza) per catturare esplicitamente le *pertinent negatives* come voci atomiche.

L'ispezione dei log relativi a due casi clinici ha evidenziato differenze nel livello di dettaglio degli output. In particolare, alcune informazioni erano rappresentate come voci separate, mentre elenchi di condizioni pregresse e farmaci erano raccolti in voci aggregate. Queste osservazioni hanno orientato la revisione del prompt e del campo `explanation`.

Ad esempio, utilizzando gemini-3.6-Flash, la versione originale mostra un'iper-aggregazione che compatta informazioni clinicamente distinte in un'unica voce generica. Nove condizioni cliniche differenti vengono fuse in un solo reperto, "*Multiple chronic medical conditions*" (“Molteplici patologie croniche”), e lo stesso accade per la terapia farmacologica, con tutti e 10 i farmaci assunti dal paziente raggruppati in un'unica voce "*Outpatient medication regimen*" (“Terapia farmacologica abituale”). Un ulteriore esempio, relativo a un secondo caso clinico, riguarda risultati di esami medici raccolti in una sola voce: *“Cardiopulmonary imaging and functional assessment showing interstitial*

fibrosis and right heart failure” (“Imaging cardiopolmonare e valutazione funzionale che mostrano fibrosi interstiziale e scompenso cardiaco destro”).

È importante circoscrivere la portata di questa osservazione: si tratta di un'ispezione qualitativa condotta su una coppia di casi clinici con un modello, non di una valutazione sistematica su un campione ampio, e non è quindi possibile stimarne la frequenza attesa su casi o modelli non osservati. Questa osservazione ha comunque motivato un intervento correttivo diretto sullo schema e sul prompt del nodo, descritto nella sezione 4.1.3.

La trasformazione del testo clinico in una rappresentazione sintetica può comportare la perdita di dettagli o relazioni presenti nella narrazione originale. In questa tesi, il termine *lossy abstraction* [26] indica questo rischio di perdita informativa. La versione ibrida cerca di limitare questo rischio mantenendo distinti i reperti e sintetizzando le informazioni riferite allo stesso elemento.

4.1.3 Consolidamento semantico dello schema dati (soluzione ibrida)

Per rispondere ai fenomeni osservati in 4.1.2, è stata sviluppata una seconda iterazione del nodo, che interviene su due livelli congiunti anziché alternativi. A livello di prompt del sistema, ovvero l’insieme delle istruzioni fornite al sistema dove vengono definiti il compito da svolgere, il ruolo assegnato al modello, le regole di generazione e, quando necessario, il formato dell’output atteso, è stata introdotta una sezione dedicata "CRITICAL" con regole esplicite di deduplicazione: non creare voci diverse per espressioni che indicano lo stesso concetto clinico e riunire invece tutte le informazioni riferite allo stesso reperto in un’unica voce, riportandone anche l’evoluzione nel tempo. A livello di schema dati, il campo `explanation` è stato riformulato da "estratto specifico" a "descrizione clinica sintetizzata" che fonde tutte le menzioni testuali dello stesso concetto in una spiegazione coerente unica, preservando decorso temporale, gravità e valori quantitativi.

Le nuove istruzioni richiedono di mantenere separati i reperti distinti e di riunire, nella spiegazione di ciascuno, le informazioni che lo riguardano. La sintesi non deve quindi trasformarsi in un raggruppamento di condizioni o farmaci diversi.

Per verificare se la soluzione ibrida risolve concretamente le problematiche descritte in 4.1.2, la versione è stata applicata agli stessi casi esaminati precedentemente, a parità di ogni altra condizione. Questo confronto qualitativo permette di isolare l'effetto della modifica del prompt dalla variabilità dovuta alla complessità del caso.

Fenomeno osservato	Versione originale		Versione ibrida
Aggregazione clinica (Gemini-3.6-Flash)	Condizioni distinte collassate in un'unica voce generica.	“Multiple chronic medical conditions”: [...] history of diabetes mellitus, hypertension, hyperlipidemia, cerebrovascular disease"	Condizioni distinte rappresentate in voci separate.
Aggregazione farmacologica (Gemini-3.6-Flash)	Tutti e dieci i farmaci raggruppati in un'unica voce.	“Outpatient medication regimen”: [...] atorvastatina, furosemide, tolazamide, nifedipina, cimetidina."	Dieci voci distinte, una per farmaco.
Aggregazione di risultati di accertamenti differenti (Gemini-3.6-Flash)	Risultati di tre accertamenti differenti aggregati in una voce unica.	"Cardiopulmonary imaging and functional assessment showing interstitial fibrosis and right heart failure"	Tre voci distinte, una per esame.

Tabella 4.1 – Confronto qualitativo di estratti dei log di esecuzione. Versione originale e versione ibrida.

Nei casi esaminati nella Tabella 4.1, la versione modificata rappresenta separatamente condizioni pregresse, farmaci ed esami clinici che nell’output originale erano raccolti in voci aggregate. Il confronto documenta una diversa organizzazione delle informazioni, ma non costituisce una valutazione sistematica della completezza dell’estrazione o dell’accuratezza diagnostica.

4.1.4 Disaccoppiamento dello stato del grafo e gestione del carico informativo

Dal punto di vista architetturale, lo stato condiviso `EpistState` del grafo LangGraph è stato strutturato in modo che il testo clinico venga reso disponibile esclusivamente al nodo di astrazione; i nodi successivi (abduzione non focalizzata, abduzione focalizzata, ranking, deduzione, induzione) operano unicamente sulla rappresentazione astratta, senza accesso diretto alla narrativa grezza.

Questa scelta architetturale risponde a due esigenze complementari: da un lato, la coerenza con l'impianto epistemologico dell'ST-Model, in cui l'astrazione ha la funzione di produrre una rappresentazione compatta e di livello superiore che sostituisce l'osservazione grezza nelle fasi successive di inferenza; dall'altro, il controllo del carico informativo (*prompt load*) lungo l’architettura, un fattore che l'articolo di riferimento [15] identifica come potenzialmente critico per le prestazioni dei modelli di dimensioni minori.

La qualità dell'astrazione agisce come collo di bottiglia informativo per l'intera catena inferenziale a valle, dove omissioni riscontrate nell'astrazione non sono compensabili dai nodi successivi, privi di

accesso al testo originale. Questo aspetto giustifica la necessità di un ulteriore passo di aggiornamento: un ciclo di retroazione incaricato di recuperare eventuali informazioni omesse, dopo l'induzione, oggetto dello sviluppo descritto nella Sezione 4.2.

4.2 Integrazione delle nuove evidenze cliniche

4.2.1 Definizione del *benchmark* e pre-processing dei dati

Il ragionamento clinico è un processo iterativo, in cui l'inquadramento diagnostico si evolve costantemente con l'acquisizione di nuovi referti, esiti strumentali o variazioni sintomatologiche. Per strutturare un ambiente di test capace di valutare quantitativamente e oggettivamente la logica di assimilazione di nuove evidenze da parte degli LLM, la metodologia di riferimento adottata è stata lo *Script Concordance Test* (SCT), descritta nel Capitolo 2.

La preparazione dell'ambiente di validazione ha richiesto un'operazione di pre-processing del dataset originale, descritto nella sezione 2.3.2, per garantirne la coerenza epistemologica:

- Filtraggio delle ipotesi: è stato applicato un filtro basato su espressioni associate a decisioni terapeutiche o gestionali, come *ordering*, *prescribing*, *treating* (richiesta, prescrizione, trattamento) per separare le ipotesi puramente diagnostiche dalle decisioni gestionali o terapeutiche. I record relativi a procedure strumentali o terapie empiriche sono stati esclusi, isolando un set di dati composto unicamente da entità cliniche;
- Normalizzazione dei punteggi: per assegnare un punteggio alle risposte del modello, le frequenze delle risposte fornite dagli esperti sono state normalizzate. Per ciascuna opzione della scala Likert, il punteggio è stato calcolato dividendo il numero di esperti che hanno scelto questa opzione per il numero di esperti che hanno scelto l'opzione più frequente (moda) [14]. Tramite questa normalizzazione, la scelta predominante nel panel ottiene credito pieno pari a 1, mentre le opzioni minoritarie ricevono valori decimali proporzionali.

Il sottoinsieme utilizzato nella valutazione comprende 42 quesiti.

4.2.2 Approccio I: valutazione del punteggio SCT (*zero-shot vs. framework-aware*)

La prima fase dello sviluppo si è concentrata sulla progettazione dei prompt e dell'architettura per la misurazione puntuale dell'impatto della nuova evidenza. Sono state confrontate due modalità differenti, per valutare l'effetto delle informazioni prodotte dal framework sul punteggio assegnato alla nuova evidenza.

Paradigma *zero-shot.* Considerato come riferimento, il modello opera come sistema a scatola nera: il prompt fornisce esclusivamente il quadro clinico, l'ipotesi diagnostica in esame e la nuova informazione, richiedendo all'LLM di restituire un punteggio da -2 a +2 con una breve spiegazione clinica. La logica sottostante fa affidamento esclusivamente sulle caratteristiche interne del modello.

Paradigma *framework-aware*. Questo approccio inserisce nel contesto dell'LLM le elaborazioni precedentemente generate dal framework: le aspettative cliniche dedotte (con i relativi gradi di priorità) e lo stato induttivo iniziale, entrambi estratti in modo mirato per la sola ipotesi in esame.

L'istruzione di sistema vincola esplicitamente il modello a giustificare l'assegnazione del punteggio basandosi non sull'intuito generale, ma sul modo in cui la nuova informazione conferma o smentisce le specifiche aspettative deduttive estratte al passo precedente.

Per applicare il paradigma *framework-aware* a tutti i dati SCT, l'ipotesi diagnostica prevista dal dataset viene aggiunta di base all'insieme delle ipotesi prima della fase deduttiva qualora non dovesse essere generata spontaneamente dall'architettura. In questo modo, il sistema può comunque produrre le conseguenze cliniche attese e valutarle rispetto alla nuova informazione, anche quando l'ipotesi non compariva inizialmente tra quelle considerate dal modello. La valutazione misura quindi l'aggiornamento del giudizio su un'ipotesi fornita dal quesito, non la capacità del sistema di generarla spontaneamente.

4.2.3 Approccio II: aggiornamento induttivo e riordinamento delle ipotesi

L'Approccio II è stato sviluppato con l'obiettivo di aggiornare la gerarchia delle ipotesi diagnostiche sulla base della nuova informazione clinica, evitando il costo computazionale di una rigenerazione totale del grafo di ragionamento.

Lo spazio delle ipotesi rimane invariato rispetto al ciclo di ragionamento iniziale: la nuova informazione viene integrata nell'astrazione clinica di partenza e, anziché rieseguire abduzione e deduzione da zero, il framework riutilizza il piano deduttivo generato nel primo ciclo, e rilancia esclusivamente la fase induttiva. Il ranking finale viene quindi ricalcolato applicando ai nuovi esiti induttivi gli stessi quattro criteri clinici del framework (parsimonia, pericolosità, costo, curabilità).

Questa configurazione richiede meno passaggi rispetto alla riesecuzione completa, poiché riutilizza ipotesi e aspettative già prodotte. La sua struttura logica, tuttavia, la espone a un effetto di ancoraggio (*Anchoring Bias*) [27]: poiché lo spazio delle ipotesi resta vincolato a quello generato nel ciclo iniziale, l'approccio non è strutturalmente in grado di introdurre nuove entità che non fossero già state considerate nella prima fase abduttiva, anche qualora la nuova informazione clinica ne suggerisse fortemente la rilevanza.

4.2.4 Approccio III: riesecuzione completa del ciclo cognitivo

Per superare i limiti strutturali del riordinamento statico delle ipotesi, in particolare l'effetto di ancoraggio descritto in 4.2.3, l'Approccio III è stato progettato come un secondo ciclo completo di ragionamento, pensato per essere attivato in presenza di evidenze in grado di stravolgere l'orientamento clinico iniziale.

La logica implementativa prevede l'esecuzione di un secondo ciclo completo del framework epistemologico:

- Nuova astrazione: la nuova informazione clinica viene fusa con il quadro anamnestico originale, generando un'astrazione aggiornata;
- Abduzione rigenerata: sulla base del nuovo quadro astratto, il sistema esegue nuovamente l'interrogazione abduttiva, sia focalizzata che non focalizzata, passando come contesto anche

le ipotesi generate nel primo ciclo, per informare (senza vincolare rigidamente) la nuova generazione. Questa fase è cruciale poiché consente di proporre ipotesi assenti dal primo ciclo;
- Nuova deduzione e induzione: le nuove ipotesi passano attraverso una nuova scomposizione deduttiva per estrarne le aspettative attese, culminando in una fase induttiva finale che produce una classifica diagnostica revisionata.

Questo approccio (III) non si limita quindi a riordinare le ipotesi già presenti, ma ripete il ciclo di ragionamento a partire dal quadro clinico aggiornato. Rispetto all'Approccio II, questa maggiore flessibilità richiede però un numero più elevato di chiamate al modello linguistico.

4.3 Progettazione e sviluppo dell'interfaccia grafica

4.3.1 Analisi dei requisiti e scelte tecnologiche

L'obiettivo primario della progettazione di un'interfaccia grafica è minimizzare il carico informativo dell'operatore, rendendo ispezionabili gli output e i passaggi del processo. Durante la fase di analisi del progetto, sono stati individuati i seguenti requisiti fondamentali:

- Ispezionabilità del processo (e*xplainability*): necessità di poter ispezionare visivamente i dati clinici grezzi e le motivazioni logiche alla base di ogni classificazione generata dal sistema;
- Prevenzione degli errori operativi: implementazione di meccanismi di validazione (come finestre modali di blocco) per prevenire alterazioni accidentali delle ipotesi diagnostiche o l'immissione di dati incongruenti;
- Interazione non bloccante: l'applicativo deve mantenere una reattività costante dell'interfaccia utente (UI) anche durante i tempi di attesa legati ai calcoli di inferenza dei modelli linguistici di grandi dimensioni (LLM).

Per soddisfare questi requisiti, mantenendo una stretta coesione con il motore computazionale sviluppato in Python (incluso il framework a grafo e l'infrastruttura di inferenza locale dei modelli linguistici), la scelta è ricaduta su Dash [28]. Questo framework open-source consente di costruire applicazioni web analitiche utilizzando esclusivamente Python.

Per organizzare in modo uniforme gli elementi dell'interfaccia, l'ecosistema nativo è stato integrato con la libreria *Dash Bootstrap Components* (DBC). I vantaggi derivanti da questa integrazione includono:

- Layout reattivi: sfruttamento del sistema a griglia (*Row, Col*) per gestire dinamicamente lo spazio di lavoro, ottimizzando la densità informativa;
- Componentistica semantica: adozione di elementi quali *card* per compartimentalizzare le diagnosi, *accordion* per isolare i log di sistema senza saturare lo schermo, e *badge* semantici per evidenziare visivamente le variazioni di posizione in classifica e le priorità cliniche.

4.3.2 Flusso dati multi-pagina e gestione dei processi (*threading e polling*)

Per garantire una rigorosa separazione degli intenti e guidare l'operatore attraverso un percorso operativo logico, l'applicazione è stata pensata secondo un paradigma multi-pagina, sfruttando il sistema di indirizzamento nativo di Dash. Questa architettura separa nettamente il momento dell'acquisizione dei dati da quello dell'analisi computazionale.

Il flusso applicativo ha inizio nella *Home Page* (indirizzo '/'), che funge da modulo di configurazione per l'acquisizione del caso clinico e del codice identificativo associato. L'interfaccia supporta due modalità di immissione: l'inserimento testuale manuale e il caricamento di file strutturati (.csv o .xlsx). Per garantire la robustezza di quest'ultima modalità, è stato sviluppato un sistema di confronto dinamico: il codice ispeziona le intestazioni del file caricato e, qualora non rilevi la nomenclatura standard attesa ("Case-Id" e "Case") o voci simili, interrompe il processo bloccando lo schermo con una finestra che obbliga l'utente a mappare esplicitamente le colonne del proprio dataset verso le variabili di sistema, prevenendo errori di formato a valle.

Prima di avviare l'esecuzione, l'operatore configura inoltre la modalità del controllo umano tramite un selettore denominato "*Enable Step-by-Step Debug Mode*" (Abilita la modalità di revisione passo-passo), il cui funzionamento è descritto in dettaglio nella Sezione 4.3.3.

Una volta completata la configurazione, si procede al trasferimento sicuro delle variabili di stato (identificativo del paziente, testo clinico e modalità operativa) verso la pagina di elaborazione, senza database persistenti esterni o stringhe di query esposte nell'URL. Il flusso dei dati è governato tramite il componente `dcc.Store` configurato come memoria a livello di sessione, secondo il seguente schema: alla pressione del pulsante "*Start Reasoning*", una *callback* di validazione, ovvero una funzione associata ad un oggetto che viene chiamata quando rileva un cambio di stato, verifica l'integrità dei campi obbligatori; in caso di esito positivo, i dati vengono impacchettati in un dizionario e inseriti nella memoria di sessione del browser; un componente `dcc.Location` esegue quindi un reindirizzamento verso la pagina dei risultati "*Results"* (indirizzo '/results'); le funzioni di inizializzazione estraggono il contenuto dalla memoria di sessione, utilizzandolo per innescare il ciclo inferenziale del framework epistemologico. Questa progettazione assicura che lo stato della singola esecuzione sia legato alla sessione web attiva, prevenendo la contaminazione delle variabili tra pazienti diversi o ricaricamenti accidentali della pagina.

Uno dei limiti strutturali intrinseci di Dash è la sua natura nativamente sincrona: l'esecuzione di una *callback*, come la chiamata all'LLM, blocca il *thread* principale fino alla restituzione del risultato. Nel contesto di un sistema basato su LLM, dove i tempi di inferenza possono richiedere da decine di secondi a minuti per singolo nodo computazionale, un approccio sincrono causerebbe il blocco totale dell'interfaccia, compromettendo l'esperienza dell'utente. Per ovviare a questo problema, l'architettura implementa un paradigma di esecuzione asincrona ad eventi:

- Alleggerimento computazionale: all'avvio del caso clinico, l'elaborazione viene delegata a un *thread* secondario in background, che gestisce l'inizializzazione del grafo e le chiamate all'LLM, liberando il *thread* principale per gestire l'interfaccia;
- Memoria condivisa (SHARED_STATE): un dizionario globale di stato funge da ponte di comunicazione tra i *thread*. Ogni volta che l'LLM completa l'esecuzione di un nodo, il *thread*

in background aggiorna i dati condivisi e imposta specifiche variabili booleane (es. `ui_changed["ranking"] = True`) per segnalare la disponibilità di nuove informazioni;

- Aggiornamento periodico dell'interfaccia tramite *polling*: un componente (`dcc.Interval`) scatta ogni 1.5 secondi innescando una *callback* che ispeziona i flag della memoria condivisa; se rileva variazioni, aggiorna selettivamente i componenti della pagina, altrimenti restituisce `dash.no_update` (indice che non sono state registrate modifiche), ottimizzando le risorse.

Infine, per un controllo preciso sui tempi di esecuzione, l'architettura sfrutta la primitiva `threading.Event`, che funge da controllo logico capace di sospendere in modo controllato il *thread* in background che gestisce le chiamate sequenziali all'LLM, meccanismo che costituisce le fondamenta dell'implementazione del controllo umano descritto nella sezione seguente.

4.3.3 Integrazione del meccanismo *Human-in-the-Loop*

Nell'ambito dei CDSS basati su intelligenza artificiale, l'esclusione dell'esperto umano dal processo inferenziale pone criticità etico-legali e di affidabilità diagnostica. Per questo motivo, il framework integra un meccanismo Human-in-the-Loop (HITL) configurabile che consente all'operatore di ispezionare e modificare il ranking prima che il grafo prosegua. Questa funzionalità è stata verificata sul piano implementativo, ma non valutata in uno studio con clinici.

A differenza di un semplice interruttore binario che abilita o disabilita la sospensione dell'algoritmo, il controllo "*Enable Step-by-Step Debug Mode*" impostato dall'operatore nella Home Page, come già anticipato, seleziona la granularità con cui il sistema si arresta per la revisione umana, offrendo due modalità operative:

- Modalità *Step-by-Step*: il *thread* in background invoca il metodo `wait()` al termine dell'esecuzione di ogni nodo. Questa modalità è pensata per un uso di debug e ispezione granulare dell'intero ciclo di ragionamento;
- Modalità *Ranking-Only*: la sospensione avviene esclusivamente in corrispondenza dei nodi di ranking (*First Ranking e Final Ranking*), lasciando che le fasi intermedie (abduzione, deduzione, induzione) procedano senza interruzione. Questa modalità riflette l'impianto epistemologico del sistema, concentrando la supervisione umana nei punti di maggiore impatto decisionale piuttosto che a ogni singolo passaggio computazionale.

In entrambe le modalità, la sospensione innesca un aggiornamento dell'interfaccia (tramite il meccanismo di *polling* descritto in 4.3.2), abilitando la visualizzazione del pannello di interazione e del pulsante "*Next Step*". In corrispondenza del *First Ranking*, indipendentemente dalla modalità selezionata, all'operatore viene presentata la lista delle ipotesi diagnostiche generate; agendo su un apposito interruttore, l'utente sblocca funzionalità di modifica che permettono di:

- riordinare le priorità, tramite pulsanti direzionali che correggono l'ordine clinico delle patologie;
- eliminare le ipotesi ritenute inappropriate dall'operatore, tramite un comando di cancellazione protetto da una finestra di conferma.

Ogni iterazione dell'utente (spostamento o cancellazione) scatena una *callback* che aggiorna e riordina in tempo reale le strutture dati Pydantic serializzate nel dizionario condiviso `SHARED_STATE`. Alla pressione di "*Next Step*" dopo il *First Ranking*, l'interfaccia di editing viene bloccata e contemporaneamente avviene il risveglio del *thread* in background, che preleva il ranking modificato o confermato dall'operatore e lo inserisce nei nodi successivi del grafo (Deduzione e Induzione), assicurando che il ragionamento a valle sia fondato su presupposti approvati.

4.3.4 Rappresentazione strutturata dei dati e tracciabilità

L'adozione di modelli generativi in ambito clinico porta con sé il rischio intrinseco dell'effetto "scatola nera", dove l'algoritmo fornisce una risposta senza palesare il percorso logico sottostante. Per mitigare questo fenomeno, l'interfaccia è stata concepita attorno ai principi della tracciabilità algoritmica, strutturando gli output testuali in componenti visivi ad alta densità informativa.

La presentazione delle ipotesi diagnostiche avviene tramite oggetti espandibili: ogni componente racchiude nome della patologia e rango, e permette la valutazione delle quattro caratteristiche associate, Parsimonia, Pericolosità, Costo, Curabilità, ciascuna con un colore dedicato (es. rosso per la Pericolosità, verde per la Curabilità) per favorire una rapida identificazione del rischio.

Il sistema di *Delta-Badge* (indicatori di variazione) mostra l'aggiornamento delle variazioni di posizione in classifica indotte dall'utente, confrontando lo stato corrente con la lista prodotta dall'LLM salvata in memoria. Il sistema distingue due casistiche: modifiche manuali (indicatore giallo con icona a forma di matita, quando il clinico altera il *First Ranking*) e aggiornamento post-induzione (indicatori cromatici verde/rosso/grigio per ascesa, discesa o stabilità per marcare le differenze tra il ranking preliminare e quello finale), una distinzione che permette di attribuire con precisione la fonte di ogni variazione di ranking.

Infine, per garantire la tracciabilità completa, è stata implementata una sezione "*Reasoning Traceability*" nella parte inferiore della pagina dei risultati, strutturata a pannelli espandibili (a*ccordion*) che contengono i log testuali delle fasi di Astrazione, Abduzione e Induzione in aree di scorrimento stilizzate come blocchi di codice, senza saturare visivamente l'area di lavoro principale.

4.3.5 Gestione dello stato globale, persistenza e logging strutturato

In un CDSS, dove il processo inferenziale si sviluppa su orizzonti temporali prolungati e l'operatore altera attivamente il ranking delle ipotesi, la cattura e la tracciabilità dello stato di esecuzione diventano requisiti imprescindibili.

L'architettura fa affidamento sul già descritto dizionario di stato globale SHARED_STATE, che assolve una duplice funzione: mezzo di comunicazione tra i *thread*, e memoria temporanea per l'output grezzo dei nodi LangGraph e le riassegnazioni di priorità del clinico.

Per garantire persistenza e tracciabilità, l'interfaccia è dotata di un modulo di salvataggio strutturato su richiesta dell'operatore (pulsante "*Save Progress*"), che memorizza l'intero stato corrente in un file JSON.

Dal punto di vista implementativo, la generazione del file ha richiesto la risoluzione di criticità legate alla serializzazione di strutture dati complesse, quali le classi Pydantic e le enumerazioni (Enum) intrinseche al motore LangGraph. Tale ostacolo è stato superato tramite un codificatore JSON personalizzato (`EpistStateEncoder`), programmato per intercettare e tradurre dinamicamente tali oggetti in un formato di testo compatibile. Il contenuto generato incapsula:

- Metadati di sessione: identificativo del paziente, data e ora del salvataggio, nodo logico corrente, stato del flag operativo manuale;
- Dati grezzi di inferenza: l'aggregato completo degli output generati da ciascun nodo del grafo fino al momento del salvataggio;
- Memoria dell'interfaccia utente: la classifica originaria calcolata dal modello generativo, affiancata al ranking eventualmente modificato dall'utente.

A livello di archiviazione, il modulo crea dinamicamente cartelle dedicate per ciascun paziente e archivia i risultati in file nominati univocamente. Questa infrastruttura non si limita a garantire la tracciabilità dell'inferenza, ma pone le fondamenta per future funzionalità di recupero, che consentiranno di ricaricare un file JSON per riprendere l'iter diagnostico da uno stato pregresso.

CAPITOLO 5: Valutazione sperimentale e risultati

5.1 Setup sperimentale e protocollo di valutazione

La valutazione sperimentale del framework è stata condotta secondo un protocollo comune, descritto in questa sezione, che definisce l'ambiente di esecuzione, i criteri di selezione dei modelli linguistici impiegati e l'insieme delle metriche utilizzate per interpretare i risultati. Le sottosezioni sperimentali successive condividono in larga parte lo stesso impianto metodologico pur applicandolo a compiti differenti: la valutazione del nodo di astrazione (Sezione 5.2), la valutazione dell'integrazione di nuove evidenze cliniche tramite gli Approcci I, II e III (Sezione 5.3) e la dimostrazione funzionale dell'interfaccia grafica (Sezione 5.4).

5.1.1 Configurazione dell'ambiente di esecuzione e selezione degli LLM

Il framework è stato eseguito nell'ambiente Python descritto nel Capitolo 3, con orchestrazione del grafo di ragionamento tramite LangGraph e validazione strutturale degli output di ciascun nodo tramite schemi Pydantic, secondo l'architettura illustrata in Sezione 3.3. Tutte le chiamate ai modelli linguistici sono state gestite attraverso la funzione comune di interrogazione, che vincola la risposta del modello allo schema dati previsto per il nodo corrente.

I casi clinici utilizzati come test provengono da due fonti già introdotte nel Capitolo 2: un sottoinsieme di 10 casi selezionati dal *benchmark* CPC-Bench, e il dataset pre-processato SCT, impiegato nella valutazione dell'Approccio I. È opportuno segnalare un limite metodologico trasversale a entrambe le fonti: i casi CPC-Bench sono estratti da casi clinici del NEJM disponibili pubblicamente, e non è possibile escludere che alcuni di essi, o presentazioni clinicamente molto simili, siano stati incontrati dai modelli durante la fase di addestramento. Questo rischio di contaminazione, comune a gran parte della letteratura che utilizza fonti cliniche pubbliche per la valutazione di LLM, costituisce una delle motivazioni alla base della proposta di validazione prospettica discussa nel Capitolo 6.

Per quanto riguarda la selezione dei modelli linguistici, il presente lavoro adotta una strategia a più livelli, motivata sia da considerazioni di rappresentatività sia da vincoli pratici.

Modello primario di validazione. Gemini 2.5 Pro è stato impiegato come modello di riferimento per tutte le valutazioni che richiedono test ripetuti e confronti statistici appaiati (Sezioni 5.2, 5.3.1 e 5.3.2), per la sua disponibilità tramite API *(Application Programming Interface*, interfacce attraverso cui l'applicazione invia richieste e riceve risposte [29]*),* le ottime prestazioni in termini di capacità di ragionamento e la possibilità di eseguire un numero di test sufficiente (5 o 10 per caso, secondo l'esperimento) a stimare la variabilità intra-caso del modello.

Modello secondario di replica. Gemini 2.5 Flash, un modello della stessa famiglia, è stato impiegato in Sezione 5.3.1 per verificare se gli effetti osservati con il modello primario si riproducessero su un modello diverso, permettendo di distinguere proprietà dell'architettura da comportamenti specifici di un singolo modello.

Modello a utilizzo limitato. Gemini-3.6-flash è stato impiegato esclusivamente per la valutazione preliminare qualitativa del nodo di astrazione, descritta nelle Sezioni 4.1.2 e 4.1.3, e non compare

nelle valutazioni del resto del Capitolo 5. La scelta di non utilizzare questo modello per le valutazioni finali è motivata dalla natura ancora preliminare al momento dello svolgimento del lavoro: la famiglia Gemini 3 è stata annunciata da Google il 18 novembre 2025 [30] e le sue varianti, inclusa quella utilizzata in questa tesi, sono rimaste per diversi mesi in stato *preview* (anteprima), con cicli di aggiornamento e deprecazione più rapidi rispetto ai modelli della serie precedente. Gemini 2.5 Pro e Gemini 2.5 Flash, invece, hanno raggiunto lo stato di disponibilità generale il 17 giugno 2025 [31], senza data di ritiro annunciata al momento della stesura di questo lavoro.

Per tutti i modelli, i parametri di generazione non sono stati oggetto di ottimizzazione sistematica e sono stati mantenuti alle impostazioni predefinite offerte dalle rispettive API, secondo una scelta che ha come obiettivo quello di catturare, tra le altre cose, le caratteristiche stocastiche dei modelli.

5.1.2 Metriche di valutazione

Questa sezione raccoglie un insieme di metriche utilizzate, distinguendo tre famiglie: metriche di accuratezza diagnostica, riportate nella Tabella 5.1.1, metriche di concordanza SCT, riportate nella Tabella 5.1.2, e metriche di stabilità/variabilità, riportate nella Tabella 5.1.3, oltre ai test statistici impiegati per il confronto tra condizioni sperimentali.

Metrica	Definizione operativa
Accuratezza Top-*k*	Proporzione di test in cui una diagnosi di riferimento, o un suo alias, corrisponde a una diagnosi predetta, secondo il criterio di similarità lessicale descritto di seguito, entro le prime k posizioni del ranking differenziale.
Miss rate	Complemento dell'accuratezza (Top-10); proporzione di test in cui la diagnosi corretta non compare in alcuna posizione del differenziale considerato.
Mean Reciprocal Rank (MRR)	Media, sui test effettuati, del reciproco della posizione della diagnosi corretta nel ranking; assume valore nullo quando la diagnosi non è presente.
Mean rank	Posizione media della diagnosi corretta, calcolata esclusivamente sui test in cui essa risulta presente entro la Top-10, utile per distinguere il caso di omissione dal caso di presenza a rango basso.

Tabella 5.1.1 – Metriche di accuratezza diagnostica. Utilizzate in Sezione 5.2 e, in forma di rango, in Sezione 5.3.2.

La valutazione delle prestazioni diagnostiche dell'architettura è stata implementata attraverso Python, effettuando un confronto di similarità lessicale tra le ipotesi diagnostiche generate (p) e l'insieme di diagnosi testuale e ontologie (UMLS/ICD [19][20]) contenuti nel dataset CPC di riferimento. Dunque, per ogni caso clinico c , la diagnosi di riferimento è stata rappresentata come un insieme di alias $a \in A_c$, costituito dall'integrazione di:

- segmentazione in sotto-entità, in corrispondenza di punti e virgole, della diagnosi testuale riportata nel dataset;
- ontologie disponibili associate a ciascuna diagnosi, a cui sono stati rimossi metadati ed etichette descrittive (es. '{ICD10CM}', '[*Disease or Syndrome*]').

Successivamente, sia le ipotesi p sia l'insieme degli alias A_c sono stati sottoposti ad un pre-processing di normalizzazione, con la rimozione della punteggiatura e *stop word* (termini ad alta frequenza e scarso valore informativo) e la conversione in caratteri minuscoli. Questa scelta riduce il rischio che termini poco discriminativi aumentino la somiglianza tra diagnosi differenti.

Il confronto considera tre possibili forme di corrispondenza. Due diagnosi possono presentare formulazioni quasi identiche e differire soltanto nell'ordine dei termini, oppure essere definite a livelli diversi di specificità (come "*Asbestosis*" all'interno di "*Pulmonary asbestosis*").

Definiti T(a) e T(p) l'insieme dei token informativi ottenuti per i due oggetti di confronto, la similarità è calcolata come massimo tra tre parametri distinti:

$$sim(p,a) = \max\Big(SM(p,a),\ J\big(T(p),T(a)\big),\ 0.7 \cdot C\big(T(p),T(a)\big)\Big)$$

con $sim(p,a) \in [0,1]$.

Il primo termine, $SM(p,a)$, è il rapporto di similarità restituito dal metodo `SequenceMatcher.ratio()` della libreria standard `difflib` di Python [33]. Il metodo confronta due stringhe normalizzate ricercando blocchi di caratteri corrispondenti e consecutivi, secondo una variante dell'algoritmo di Ratcliff e Obershelp [34]. Il rapporto è definito come $2 \cdot \frac{M}{T}$, dove M indica il numero di corrispondenze e T il numero totale di caratteri nelle due stringhe. Il valore è pari a 1 per stringhe identiche e a 0 in assenza di corrispondenza e ricerca formulazioni quasi identiche, differenti per piccole variazioni lessicali.

Il secondo termine è l'indice di Jaccard: $J\big(T(p),T(a)\big) = \frac{|T(p) \cap T(a)|}{|T(p) \cup T(a)|}$ [35]. Misura la proporzione di token condivisi rispetto all'insieme complessivo dei token presenti nei due insiemi, senza considerarne l'ordine. Il valore è pari ad 1 quando i due insiemi coincidono e a 0 quando non condividono alcun token.

Il terzo termine è il coefficiente di sovrapposizione di Szymkiewicz-Simpson [36]: $C\big(T(p),T(a)\big) = \frac{|T(p) \cap T(a)|}{\min(|T(p)|,|T(a)|)}$. A differenza dell'indice di Jaccard, assume valore 1 quando tutti i token dell'insieme più piccolo sono presenti nell'altro. Permette dunque di riconoscere corrispondenze tra formulazioni di diversa specificità. Il suo contributo viene pesato per un coefficiente pari a 0.7, così da non equiparare automaticamente una semplice inclusione lessicale a una corrispondenza perfetta.

Una predizione p è considerata corretta dal criterio adottato se almeno uno degli alias supera la soglia: $sim(p, a) \geq 0.65$. Il vincolo della soglia pari a 0.65 è stata una scelta progettuale, motivata da diverse ragioni:

- All'interno della documentazione della libreria `SequenceMatcher`, viene indicata come regola pratica la scelta di una soglia minima di 0.6 [33]. Il valore 0.65 rende il criterio leggermente più restrittivo;
- Il fattore 0.7 associato al coefficiente Szymkiewicz-Simpson (C) fa sì che una sovrapposizione incompleta non superi da sola la soglia: per esempio, $C = 0.9$ produce un contributo pari a 0.63, inferiore alla soglia definita.

La posizione risultante in classifica viene associata alla prima ipotesi che supera la soglia di similarità; in caso di mancata corrispondenza verrà segnata un'omissione. È stato adottato un confronto lessicale, anziché basato su modelli di *embedding*, reti neurali che traducono il testo in vettori numerici per il calcolo matematico della similarità, per mantenere il criterio di valutazione deterministico e verificabile: ogni punteggio deriva da risultati esplicitamente ispezionabili. I modelli di *embedding* possono catturare relazioni semantiche che il confronto lessicale non riconosce, ma la decisione della corrispondenza dipende dal modello, dall'addestramento e dalla rappresentazione vettoriale utilizzata [37].

Metrica	Definizione operativa
Concordanza (punteggio SCT)	Punteggio normalizzato attribuito a ciascuna risposta del modello, calcolato rapportando la frequenza dell'opzione scelta nel panel alla frequenza dell'opzione modale degli esperti.
Accordo con la moda degli esperti	Variabile binaria che indica se la scelta modale del modello sulle esecuzioni ripetute coincide con la scelta modale del panel.

Tabella 5.1.2 – Metriche di concordanza SCT. Utilizzate in Sezione 5.3.1.

Metrica	Definizione operativa
Deviazione standard intra-caso	Dispersione del punteggio (concordanza o rango) tra test ripetuti sullo stesso caso clinico, a parità di condizione sperimentale.
Self-consistency	Percentuale di osservazioni, sui test ripetuti per il medesimo caso, che riproducono la scelta modale del modello.
Delta di rango (Δ)	Nella valutazione degli Approcci II e III (Sezione 5.3.2), differenza tra la posizione in classifica della diagnosi corretta prima e dopo l'inserimento dell'evidenza e la sua elaborazione. Un valore positivo indica un miglioramento della posizione.
Tasso di recupero	Proporzione di test in cui una diagnosi inizialmente assente dal ranking (omissione) risulta presente nel ranking finale dopo l'elaborazione di nuova evidenza.

Tabella 5.1.3 – Metriche di stabilità e variabilità. Utilizzate in Sezione 5.3.1 e 5.3.2.

Per il confronto di punteggi continui appaiati tra due condizioni sperimentali sullo stesso insieme di casi (ad esempio, *zero-shot* contro *framework-aware* in Sezione 5.3.1) è stato impiegato il test non parametrico dei ranghi con segno di Wilcoxon, scelto in sostituzione del t-test appaiato per l'assenza di un'assunzione di normalità sulla distribuzione delle differenze, assunzione difficilmente verificabile con la numerosità limitata dei campioni disponibili in questo lavoro.

Per il confronto di variabili binarie appaiate (ad esempio, accordo/disaccordo con la moda degli esperti prima e dopo un intervento) è stato impiegato un test esatto sulle coppie discordanti, equivalente a un test di McNemar in condizioni di campione piccolo, tramite distribuzione binomiale esatta.

Per il confronto della varianza tra due gruppi indipendenti di osservazioni (ad esempio, la variabilità del delta di rango tra Approccio II e Approccio III in Sezione 5.3.2) è stato impiegato il test di Levene, robusto a deviazioni dalla normalità.

In tutti i casi, data la numerosità limitata dei campioni disponibili, un numero di casi clinici compreso tra 4 e 10 a seconda dell'esperimento, i risultati dei test statistici sono stati interpretati come indicazioni di tendenza e non come prove conclusive di superiorità di una condizione sull'altra, coerentemente con l'approccio interpretativo adottato nelle Sezioni 5.2 e 5.3.

5.2 Valutazione del nodo di astrazione

L’analisi sperimentale del nodo di astrazione ha l'obiettivo di quantificare l'effetto della transizione dalla formulazione preliminare, basata su un prompt discorsivo e privo di vincoli di deduplicazione, alla formulazione ottimizzata ibrida, descritta nella Sezione 4.1.3. L'analisi non si limita a valutare l'output diretto del nodo, ma verifica in che misura la diversa rappresentazione del caso clinico influenzi la catena inferenziale a valle, abduzione, ranking, deduzione e induzione, attraverso due dimensioni: le prestazioni diagnostiche del framework completo e il profilo computazionale dell'output prodotto dal nodo.

5.2.1 Confronto delle prestazioni diagnostiche

Le sperimentazioni hanno coinvolto 10 casi clinici complessi selezionati dal *benchmark* CPC-Bench [17]. Per ciascun caso e per ciascuna configurazione del nodo di astrazione sono stati eseguiti 10 test indipendenti dell'intero framework mediante il modello Gemini 2.5 Pro, per un totale di 100 esecuzioni per condizione. Il confronto tra le diagnosi prodotte e la diagnosi di riferimento è stato effettuato con il criterio di similarità lessicale ibrida descritto nella sezione 5.1.2. Le metriche sono calcolate sull’insieme delle 100 esecuzioni per condizione.

La Tabella 5.2.1 riporta le metriche aggregate ottenute dalle due configurazioni del nodo di astrazione.

Metrica	Astrazione originale	Astrazione ottimizzata	Variazione assoluta
Accuratezza Top-1	47.0%	36.0%	−11.0 p.p.
Accuratezza Top-3	62.0%	54.0%	−8.0 p.p.
Accuratezza Top-5	65.0%	63.0%	−2.0 p.p.
Accuratezza Top-10	72.0%	74.0%	+2.0 p.p.
MRR	0.552	0.476	−0.076
Miss rate	28.0%	26.0%	−2.0 p.p.
Mean rank*	2.06	2.65	+0.59

Tabella 5.2.1 – Confronto delle prestazioni diagnostiche del framework delle due configurazioni del nodo di astrazione**.** *Posizione media (*mean rank*) calcolata esclusivamente sui test in cui la diagnosi corretta risulta presente entro le prime 10 posizioni del ranking.

I dati evidenziano un comportamento differenziale netto tra le due versioni. Con la configurazione ottimizzata, la probabilità che il sistema collochi la diagnosi corretta esattamente al primo rango scende dal 47.0% al 36.0%, una variazione di 11 punti percentuali; la Top-3 si abbassa dal 62.0% al 54.0% e l'MRR diminuisce da 0.552 a 0.476. Coerentemente, quando la diagnosi corretta viene identificata, essa occupa in media una posizione più bassa nel ranking, con il rango medio che passa da 2.06 a 2.65.

Nel campione analizzato, la configurazione modificata colloca meno frequentemente la diagnosi di riferimento nelle prime posizioni, mentre la sua presenza entro la Top-10 varia dal 72% al 74%. Quest'ultima differenza corrisponde a due esecuzioni aggiuntive con corrispondenza diagnostica e non dimostra un miglioramento generale.

5.2.2 Analisi del carico informativo

Oltre alle prestazioni diagnostiche, è stato misurato il volume testuale dell'output prodotto dal nodo di astrazione sulle 100 esecuzioni analizzate, poiché tale output costituisce l'unico input disponibile alle fasi successive del grafo, secondo il disaccoppiamento descritto nella Sezione 4.1.4.

Metrica	Astrazione originale	Astrazione ottimizzata	Riduzione
Numero medio di token stimati	2451	1708	30.3%
Numero medio parole	1491.3	973.8	34.7%
Numero medio caratteri (spazi inclusi)	9804.5	6832.7	30.3%
Numero medio caratteri (spazi esclusi)	8314.2	5492.1	33.9%

Tabella 5.2.2 – Confronto del volume testuale medio dell'output del nodo di astrazione, calcolato sulle 100 esecuzioni analizzate relative ai 10 casi clinici.

La configurazione originale produce un output di 9804.5 caratteri e 1491.3 parole in media per caso clinico, corrispondenti a circa 2451 token. La configurazione ottimizzata riduce l'output a 6832.7 caratteri e 973.8 parole, pari a circa 1708 token. I token sono stimati dividendo il numero di caratteri per 4, un'approssimazione valida per testo in lingua inglese [38]; la stima non è un conteggio effettuato con il *tokenizer* del modello utilizzato, ovvero il contatore effettivo dei token, e i valori

vanno quindi considerati come indicatore orientativo. La riduzione relativa, pari al 30.3% in caratteri e al 34.7% in parole propagate ai nodi successivi del grafo, non dipende dalla stima.

La deviazione standard del numero di parole diminuisce da 285.8 a 196.9. La riduzione della dispersione assoluta si accompagna alla diminuzione della lunghezza media e non dimostra, da sola, una maggiore uniformità relativa degli output.

È importante sottolineare che la diminuzione del volume testuale non corrisponde, secondo quanto discusso nella Sezione 4.1.3, a una riduzione del numero di entità cliniche estratte: l'introduzione del campo di stato dell'evidenza e delle regole di deduplicazione semantica ha permesso di rappresentare un numero maggiore di reperti clinici, inclusa la scomposizione delle *pertinent negatives* precedentemente aggregate in un'unica stringa narrativa, utilizzando un numero inferiore di caratteri complessivi. La Tabella 5.2.3 confronta un estratto dell'output di astrazione registrato nel file log della versione originale e di quella ottimizzata, mostrando le differenze nella struttura e nell'organizzazione delle informazioni ricavate.

ABSTRACTION — The features abstracted by the text are:	ABSTRACTION ========== CATEGORY: HISTORY OF PRESENT ILLNESS [PRESENT] DYSPNEA ↳ Progressive shortness of breath starting 6 months before admission, with exercise capacity gradually decreasing from walking three blocks to climbing one flight of stairs. Acutely worsened 3 days before admission, causing difficulty walking two to three steps. On arrival of EMS, breathing was labored. Dyspnea on exertion persisted throughout the hospital stay, with recurrent episodes of desaturation with minimal activity despite oxygen therapy. [PRESENT] ALTERED MENTAL STATUS ↳ For 2 days prior to admission, the patient was noted by his children to be somnolent and less interactive than usual. [PRESENT] DECREASED APPETITE ↳ Noted by his children for 2 days prior to admission. [PRESENT] DECREASED URINE OUTPUT ↳ Noted by his children for 2 days prior to admission. [PRESENT] UNSTEADY GAIT ↳ Noted by his children for 2 days prior to admission. [PRESENT] WEIGHT LOSS ↳ Lost approximately 2.3 kg within the 2 months prior to admission. [PRESENT] COUGH [ABSENT] FEVER ↳ The patient did not have fevers. [ABSENT] CHILLS ↳ The patient did not have chills. [ABSENT] NIGHT SWEATS ↳ The patient did not have night sweats. [ABSENT] CHEST PAIN ↳ The patient did not have chest pain. CATEGORY: PAST MEDICAL HISTORY [PRESENT] DIABETES MELLITUS ↳ History of diabetes mellitus. [PRESENT] CORONARY ARTERY DISEASE ↳ History of coronary artery disease, treated with a stent 10 years prior to admission.
* [1] The category is "History of Present Illness" for the finding "Progressive dyspnea over 6 months, with acute worsening over 3 days" and the explanation is "The patient had been in his usual state of health... until 6 months before admission, when shortness of breath developed. His exercise capacity gradually decreased from walking three blocks to climbing one flight of stairs. Three days before admission, dyspnea increased, and he had difficulty walking two to three steps on a level plane.". * [2] The category is "History of Present Illness" for the finding "Acute change in mental status, decreased appetite, and decreased urine output 2 days prior to admission" and the explanation is "During the next 2 days, his children noted that he was somnolent and less interactive than usual, with decreased appetite and urine output and an unsteady gait.". * [3] The category is "History of Present Illness" for the finding "Associated symptoms include an unchanged chronic dry cough and a weight loss of 2.3 kg in the last 2 months" and the explanation is "A chronic, dry cough was unchanged. He had lost approximately 2.3 kg within the previous 2 months.". * [4] The category is "History of Present Illness" for the finding "No fevers, chills, night sweats, chest pain, or leg edema reported" and the explanation is "The patient did not have fevers, chills, night sweats, chest pain, or leg edema.". * [5] The category is "History of Present Illness" for the finding "Pre-hospital treatment with oxygen and furosemide resulted in improvement" and the explanation is "During transport to this hospital by ambulance, oxygen (15 liters per minute) and furosemide were administered, with improvement.". * [6] The category is "History of Present Illness" for the finding "Syncopal episode with transient hypotension (SBP 60 mmHg) on day of admission" and the explanation is "Later that day, while walking with assistance, a syncopal episode occurred. The systolic blood pressure was initially 60 mm Hg; 5 minutes later, his blood pressure was 140/68 mm Hg.".	

Tabella 5.2.3 – Confronto visuale dei risultati dei file log. A sinistra l'astrazione risultante con l'implementazione originale e a destra l'astrazione ottenuta con il nuovo approccio.

5.3 Valutazione dell'integrazione di nuove evidenze cliniche

5.3.1 Analisi dei risultati del framework SCT (Approccio I)

L'Approccio I ha valutato la capacità del framework di stimare l'impatto di una nuova evidenza clinica su un'ipotesi diagnostica già formulata, secondo l'approccio SCT. Il confronto è stato condotto tra due modalità: approccio *zero-shot*, nel quale il modello riceve esclusivamente il contesto clinico, l'ipotesi e la nuova informazione, e f*ramework-aware*, nel quale al modello vengono inoltre fornite le aspettative deduttive e gli esiti induttivi prodotti dalle fasi precedenti del framework epistemologico.

La valutazione è stata effettuata su 42 quesiti SCT. Per ciascuna combinazione modello-approccio sono stati eseguiti 5 test indipendenti per caso, per un totale di 210 risposte. Ogni risposta è stata valutata tramite il criterio di punteggio SCT, nel quale il valore assegnato riflette il grado di concordanza tra la scelta del modello e la distribuzione delle risposte del panel di esperti. In aggiunta alla concordanza media, sono stati analizzati l'accordo con la risposta modale degli esperti, la distribuzione delle risposte nella scala Likert, la variabilità intra-caso e la consistenza del modello tra test ripetuti.

La Tabella 5.3.1 riporta i principali risultati aggregati per i due modelli considerati.

Modello	Approccio	Concordanza media	Accordo con moda esperti	Deviazione standard intra-caso	*Self-consistency* media
Gemini 2.5 Pro	*Zero-Shot*	0.6481	35.7% (15/42)	0.0573	93.8%
Gemini 2.5 Pro	*Framework-Aware*	0.6709	45.2% (19/42)	0.0499	92.9%
Gemini 2.5 Flash	*Zero-Shot*	0.6612	42.9% (18/42)	0.0959	88.1%
Gemini 2.5 Flash	*Framework-Aware*	0.6927	47.6% (20/42)	0.1224	81.9%

Tabella 5.3.1 – Risultati SCT aggregati per i paradigmi *zero-shot* e *framework aware.*

La concordanza è calcolata rispetto al punteggio basato sulla distribuzione delle risposte del panel di esperti; la *self-consistency* rappresenta la percentuale media di test che ripetono la scelta modale del modello per il medesimo caso.

In entrambi i modelli, l'impiego del contesto epistemologico produce un incremento della concordanza media. Per Gemini 2.5 Pro, il punteggio passa da 0.6481 a 0.6709, con un incremento assoluto di 0.0228, equivalente a 2.28 punti percentuali. La mediana dei punteggi medi per caso aumenta da 0.7800 a 0.8235; parallelamente, l'accordo con la risposta modale del panel di esperti cresce dal 35.7% al 45.2%, ossia da 15 a 19 quesiti su 42. Il numero di quesiti con concordanza media non superiore a 0.5 diminuisce da 18 a 14.

Una tendenza analoga si osserva per Gemini 2.5 Flash. In questo caso, la concordanza media aumenta da 0.6612 a 0.6927, con una differenza assoluta di 0.0315, pari a 3.15 punti percentuali. L'accordo con la moda esperta aumenta dal 42.9% al 47.6%, mentre il numero di casi a bassa concordanza diminuisce da 13 a 10. Il miglioramento descrittivo risulta quindi osservabile in entrambi i modelli e non appare limitato alla sola configurazione Gemini 2.5 Pro.

Per valutare se le differenze osservate fossero compatibili con un miglioramento sistematico, è stato applicato il test non parametrico dei ranghi con segno di Wilcoxon alle 42 coppie di punteggi medi per caso, confrontando *zero-shot* e *framework-aware* separatamente per ciascun modello.

Modello	Statistica (W)	Valore p
Gemini 2.5 Pro	33.5	0.402
Gemini 2.5 Flash	142	0.395

Tabella 5.3.2 – Risultati test non parametrico di Wilcoxon. Soglia significatività 0.05.

Come mostrato nella tabella 5.3.2, per nessuno dei due modelli si osserva una differenza statisticamente significativa.

I risultati dei due test devono essere interpretati con cautela. Il campione include soltanto 42 quesiti e la maggior parte dei casi presenta punteggi invariati, in particolare per Gemini 2.5 Pro. Il numero effettivo di coppie informative per il test, ossia quelle con differenza diversa da zero, è pertanto ridotto. Di conseguenza, l'esperimento è in grado di descrivere una direzione dell'effetto e di individuare variazioni caso-specifiche, ma ha una potenza limitata nel rilevare miglioramenti di piccola entità.

Accanto al punteggio SCT, è stata considerata una misura discreta di accordo: per ciascun caso è stata confrontata la scelta modale dell'LLM sulle 5 esecuzioni per quesito con la scelta modale del panel di esperti.

Per Gemini 2.5 Pro, 15 casi risultano concordanti con gli esperti in entrambe le condizioni e 23 risultano discordanti in entrambe. Il passaggio al *framework-aware* determina 4 transizioni delle scelte da disaccordo ad accordo, senza alcun caso in cui una risposta precedentemente concordante con gli esperti diventi discordante. Per Gemini 2.5 Flash, 14 quesiti risultano concordanti in entrambe le condizioni e 18 discordanti in entrambe. Sono presenti 6 transizioni favorevoli al *framework-aware* e 4 transizioni sfavorevoli. Il test esatto sulle coppie discordanti, 4 per il modello Pro e 10 per il modello Flash, equivalente a un test di McNemar esatto nel caso di dati binari appaiati, restituisce i valori riportati in Tabella 5.3.3.

Modello	Transizioni da sfavorevoli a favorevoli	Transizioni da favorevoli a sfavorevoli	Numero coppie discordanti	Valore p
Gemini 2.5 Pro	4	0	4	0.125
Gemini 2.5 Flash	6	4	10	0.754

Tabella 5.3.3 – Confronto appaiato dell'accordo con la moda degli esperti tra *zero-shot* e *framework-aware*, eseguiti per ciascun modello. Le transizioni riportano il numero di casi che passano da disaccordo ad accordo e da accordo a disaccordo rispetto alla moda del panel di esperti. Il valore p è stato calcolato con il test esatto di McNemar sulle sole coppie discordanti.

I risultati ottenuti non sono statisticamente significativi considerando una soglia per il valore p pari a 0.05.

L'analisi dei permette di valutare quanto le scelte del modello siano riproducibili a parità di input. Per Gemini 2.5 Pro la dispersione dei punteggi diminuisce leggermente, mentre la ripetizione della scelta modale resta elevata ma passa dal 93.8% nello *zero-shot* al 92.9% nel f*ramework-aware*. Per il modello Flash, entrambe le misure indicano una maggiore variabilità nell'approccio *framework-aware.*

Tale risultato è coerente con l'ipotesi che un prompt più ricco, contenente le aspettative deduttive e i relativi esiti di induzione, offra elementi aggiuntivi per la decisione ma aumenti al contempo la sensibilità del modello a variazioni stocastiche nella generazione.

La differenza tra Pro e Flash è rilevante anche sul piano architetturale. Per il modello Pro, il contesto strutturato sembra agire come un vincolo leggero, capace di orientare la risposta senza alterarne in misura sostanziale la riproducibilità. Per il modello Flash, invece, lo stesso contesto appare associato a una maggiore instabilità, pur osservando una concordanza media superiore. Il beneficio dell'Approccio I risulta pertanto dipendente non soltanto dal contenuto informativo fornito al modello, ma anche dalla capacità del modello stesso di gestire prompt più complessi.

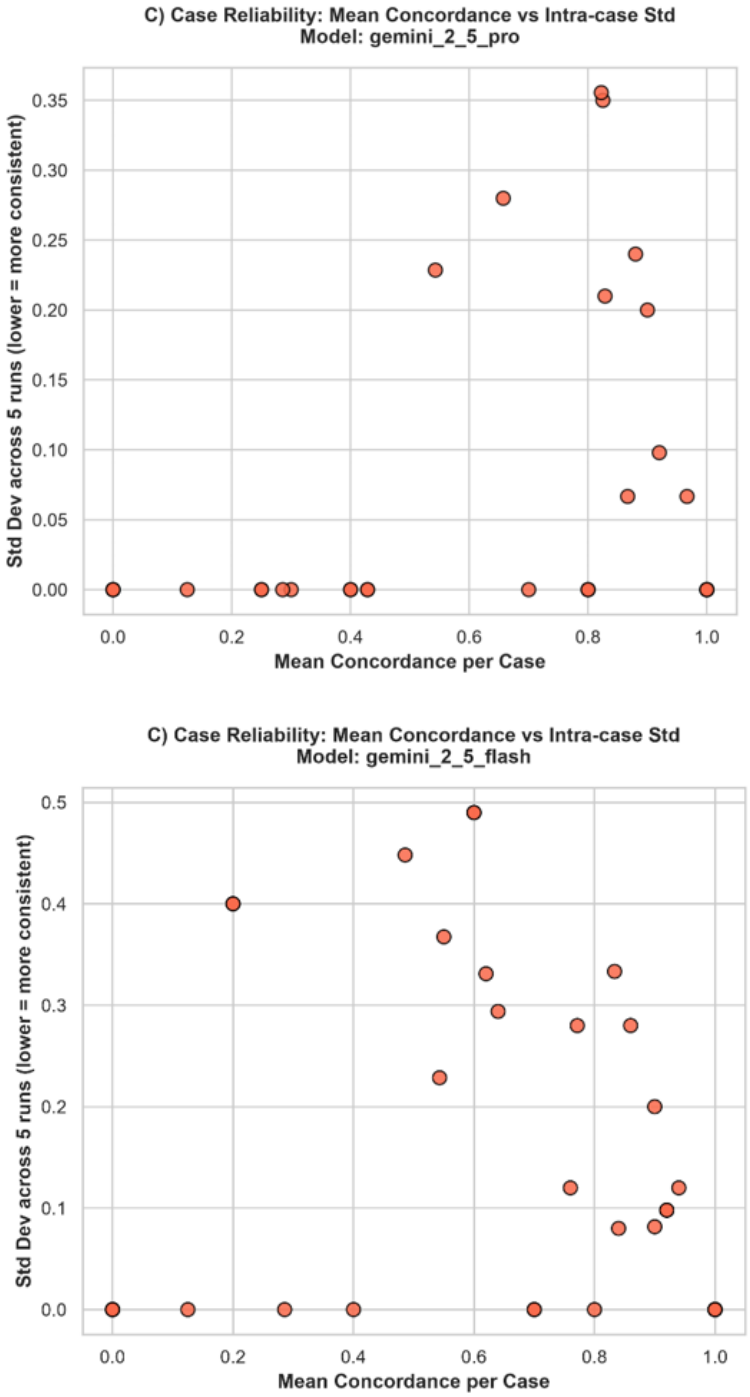


Figura 5.3.1 – Relazione tra concordanza media e variabilità intra-caso, architettura *framework-aware*. Ogni punto rappresenta un caso SCT (n = 42), posizionato secondo la concordanza media sulle 5 esecuzioni per quesito (asse x) e la deviazione standard tra le esecuzioni per lo stesso quesito (asse y; valori più bassi indicano maggiore consistenza). In alto, Gemini 2.5 Pro; in basso, Gemini 2.5 Flash.

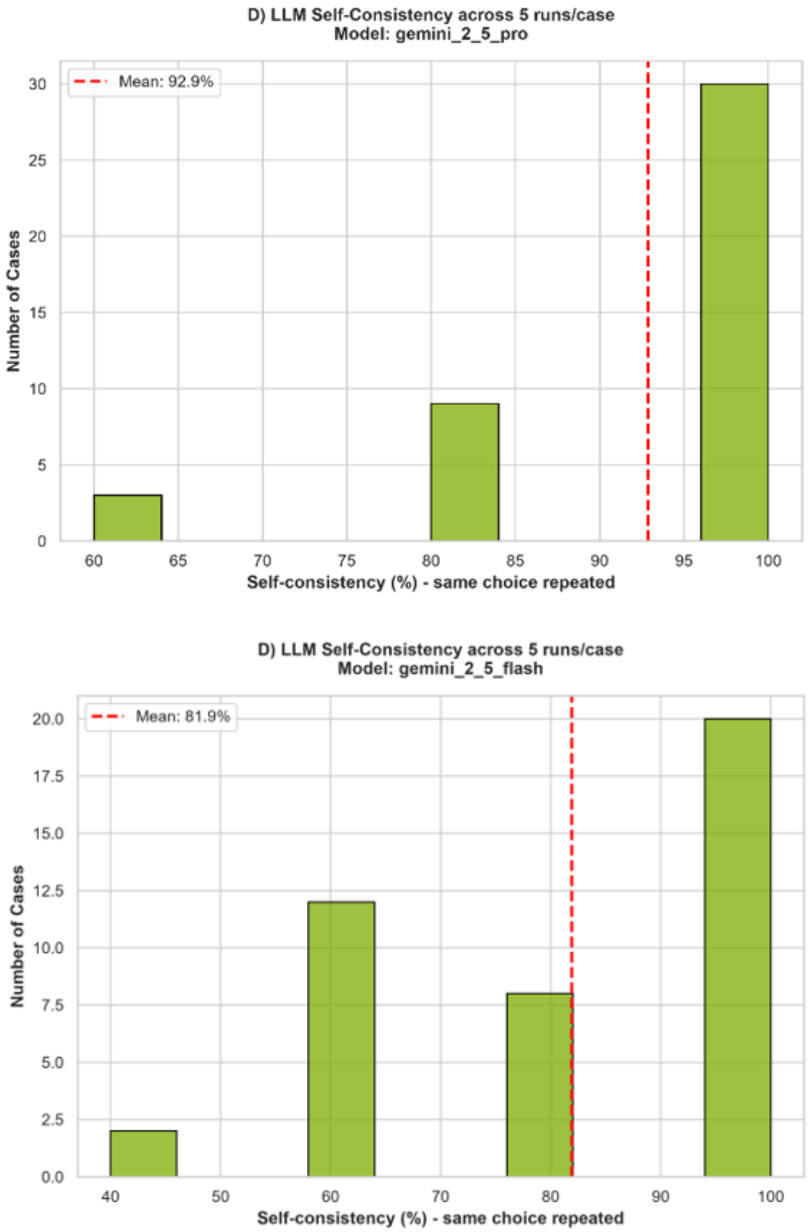


Figura 5.3.2 – Distribuzione della *self-consistency* nell'architettura *framework-aware*. La figura mostra, per ciascun modello, il numero di casi in cui la stessa scelta viene ripetuta nei 5 test. In alto è riportato Gemini 2.5 Pro, in basso Gemini 2.5 Flash. La linea tratteggiata indica la *self-consistency* media del modello: 92.9% per Gemini 2.5 Pro e 81.9% per Gemini 2.5 Flash.

La distribuzione osservata in Fig. 5.3.2 conferma visivamente la differenza di stabilità tra i due modelli. In Gemini 2.5 Pro, la maggior parte dei casi presenta una *self-consistency* pari al 100%, mentre Gemini 2.5 Flash mostra una frequenza maggiore di casi con consistenza compresa tra il 40% e l'80%. Il risultato è coerente con la deviazione standard intra-caso media precedentemente osservata (Fig. 5.3.1). Il paradigma *framework-aware* sembra quindi produrre un comportamento più stabile nel modello Pro, mentre nel modello Flash l'aumento della concordanza media è accompagnato da una maggiore variabilità tra le esecuzioni sullo stesso caso.

L’analisi della distribuzione delle scelte sulla scala Likert consente di osservare come i due approcci modifichino il comportamento decisionale dei modelli, oltre il solo punteggio di concordanza.

Per Gemini 2.5 Pro (Fig. 5.3.3), il *framework-aware* riduce leggermente la frequenza della risposta estrema positiva +2, dal 54.3% al 48.6%, e aumenta marginalmente il numero di risposte intermedie. Tuttavia, il modello continua a privilegiare nettamente le scelte estreme rispetto alla distribuzione del panel medico, che utilizza con maggiore frequenza le categorie –1 e +1. Questa osservazione è coerente con quanto riportato nello studio sul *benchmark* SCT [14], nel quale i modelli linguistici mostrano una tendenza a sottoutilizzare la risposta neutra e a sovrarappresentare variazioni forti di probabilità.

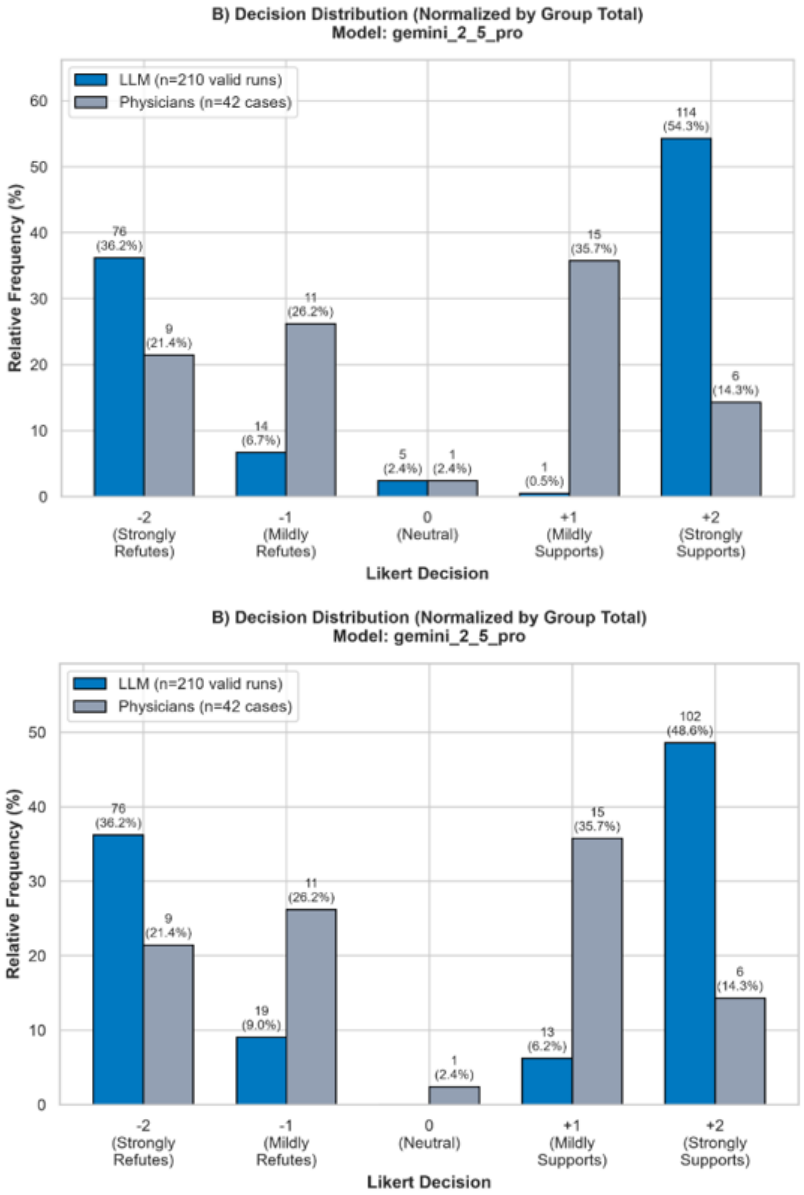


Figura 5.3.3 – Distribuzione delle risposte su scala Likert. La figura mostra i risultati del modello gemini-2.5-pro con approccio *zero-shot* (in alto) e *framework-aware* (in basso) a confronto con le risposte degli esperti.

Per Gemini 2.5 Flash, la redistribuzione è più evidente (Fig. 5.3.4). Il *framework-aware* riduce le risposte +2 dal 50.5% al 38.1% e aumenta le risposte +1 dal 3.3% al 13.8%; aumenta inoltre la frequenza di -2, dal 14.8% al 21.9%.

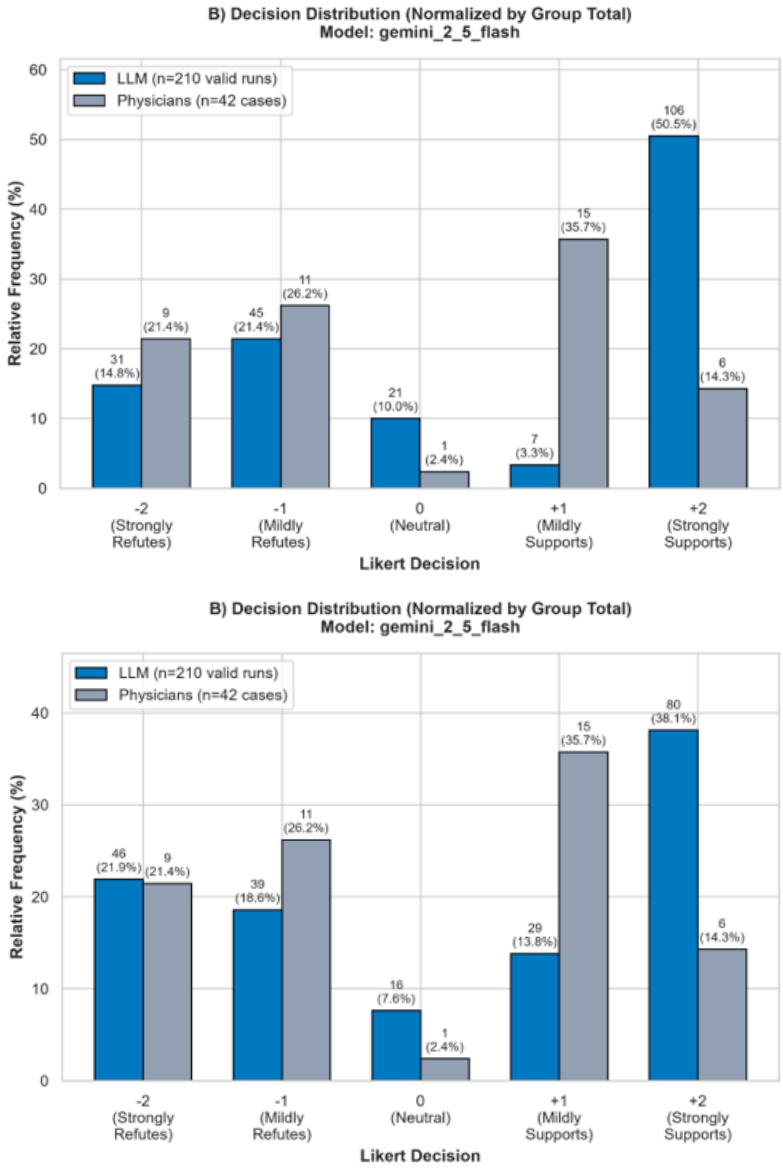


Figura 5.3.4 – Distribuzione delle risposte su scala Likert. La figura mostra i risultati del modello gemini-2.5-flash con approccio *zero-shot* (in alto) e *framework-aware* (in basso).

Il framework non determina quindi una semplice attenuazione delle risposte estreme: piuttosto, indica una modifica del profilo delle decisioni, riducendo il forte supporto per l'ipotesi e rendendo più frequente sia il supporto moderato sia, in parte, la forte confutazione. In altre parole, il contesto epistemologico non impone una regola di cautela generalizzata, ma può rendere il giudizio più sensibile alle aspettative cliniche per la singola ipotesi.

5.3.2 Confronto delle prestazioni: re-ranking (Approccio II) vs. re-abduzione (Approccio III) nel recupero delle omissioni diagnostiche

Il confronto tra l'Approccio II (aggiornamento induttivo e re-ranking) e l'Approccio III (riesecuzione completa del ciclo cognitivo), descritti nelle Sezioni 4.2.3 e 4.2.4, è stato condotto su 4 casi clinici del CPC-Bench. Per ciascun caso, alcune informazioni a supporto della diagnosi di riferimento sono state separate manualmente dal testo originale e fornite come nuova evidenza ai due approcci, per confrontare il modo in cui ciascuno le integra. La corrispondenza tra diagnosi predetta e di riferimento segue il criterio descritto nella Sezione 5.1.2. Per ciascun caso e ciascun approccio sono state eseguite 10 ripetizioni con il modello Gemini 2.5 Pro, per un totale di 40 esecuzioni per approccio e 80 complessive.

Per entrambi gli approcci è stato confrontato il ranking ottenuto dopo l'induzione del primo ciclo (senza evidenza) con il ranking ottenuto dopo l'aggiornamento basato sulle nuove evidenze. Nell'Approccio II vengono rieseguite l'induzione e il ranking; nell'Approccio III viene ripetuto l'intero ciclo, compresa la generazione delle ipotesi. Sono stati considerati separatamente il recupero di una diagnosi inizialmente assente e la variazione della posizione di una diagnosi già presente.

In 3 dei 4 casi clinici, la diagnosi di riferimento compare nel ranking iniziale in tutte le 10 esecuzioni di entrambe le configurazioni. In queste condizioni non vi sono omissioni da recuperare, ma è possibile osservare se la nuova evidenza modifica la posizione della diagnosi.

Nel quarto caso, la diagnosi è invece assente dal ranking iniziale in 6 esecuzioni su 10 per l'Approccio II e in 4 su 10 per l'Approccio III. Le omissioni iniziali riguardano quindi un unico caso clinico, analizzato attraverso esecuzioni ripetute. I ranking iniziali dei due approcci presentano quindi una diversa frequenza di omissione, da tenere distinta dall'effetto del successivo aggiornamento.

Approccio	Esecuzioni con omissione	Recuperate nel ranking finale	Tasso di recupero
Approccio III	4	0	0.0%

Tabella 5.3.4 – Esito del tentativo di recupero della diagnosi omessa.

Come mostrato nella Tabella 5.3.4, in nessuna delle 4 esecuzioni con omissione la diagnosi viene recuperata nel ranking finale.

Il risultato mostra che, in questo caso, la possibilità di rieseguire il ragionamento non è stata sufficiente a recuperare la diagnosi. La numerosità limitata e l'analisi del solo esito finale non permettono di stabilire la causa del mancato recupero.

Quando invece la diagnosi di riferimento è presente nel ranking iniziale, rimane presente anche dopo l'aggiornamento in 36 esecuzioni su 36 per l'Approccio III. Nel campione analizzato non si osservano quindi perdite della diagnosi già inclusa nell'elenco iniziale.

Per confrontare la dinamica della classifica prima e dopo l'introduzione delle evidenze aggiuntive è stato utilizzato, per ciascun approccio, un sottoinsieme di esecuzioni. In particolare, a partire dalle 40 esecuzioni per ognuno, sono stati rimossi:

- i casi in cui la diagnosi non compare nel primo ranking o nel ranking finale (6 per l'Approccio II e 4 per l'Approccio III) poiché non sarebbe possibile calcolare una variazione;
- le esecuzioni relative ad un caso clinico specifico (10 per ciascuno), poiché la diagnosi target è rimasta fissa in prima posizione durante tutte le esecuzioni.

Approccio	N. esecuzioni analizzate	Δ rango medio	Deviazione standard	Esecuzioni migliorate	Esecuzioni peggiorate	Esecuzioni invariate
Approccio II	24	+0.42	1.14	12	4	8
Approccio III	26	+0.54	2.37	11	6	9

Tabella 5.3.5 – Variazione del rango della diagnosi. Test con diagnosi presente sia nel primo ranking sia nel ranking finale. Il delta è calcolato come rango iniziale meno rango finale; i valori positivi indicano un miglioramento.

Nel sottoinsieme analizzato, il miglioramento medio è di 0.42 posizioni per l'Approccio II e di 0.54 per l'Approccio III, come osservabile in tabella 5.3.5. La dispersione è maggiore per l'Approccio III, con una deviazione standard di 2.37 rispetto a 1.14. Il risultato descrive aggiornamenti di entità più variabile per l'Approccio III, senza dimostrare una sua superiorità nel miglioramento del ranking.

Il test di Levene restituisce $p = 0.043$. Il risultato riguarda la dispersione nel sottoinsieme analizzato e deve essere interpretato considerando le ripetizioni sugli stessi casi clinici.

La maggiore dispersione osservata per l'Approccio III è compatibile con una procedura che rigenera ipotesi e valutazioni, anziché aggiornare soltanto quelle esistenti. Il confronto non permette tuttavia di separare l'effetto della procedura dalla variabilità delle risposte del modello sui casi considerati.

5.4 Dimostrazione funzionale e tracciabilità dell'interfaccia grafica

A differenza delle valutazioni quantitative delle Sezioni 5.2 e 5.3, questa sezione ha un obiettivo dimostrativo della funzionalità dell'interfaccia grafica. Non si tratta quindi di misurare l'accuratezza diagnostica del framework, già trattata nelle sezioni precedenti, ma di mostrare che l'infrastruttura di interazione uomo-macchina progettata nel Capitolo 4 è funzionalmente operativa e ispezionabile.

Per questa dimostrazione è stato utilizzato un caso clinico di prova creato sinteticamente, denominato NEJM_TEST, distinto dai 10 casi del CPC-Bench impiegati nella valutazione del nodo di astrazione (Sezione 5.2). La scelta di un caso sintetico, anziché di uno dei 10 casi già utilizzati nella sperimentazione, è stata pensata per limitare l'uso dei casi NEJM alla valutazione quantitativa e qualitativa dell'accuratezza del framework, mentre NEJM_TEST viene impiegato esclusivamente come fonte dimostrativa per illustrare il funzionamento dell'interfaccia, senza sovrapporre le due finalità e senza che eventuali interventi manuali di modifica sul ranking, effettuati a scopo puramente illustrativo, possano interferire con l'interpretazione dei risultati sperimentali discussi in precedenza.

L'interfaccia si apre sulla *Home Page*, che ricopre la funzione di modulo di configurazione e acquisizione dati descritta in Sezione 4.3.2. Come osservabile in Figura 5.4.1, la pagina presenta due caselle di testo e alcuni ulteriori elementi di interazione:

- Una prima casella di testo (*Patient / Case ID*), in cui è possibile inserire manualmente il codice identificativo univoco del caso clinico in esame;
- Una seconda casella di testo (*Clinical Text)*, in cui è possibile inserire manualmente la narrativa clinica del caso;
- Un componente di tipo *drag-and-drop*, che permette di trascinare o selezionare dalla memoria del dispositivo un file con estensione .csv o .xlsx. Se le intestazioni del file caricato corrispondono alla nomenclatura standard attesa ("Case-Id" e "Case") o simili, i due campi precedenti vengono compilati automaticamente; in caso contrario, si apre una finestra che obbliga l'utente a mappare manualmente le colonne del proprio file verso le variabili di sistema, come mostrato in Figura 5.4.2;
- Un interruttore "*Enable Step-by-Step Debug Mode"* per la definizione della modalità con cui si intende procedere:
    - se attivo, il *thread* di elaborazione verrà sospeso dopo l'esecuzione di ogni nodo;
    - se non attivo, il *thread* di elaborazione verrà sospeso solo al termine delle fasi di ranking.
- Due pulsanti, rispettivamente per cancellare i contenuti inseriti (*Clear Inputs*) e per avviare il ciclo di ragionamento ("*Start Reasoning"*).

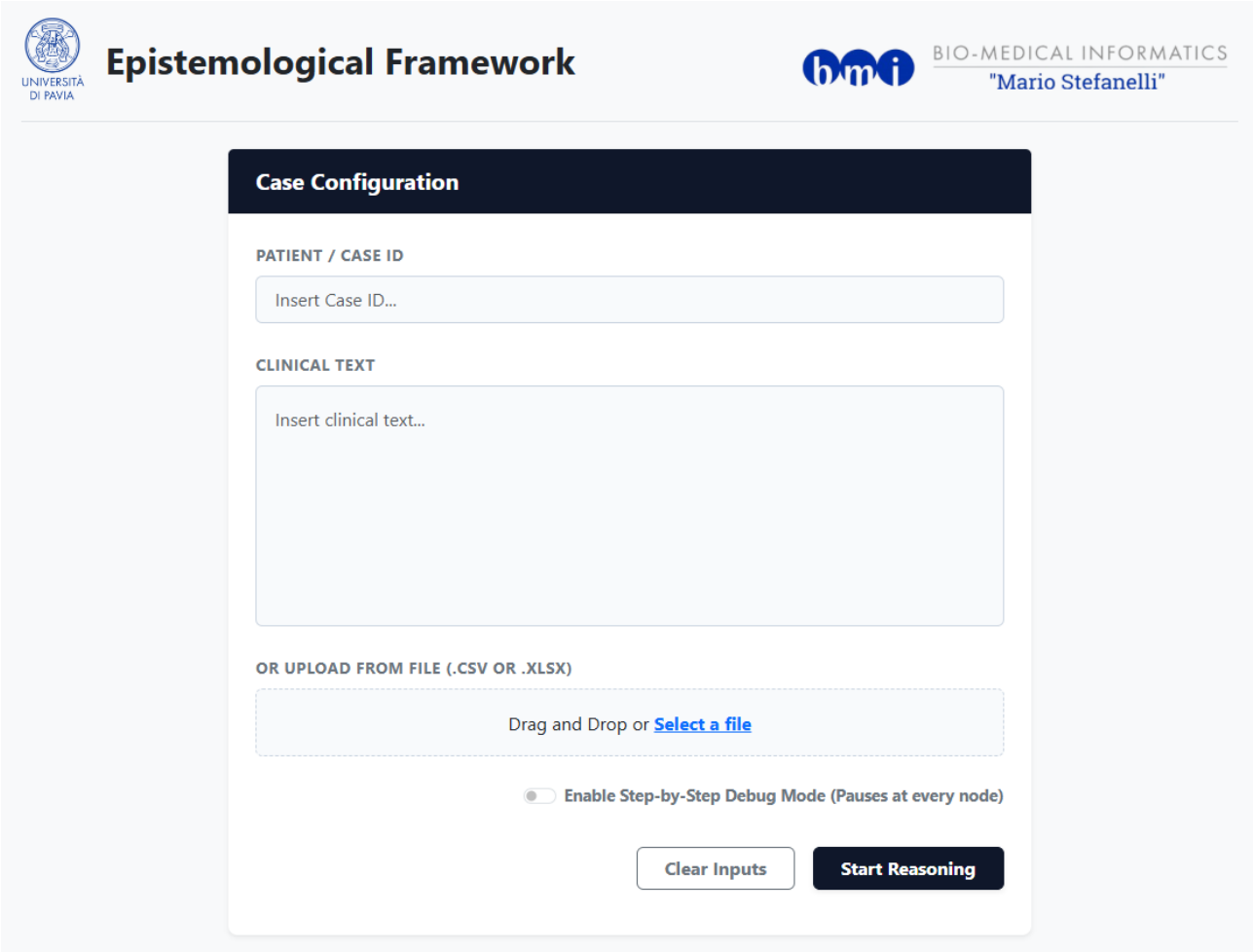


Figura 5.4.1 - Home page. Moduli di configurazione e acquisizione del caso clinico.

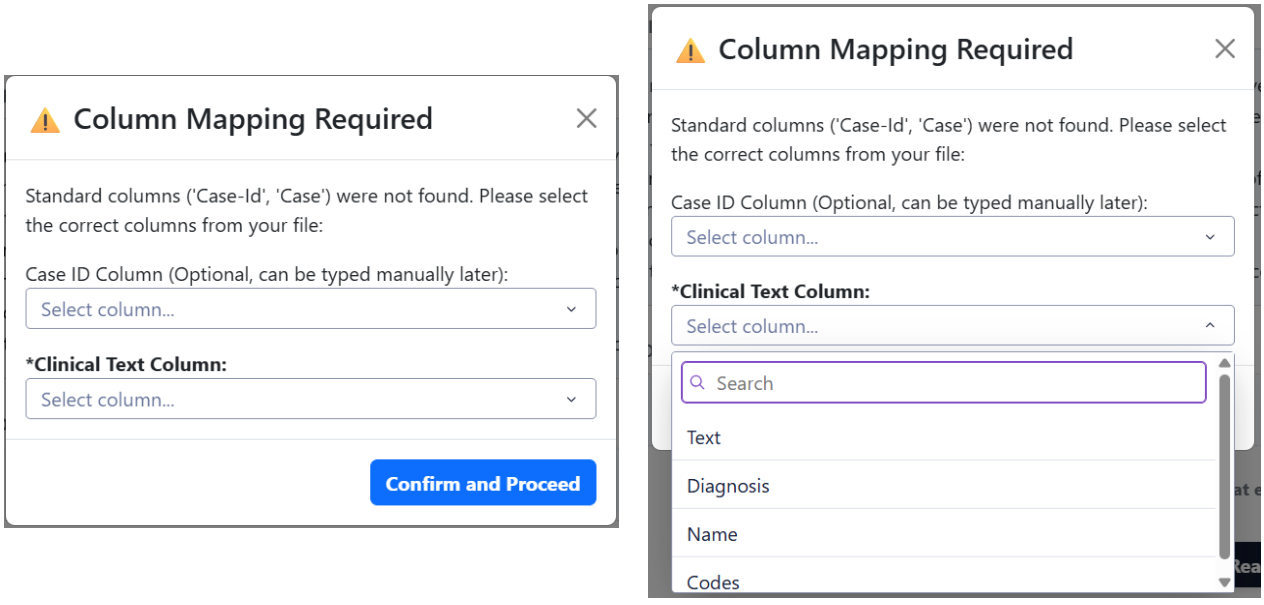


Figura 5.4.2 – Selezione manuale delle corrispondenze. Finestra di dialogo per la selezione delle corrispondenze tra colonne del file caricato e variabili di sistema.

Una volta completato il processo di inserimento, viene mostrato un avviso di successo e diventa possibile avviare il ragionamento. Alla conferma, i dati inseriti vengono validati e trasferiti alla pagina di elaborazione tramite il meccanismo di sessione descritto in Sezione 4.3.2, senza il ricorso a parametri esposti nell'URL.

A questo punto l'interfaccia reindirizza automaticamente l'operatore alla pagina "*Results"*, nella quale viene mostrato il processo di ragionamento in corso. Come illustrato in Figura 5.4.3, la pagina è organizzata in quattro aree funzionali:

- Nella barra superiore viene mostrato il codice identificativo del paziente, insieme a due pulsanti interattivi: "*Save Progres*s", che consente di salvare lo stato corrente dell'esecuzione, disponibile solo quando il ciclo di ragionamento non è attivo, per evitare corruzioni dello stato condiviso, e *Start New Case*, che permette di avviare un nuovo caso clinico in qualsiasi momento, interrompendo l'esecuzione in corso;
- Nella parte centrale si trova un contenitore a schede (*tab*) che raccoglie i risultati principali del ragionamento: la scheda *Diagnostic Ranking*, che mostra la classifica diagnostica non appena il nodo di ranking corrispondente ha terminato l'esecuzione, e la scheda *Expected Evidenc*e, che mostra le conseguenze cliniche attese per ciascuna ipotesi al termine dell'esecuzione del nodo di deduzione;
- Nella parte inferiore si trovano i pannelli espandibili (*accordion*) che implementano il principio di tracciabilità descritto in Sezione 4.3.4: una volta completato il nodo corrispondente, ciascun pannello rende disponibile il contenuto testuale grezzo prodotto da quella fase, come illustrato in Figura 5.4.4 per il nodo di astrazione;
- Nella colonna laterale destra è presente un componente che visualizza l'avanzamento del framework lungo l'intera sequenza dei nodi del grafo, colorando in blu il nodo attualmente in elaborazione e in verde i nodi la cui esecuzione è stata completata. Al di sopra di questo schema si trova il pulsante di avanzamento *Next Step*, tramite il quale l'operatore autorizza la ripresa dell'esecuzione ogni volta che il ciclo si sospende in attesa di revisione umana.

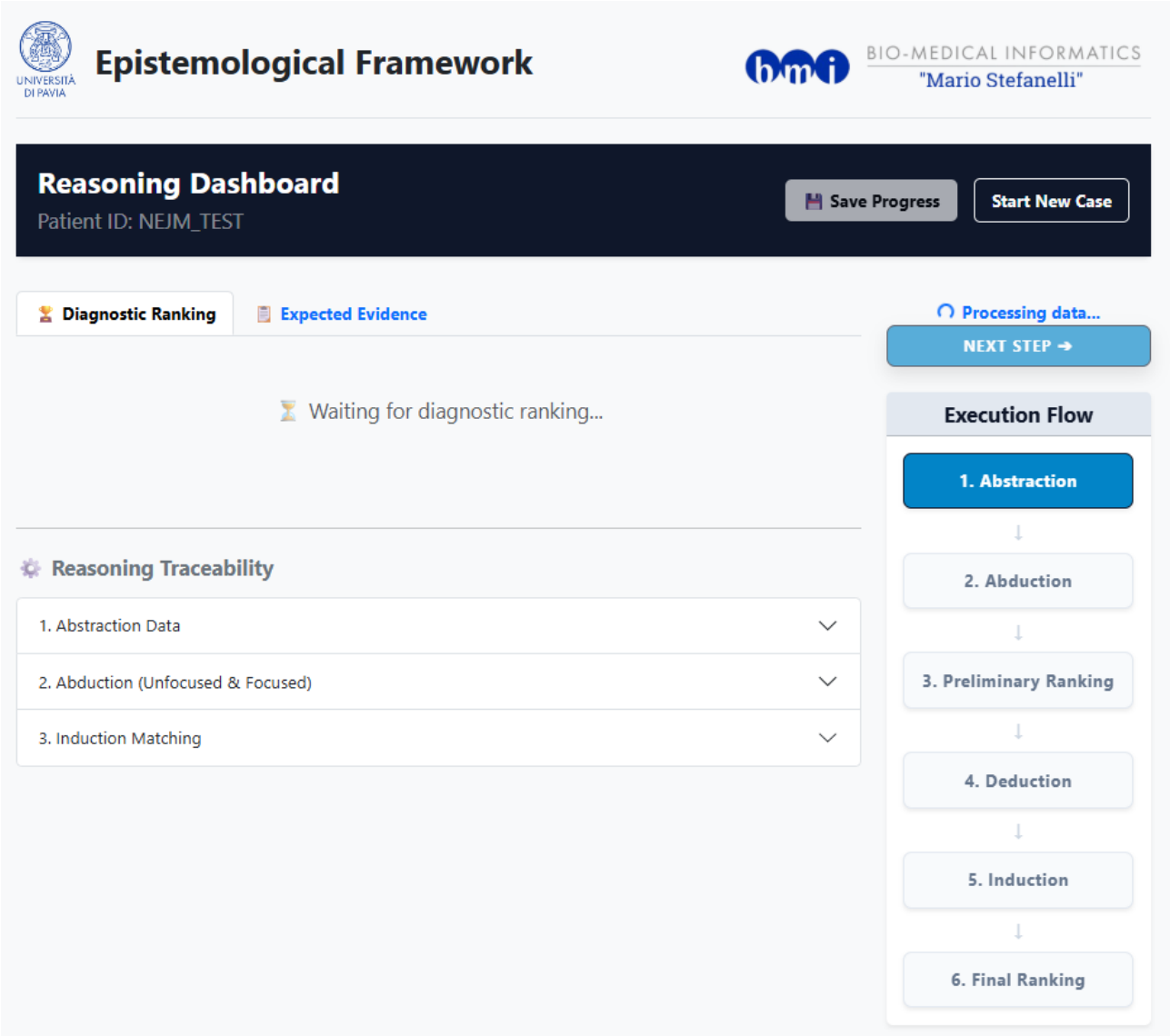


Figura 5.4.3 – Pagina *"Results"*. Panoramica generale durante l'esecuzione del framework.

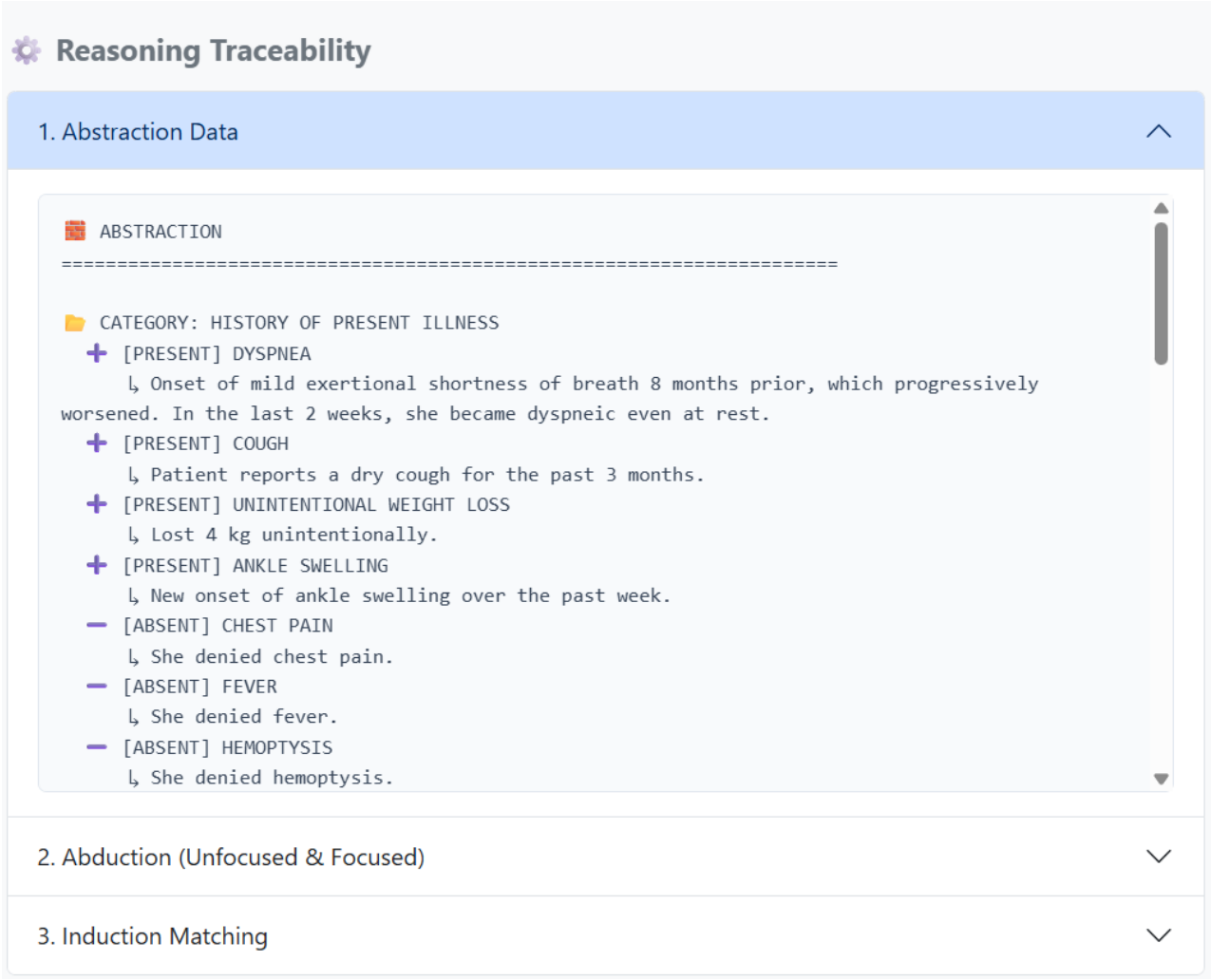


Figura 5.4.4 – Esempio di visualizzazione della tracciabilità. Contenuto del nodo di astrazione.

Al completamento del nodo di ranking (preliminare, calcolato subito dopo l'abduzione focalizzata), la classifica stilata dal modello viene visualizzata nel tab *Diagnostic Ranking*, come mostrato in Figura 5.4.5. In questa configurazione è possibile osservare l'elenco delle diagnosi con la relativa posizione in classifica; cliccando sul pulsante *Details* associato a ciascuna diagnosi, l'utente può espandere i dettagli nelle quattro dimensioni cliniche del framework, parsimonia (*Parsimony*), pericolosità (*Danger*), costo (*Cost*) e curabilità (*Curability*), così da ispezionare direttamente le giustificazioni prodotte dal modello per ciascun criterio, come mostrato in Figura 5.4.6.

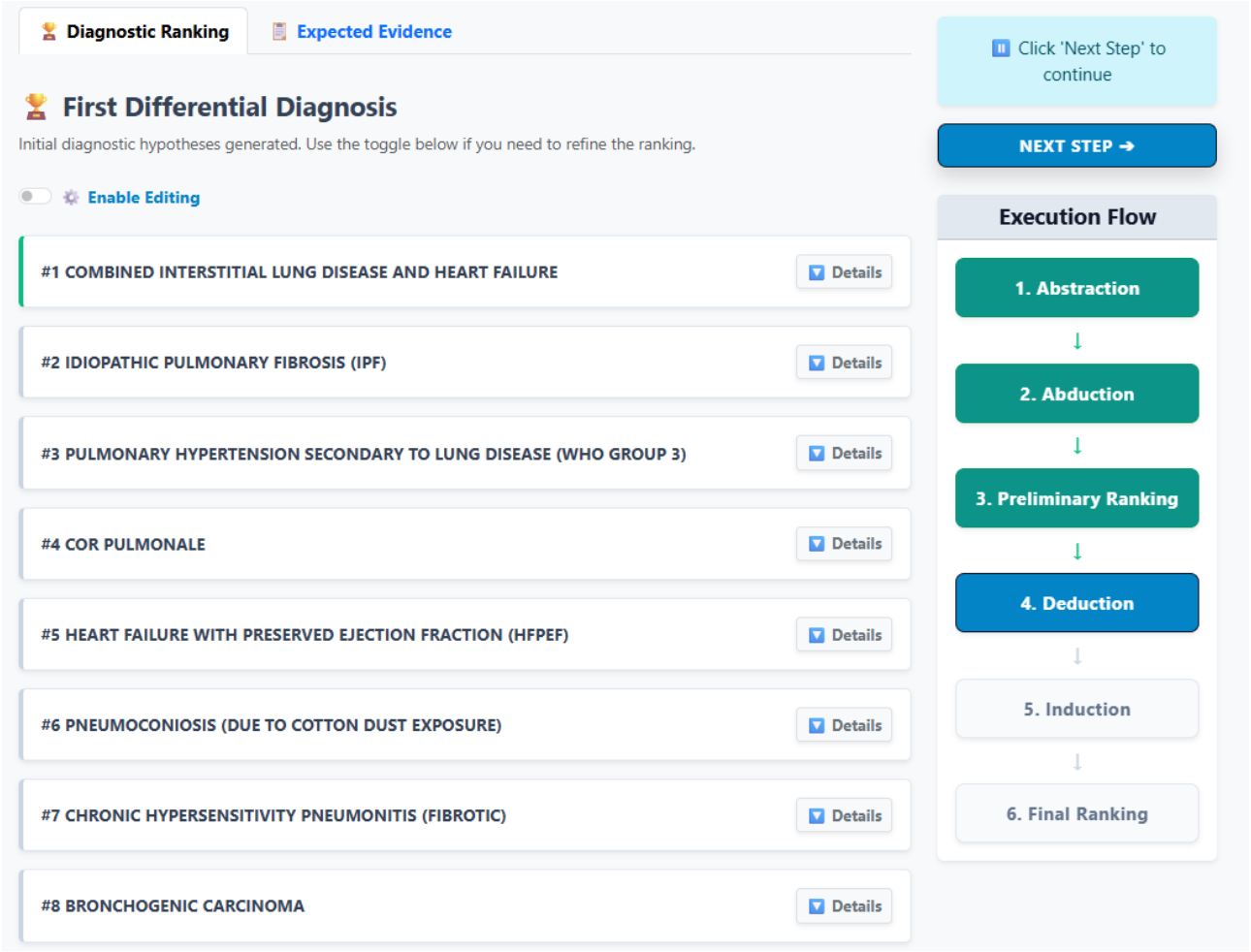


Figura 5.4.5 – Visualizzazione del ranking preliminare.

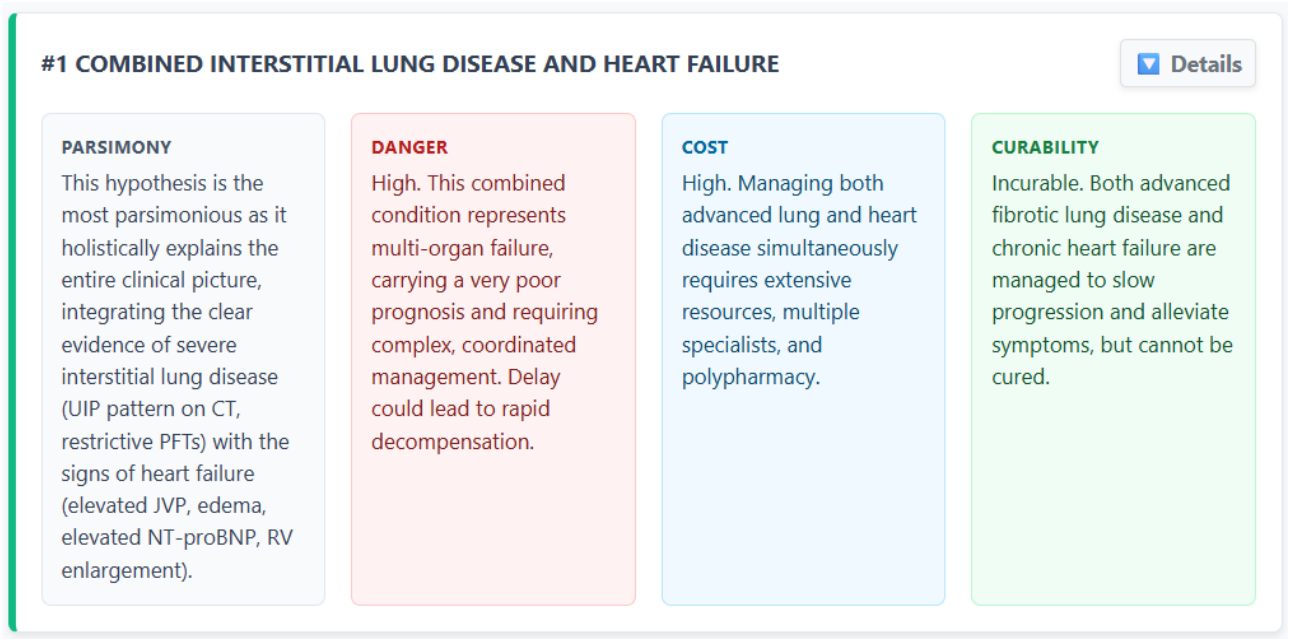


Figura 5.4.6 – Visualizzazione dei dettagli della singola diagnosi.

È inoltre possibile abilitare la modifica del ranking tramite l'apposito interruttore "*Enable Editing*", che sblocca le funzionalità di modifica mostrate in Figura 5.4.7. Attraverso questa interfaccia, l'operatore può:

- modificare l'ordine delle diagnosi utilizzando le apposite frecce direzionali, realizzando concretamente l'intervento HITL discusso in Sezione 4.3.3: il clinico può correggere il ranking qualora rilevi diagnosi poco probabili collocate nelle prime posizioni, o viceversa diagnosi clinicamente rilevanti posizionate in fondo alla classifica. Ogni spostamento genera immediatamente, accanto alla diagnosi interessata, un indicatore a sfondo giallo con icona a forma di matita (✏️), che permette all'utente di tenere traccia delle modifiche manuali effettuate e della relativa variazione di posizione rispetto al ranking originario proposto dal modello;
- eliminare una diagnosi ritenuta clinicamente improbabile, tramite il pulsante con la croce rossa (❌). Per prevenire cancellazioni accidentali, ogni pressione del pulsante genera una finestra modale di conferma dell'azione (Figura 5.4.8).

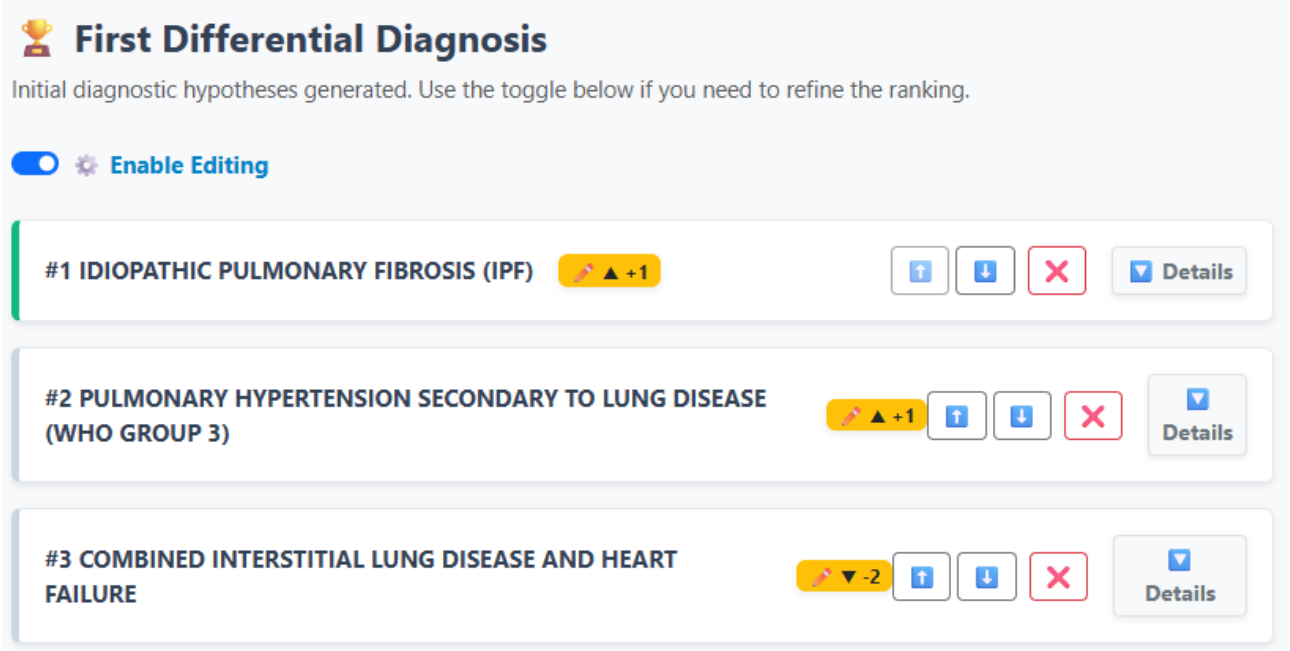


Figura 5.4.7 – Abilitazione della modifica del ranking.

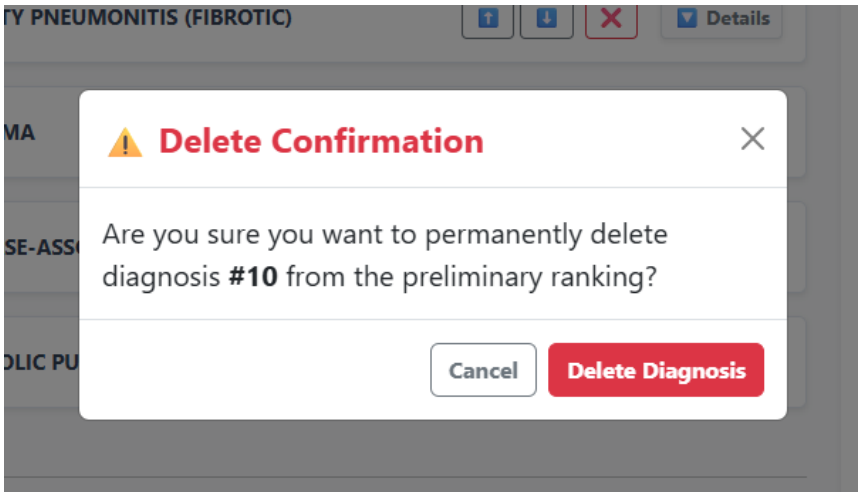


Figura 5.4.8 – Finestra di conferma per l'eliminazione di una diagnosi.

Terminate le eventuali modifiche, l'utente può far ripartire il ciclo di ragionamento tramite il pulsante *Next Step*. A questo punto, il nodo successivo del grafo (*Deduction*) opera sul ranking così come validato, e non necessariamente modificato, dall'operatore.

Al termine della fase di deduzione, il tab *Expected Evidence* mostra le evidenze cliniche attese per ciascuna diagnosi, come illustrato in Figura 5.4.9. Una legenda associa un colore distintivo a ciascun grado di priorità (*High, Medium, Low*) e identifica le diverse tipologie di evidenza previste dallo schema del framework (*Manifestation, Test_result, Clinical_course*), coerentemente con la struttura descritta in Sezione 3.2.4. Questa componente è stata introdotta per rendere esplicito, prima ancora del confronto induttivo con il caso osservato, quali elementi clinici il sistema si attende di trovare per ciascuna ipotesi, realizzando così, anche in questa fase intermedia del ciclo, il principio di trasparenza diagnostica richiesto in Sezione 4.3.1.

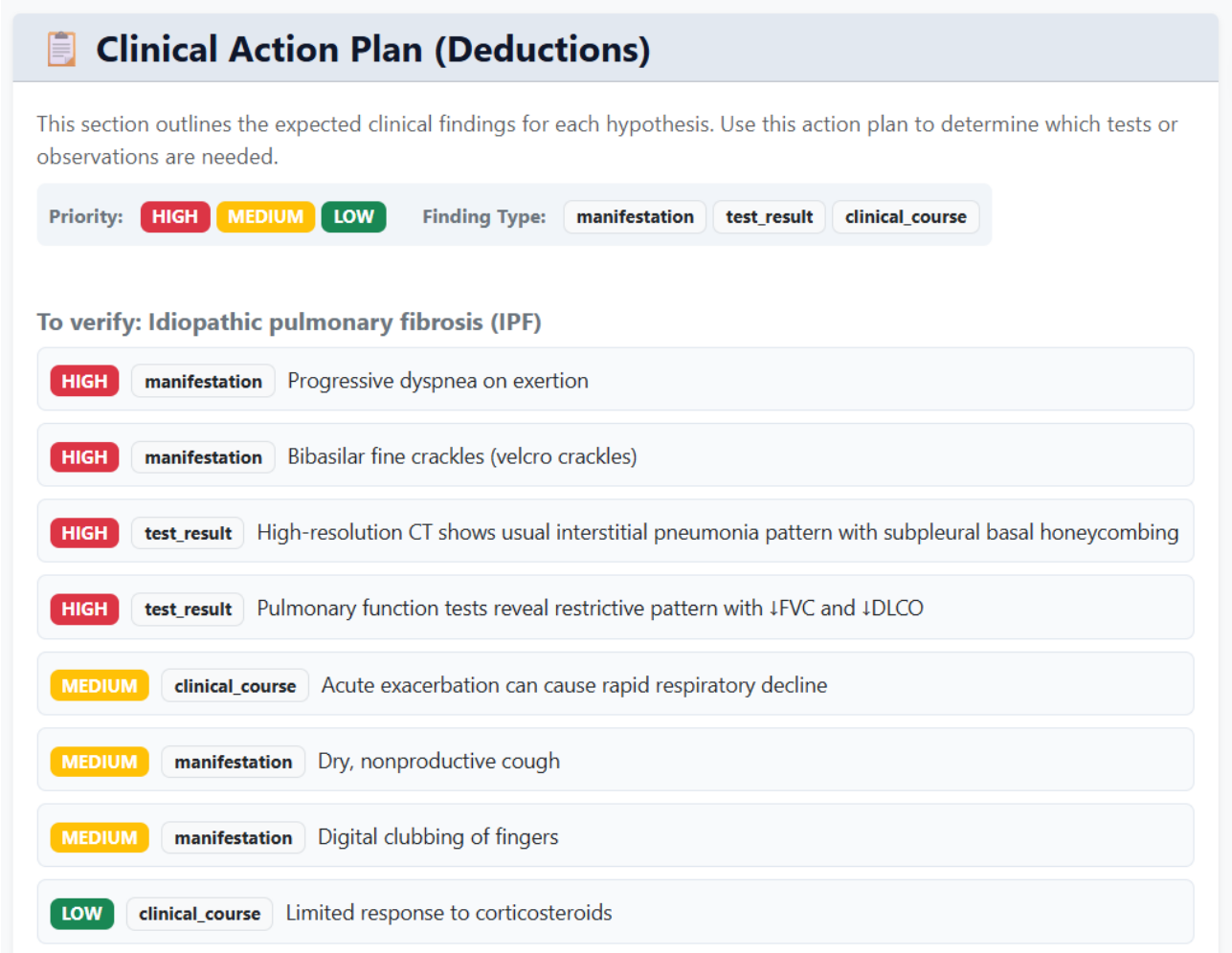


Figura 5.4.9 – Pagina di visualizzazione delle evidenze attese (*Deductions*).

Al termine dell'intero processo, l'interfaccia mostra un messaggio di completamento e il ranking si aggiorna per presentare la classifica finale. Questa vista riprende gli stessi elementi del ranking preliminare, arricchiti dagli indicatori di variazione descritti in Sezione 4.3.4 (Figura 5.4.10): un indicatore verde con il numero di posizioni guadagnate segnala un miglioramento in classifica rispetto al ranking di partenza; un indicatore rosso segnala invece il numero di posizioni perse; un indicatore grigio con il simbolo "=" segnala infine una posizione rimasta invariata. Questa codifica cromatica, distinta da quella gialla usata per le modifiche manuali del *First Ranking*, consente all'operatore di distinguere con precisione le variazioni introdotte dall'intervento umano da quelle prodotte dal successivo ciclo di deduzione e induzione.

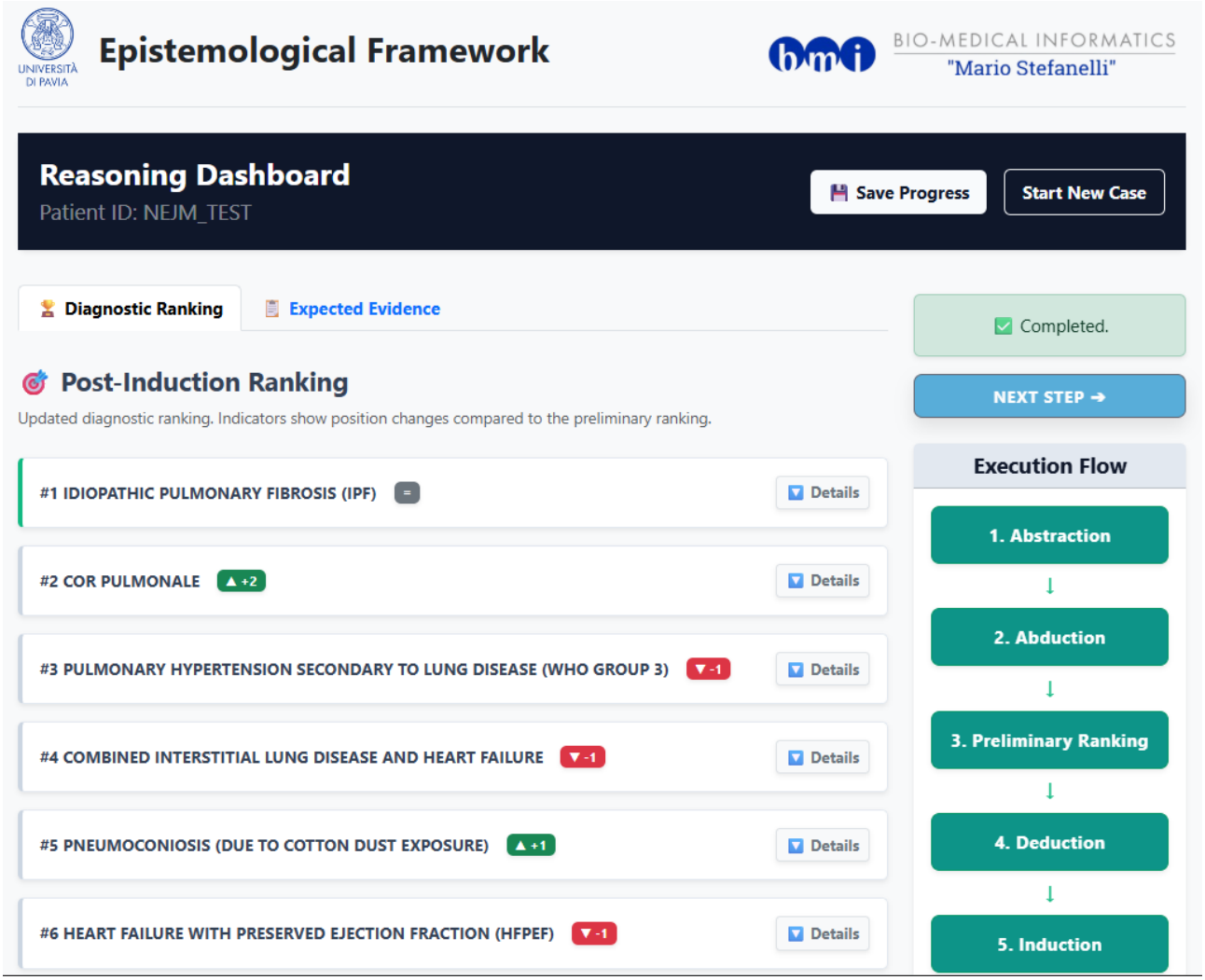


Figura 5.4.10 – Visualizzazione del ranking finale.

CAPITOLO 6: Discussioni e prospettive future

6.1 Bilanciamento tra vincoli di contesto e *lossy abstraction*

Le modifiche al nodo di astrazione hanno riguardato l'organizzazione dei reperti e la sintesi delle informazioni associate a ciascuno. Inoltre, nella configurazione ottimizzata, i nodi successivi utilizzano questa rappresentazione e i risultati intermedi del ragionamento, senza accedere direttamente al testo clinico originale. I risultati della Sezione 5.2 permettono di discutere due aspetti: la riduzione del volume testuale e le variazioni delle prestazioni diagnostiche.

Il primo aspetto riguarda il carico informativo: il numero medio di caratteri dell'output di astrazione diminuisce del 30.3%. Il risultato misura la dimensione della rappresentazione prodotta, non il consumo complessivo di token o i tempi di esecuzione dell'intero framework. Il lavoro di riferimento suggerisce che un elevato carico di prompt possa penalizzare soprattutto i modelli più piccoli [15]. Tuttavia, il confronto tra le due configurazioni dell'astrazione è stato condotto soltanto con Gemini 2.5 Pro e non permette di verificare un beneficio specifico per modelli di dimensioni inferiori.

Il secondo riguarda l'ordinamento. Le metriche Top-1, Top-3 e MRR peggiorano, mentre l'accuratezza Top-10 aumenta solo marginalmente in senso opposto (Sezione 5.2.1). Una possibile spiegazione, che resta un'ipotesi e non una relazione dimostrata, riguarda il modo in cui gli LLM elaborano il contesto. Nel testo clinico discorsivo, i reperti sono presentati insieme al loro contesto, per esempio l'esordio di un sintomo e il suo successivo peggioramento. Suddividendo il testo in campi separati, ogni informazione diventa più facile da individuare e utilizzare singolarmente, come un farmaco o un reperto negativo; tuttavia, questa suddivisione può rendere meno evidente il legame tra i diversi reperti e la loro evoluzione nel tempo, elementi che nel testo originale potevano aiutare il modello a formulare le prime ipotesi diagnostiche. Con campioni di queste dimensioni, la lieve variazione della Top-10 va letta come tendenza descrittiva e non come beneficio compensativo.

Il valore del framework non si misura solo nella posizione della diagnosi corretta, ma comprende anche la disponibilità di ipotesi e valutazioni intermedie ispezionabili. Questo contributo non elimina però la necessità di migliorare l'ordinamento diagnostico. La possibilità che ulteriori aggiornamenti o la supervisione umana compensino il calo osservato richiede una valutazione specifica.

La *lossy abstraction* è quindi un compromesso strutturale del progetto tra la conservazione delle informazioni cliniche estratte e tre obiettivi: scambio di dati strutturati tra i nodi, tracciabilità degli output e riduzione del carico informativo. La stessa scelta comporta un limite strutturale: l'astrazione diventa un collo di bottiglia informativo, perché un reperto omesso o aggregato in modo inadeguato non è recuperabile dai nodi successivi, privi di accesso al testo originale. La versione ibrida del nodo mitiga questo rischio con regole di deduplicazione e sintesi temporale, ma non lo elimina. Questa osservazione è alla base delle limitazioni discusse nella Sezione 6.2.

6.2 Limiti dell'implementazione attuale e natura preliminare del ciclo di retroazione post-induzione

Nonostante le modifiche introdotte nel Capitolo 4, l'architettura implementata presenta limiti significativi rispetto al modello epistemologico completo di Ramoni et al. (Capitolo 2). Alcuni di essi sono già stati individuati nell'architettura di base (Sezione 3.4) e restano solo parzialmente superati, altri emergono dai risultati sperimentali del Capitolo 5.

Il primo limite riguarda la natura statica dell'input. Il testo clinico è fisso al momento dell'avvio e non esiste ancora un meccanismo strutturato per inserire nuove evidenze durante l'esecuzione del grafo, nemmeno nell'interfaccia grafica, che consente di inserire un caso e di modificare il ranking ma non di aggiornare il quadro clinico. Il secondo limite è l'assenza di un ciclo di acquisizione attiva delle informazioni. Nel modello originale la fase di test può richiedere ulteriori dati e riaprire il ciclo abduttivo, mentre il sistema attuale non formula richieste informative. Gli Approcci II e III sono un primo tentativo di colmare questa mancanza, ma restano sperimentali, valutati offline (non integrati nell'interfaccia) e limitati a un singolo ciclo di aggiornamento.

Nell'unico caso con omissione reale, la ripetizione dell'intero ciclo non ha dimostrato capacità di recuperare la diagnosi mancante (Sezione 5.3.2). Questo indica che la libertà di rigenerare lo spazio delle ipotesi, pur necessaria, non basta se il modello non associa la nuova evidenza alla patologia corretta. Nelle esecuzioni considerate non è stata osservata la perdita di diagnosi già presenti.

I risultati vanno letti alla luce di limiti metodologici che ne circoscrivono la generalizzabilità:

- Numerosità**:** i campioni sono ridotti (10 casi CPC-Bench per l'astrazione, 42 quesiti SCT, 4 casi per il confronto tra Approcci II e III). Per questo i test statistici (Wilcoxon, test esatto sulle coppie discordanti, Levene) sono interpretati come indicazioni di tendenza e hanno potenza limitata per differenze piccole;
- Un solo caso con omissione: la conclusione sul mancato recupero si basa su 4 esecuzioni relative a un unico caso clinico e non è estendibile ad altre patologie o altri pattern di evidenza;
- Soggettività della separazione delle evidenze: la distinzione tra narrativa iniziale ed evidenza successiva è stata fatta manualmente, senza un criterio standardizzato o validato da esperti;
- Un solo modello per gli Approcci II e III: il confronto è stato condotto solo con Gemini 2.5 Pro. Non si può quindi stabilire se la maggiore volatilità dell'Approccio III dipenda dal meccanismo o dal modello;
- Criterio di corrispondenza diagnostica: il confronto lessicale è riproducibile, ma può non riconoscere formulazioni clinicamente equivalenti oppure associare espressioni simili riferite a condizioni diverse. La soglia adottata non è stata validata su corrispondenze annotate da esperti;
- Contaminazione dei *benchmark*: i casi CPC-Bench e SCT sono pubblici e potrebbero essere stati incontrati dai modelli durante l'addestramento (Sezione 5.1.1).

Questi limiti, insieme alla natura preliminare del meccanismo di retroazione non monotona, motivano gli sviluppi futuri (Sezione 6.4). In particolare, indicano come priorità l'estensione della validazione

a più casi con omissione documentata, a una procedura standardizzata di separazione dell'evidenza incrementale e alla revisione della logica di gestione delle nuove evidenze.

6.3 L'interfaccia grafica come strumento di allineamento uomo-macchina

L'interfaccia grafica sviluppata nel Capitolo 4 non va interpretata come un semplice strato di visualizzazione degli output algoritmici, ma come un componente strutturale del framework, progettato per realizzare concretamente il paradigma HITL discusso nella Sezione 2.4. L'allineamento tra umani e AI non riguarda solo l'output finale, ma anche il processo inferenziale sottostante: l'output del modello deve essere interpretabile in termini di concetti cognitivi umani (ipotesi, aspettative, conferme) e il processo deve essere tracciabile e ispezionabile.

L'architettura implementata risponde a questi requisiti attraverso tre meccanismi distinti. In primo luogo, la sospensione controllata dell'esecuzione (Sezione 4.3.3) consente al clinico di intervenire nei punti decisionali del grafo, in particolare ai nodi di ranking, modificando attivamente la classifica delle ipotesi prima che il ciclo prosegua verso deduzione e induzione. Questa funzionalità realizza concretamente il principio secondo cui l'esperto umano mantiene l'autorità finale sulla diagnostica, intercettando potenziali errori del modello prima che influenzino le fasi successive.

In secondo luogo, il sistema di *Delta-Badge* e tracciabilità delle modifiche (Sezione 4.3.4) permette di attribuire con precisione la fonte di ogni variazione di ranking. Questa distinzione non è puramente estetica ma rende esplicita la distinzione tra correzione umana e aggiornamento prodotto dal sistema.

In terzo luogo, la persistenza strutturata dello stato (Sezione 4.3.5) garantisce che l'intero processo, inclusi gli interventi umani, sia archiviabile e ispezionabile a posteriori. Il contenuto JSON generato dal pulsante "*Save Progress"* incapsula non solo i dati grezzi di inferenza (output di ciascun nodo del grafo), ma anche la memoria dell'interfaccia utente: il ranking originario calcolato dal modello, affiancato al ranking eventualmente modulato dal clinico. Questa infrastruttura pone le fondamenta per future funzionalità di recupero, che consentiranno di ricaricare un file JSON per riprendere il processo diagnostico da uno stato pregresso, un requisito operativo rilevante in contesti clinici reali, dove il processo decisionale può svilupparsi su orizzonti temporali prolungati e richiedere pause e riprese.

La dimostrazione funzionale della Sezione 5.4 ha verificato che questi meccanismi operino correttamente lungo l'intero ciclo di esecuzione, dalla sospensione del *thread* di ragionamento alle modifiche dell'operatore, fino alla ripresa del ciclo e alla persistenza tracciabile dello stato finale. Non si è trattato di misurare l'accuratezza diagnostica del framework (già trattata nelle Sezioni 5.2 e 5.3), ma di mostrare che l'infrastruttura di interazione uomo-macchina progettata nel Capitolo 4 è funzionalmente operativa e ispezionabile.

In questa prospettiva, l'interfaccia grafica non è un accessorio opzionale, ma uno strumento fondamentale: traduce concetti astratti del modello ST (abduzione, ranking, deduzione, induzione) in oggetti visivi manipolabili (*card, badge, accordion*), rendendo il processo inferenziale comprensibile e correggibile da parte del clinico. Questo allineamento tra rappresentazione computazionale e

cognizione umana è una condizione necessaria, sebbene non sufficiente, per l'adozione di sistemi di supporto alle decisioni cliniche in contesti reali.

6.4 Sviluppi futuri

6.4.1 Integrazione del ciclo non-monotonico nell'interfaccia: inserimento di nuove evidenze cliniche

Come discusso nella Sezione 6.2, il limite più rilevante dell'architettura attuale è l'assenza di un vero ciclo di acquisizione attiva delle informazioni: il testo clinico è fisso al momento dell'avvio e gli Approcci II e III (Sezioni 4.2.3-4.2.4) sono stati validati come esperimenti separati dall'interfaccia grafica, e non come funzionalità integrate.

Uno sviluppo plausibile consiste nell'esporre direttamente nell'interfaccia un meccanismo di inserimento delle nuove evidenze, che consenta al clinico di aggiungere una nuova osservazione (un referto, un esito di laboratorio, un'evoluzione sintomatologica) in punti di aggiornamento definiti del ciclo diagnostico, innescando automaticamente uno dei due meccanismi di aggiornamento delle ipotesi. Dal punto di vista dell'interazione, ciò richiederebbe l'introduzione di un nuovo componente nell'applicazione, ad esempio un campo di testo persistente accanto al pannello "*Reasoning Traceability",* collegato a una funzione che, al momento dell'inserimento, sospende il *thread*, inserisce la nuova informazione nello stato condiviso e innesca il processo di aggiornamento.

Una seconda direzione riguarda l'Approccio II. Il re-ranking attuale riordina le ipotesi sulla base dei nuovi esiti induttivi e dei quattro criteri del framework, senza una misura esplicita di quanto la nuova informazione modifichi ciascuna ipotesi. Si potrebbe introdurre un punteggio di concordanza analogo a quello dell'Approccio I (Sezione 4.2.2) come input diretto della fase di ranking, in modo da rendere lo spostamento di posizione più strutturato e quantificabile.

In parallelo l'interfaccia grafica potrebbe essere estesa per consentire non soltanto la modifica dell'ordine o l'eliminazione delle ipotesi già presenti nel ranking, ma anche l'inserimento manuale di nuove diagnosi. Questa funzionalità risponderebbe ad un limite del sistema attuale, ovvero che se una diagnosi rilevante dovesse essere stata omessa durante l'abduzione, l'utente non potrebbe correggere l'omissione, ma limitarsi a riordinare le ipotesi prodotte.

6.4.2 Potenziamento epistemologico tramite moduli RAG (*Retrieval-Augmented Generation*)

L'architettura descritta nel Capitolo 3 fa affidamento esclusivamente sulla conoscenza parametrica del modello linguistico sottostante: ogni fase, astrazione, abduzione, ranking, deduzione, induzione, opera unicamente sul testo clinico e sulle rappresentazioni intermedie prodotte dai nodi precedenti, senza alcun accesso a fonti esterne di conoscenza medica strutturata (linee guida cliniche, letteratura scientifica, banche dati nosologiche).

Un'estensione del framework consiste nell'integrazione di un modulo di *Retrieval-Augmented Generation* (RAG) [39], approccio che integra la generazione dell'LLM con informazioni recuperate da fonti esterne, in punti mirati dell'architettura. Il caso d'uso più immediato riguarda il nodo di

abduzione non focalizzata (Sezione 3.2.2): l'ampliamento dello spazio iniziale delle ipotesi potrebbe beneficiare del recupero di famiglie diagnostiche o sindromi da una base di conoscenza strutturata (ad esempio linee guida mediche), riducendo la dipendenza dalla sola conoscenza parametrica del modello e mitigando il rischio, discusso in Sezione 5.3.2, che una diagnosi rilevante non venga mai generata perché il modello non la associa spontaneamente al quadro clinico osservato.

Un secondo punto di applicazione riguarda il nodo di deduzione (Sezione 3.2.4): il recupero di linee guida cliniche specifiche per una data ipotesi potrebbe arricchire la generazione delle conseguenze attese con informazioni aggiornate, criteri diagnostici recenti o correlazioni non ancora consolidate nella conoscenza di addestramento del modello.

I contenuti recuperati dovrebbero essere mantenuti distinti dai reperti del paziente, conservando il riferimento alla fonte. La selezione dei passaggi pertinenti e un limite alla loro lunghezza permetterebbero inoltre di controllare il volume del contesto fornito al modello.

6.4.3 Infrastruttura di inferenza modello-indipendente tramite SGLang

Un possibile sviluppo del framework riguarda l'infrastruttura utilizzata per eseguire i modelli linguistici. Attualmente, la funzione di chiamata all'LLM trasmette lo schema previsto per ciascun nodo tramite il parametro `response_format` (Sezione 3.3). La validazione permette di controllare il risultato prodotto, ma la gestione dei vincoli durante la generazione dipende anche dal modello e dall'interfaccia utilizzata; per rendere questo controllo più uniforme, si potrebbe adottare SGLang [41], un'infrastruttura per l'esecuzione di modelli linguistici. Due funzionalità sono particolarmente utili per il framework: la decodifica vincolata e il riutilizzo dei calcoli relativi alle parti comuni delle richieste [40][41].

La decodifica vincolata limita la generazione alle sequenze compatibili con il formato richiesto. Quando un nodo deve restituire un oggetto JSON conforme a uno schema, il sistema impedisce la selezione di token che violerebbero i vincoli supportati. Il controllo interviene quindi durante la produzione della risposta, anziché affidarsi soltanto alle istruzioni fornite al modello [41]. Questa funzionalità potrebbe ridurre gli errori di formato e la necessità di correggere o ripetere le risposte. SGLang renderebbe quindi più uniforme lo scambio di dati tra i nodi, creando uno standard di output fisso e indirizzando il framework ad ottenere prestazioni simili tra LLM diversi.

La seconda funzionalità è *RadixAttention*, che riutilizza i risultati intermedi dei calcoli quando più richieste iniziano con la stessa sequenza di token. Questi risultati vengono conservati nella memoria temporanea del modello, denominata *KV cache*, evitando di elaborare nuovamente le parti comuni [40]. Nel framework, questo meccanismo potrebbe ridurre i tempi delle esecuzioni ripetute o delle richieste con istruzioni iniziali identiche. Il beneficio dipenderebbe dalla quantità di contenuto effettivamente condiviso.

L'integrazione richiederebbe un adattamento della funzione di chiamata all'LLM, affinché invii le richieste e gli schemi a un server SGLang. I nodi continuerebbero a utilizzare le strutture dati già definite, mentre l'infrastruttura gestirebbe i vincoli durante la generazione [41]. La validazione degli output resterebbe comunque necessaria.

6.4.4 Standardizzazione delle diagnosi

Un ulteriore sviluppo riguarda la standardizzazione dell'output mediante l'associazione delle ipotesi generate dal framework a codifiche cliniche standardizzate, in particolare alla classificazione ICD-10 [20]. L'attuale protocollo di valutazione confronta la diagnosi prodotta con un insieme di alias della diagnosi di riferimento, applicando un criterio di similarità lessicale (Sezione 5.1.2). Questo criterio è deterministico, riproducibile e indipendente da modelli di *embedding*, ma non garantisce una corrispondenza semantica precisa: formulazioni clinicamente equivalenti possono non superare la soglia, mentre espressioni lessicalmente vicine possono riferirsi a entità nosologiche diverse.

L'integrazione di un modulo di normalizzazione permetterebbe di associare ciascuna ipotesi prodotta dal framework a uno o più codici ICD-10 [20] o a concetti di terminologie cliniche come SNOMED CT [32], eventualmente utilizzando risorse di integrazione terminologica come UMLS [19]. Questo passaggio dovrebbe essere concepito come una fase di validazione, non come una semplice mappatura lessicale automatica: in presenza di più candidati plausibili, il sistema potrebbe presentare all'operatore le alternative ordinate per similarità semantica, contesto clinico e specificità diagnostica, lasciando all'utente la conferma finale. Una codifica standardizzata renderebbe inoltre le metriche di accuratezza più robuste e confrontabili, riducendo la dipendenza da alias testuali e dalla soglia di similarità adottata.

6.4.5 Validazione prospettica in ambienti di simulazione clinica

Le valutazioni presentate nel Capitolo 5 sono state ottenute su casi clinici retrospettivi tratti dal *benchmark* CPC-Bench e dal dataset SCT (Sezione 5.1.1), entrambi costruiti a partire da casi clinici disponibili pubblicamente. Come segnalato nella stessa sezione, questo comporta un limite metodologico non trascurabile: non è possibile escludere che alcuni di questi casi, o presentazioni clinicamente molto simili, siano stati incontrati dai modelli durante la fase di addestramento, un rischio di contaminazione comune a gran parte della letteratura che utilizza casi clinici pubblici per la valutazione di LLM.

Una direzione di sviluppo necessaria consiste quindi nella validazione del framework in un ambiente di simulazione clinica, nel quale i casi non siano tratti da fonti pubbliche, ad esempio tramite scenari simulati con pazienti standardizzati, o attraverso la collaborazione con strutture cliniche per l'anonimizzazione e l'impiego di casi reali non ancora pubblicati. Questo tipo di validazione ridurrebbe il rischio di valutare risposte influenzate dalla precedente esposizione agli stessi casi.

Un secondo aspetto di questa validazione riguarderebbe l'estensione della numerosità campionaria: come discusso ripetutamente nelle Sezioni 5.2 e 5.3, i campioni utilizzati sono limitati e non consentono di raggiungere una potenza statistica sufficiente per rilevare differenze di piccola entità. Un ambiente di simulazione prospettica, se progettato con un protocollo di raccolta dati continuativo, offrirebbe l'opportunità di accumulare progressivamente un campione più ampio, superando i limiti di potenza statistica evidenziati nella Sezione 5.3.1.

Infine, la validazione in ambiente simulato consentirebbe di testare l'interfaccia grafica in condizioni d'uso più realistiche, con clinici che interagiscono con il sistema in tempo reale.

6.4.6 Estensione del framework ai compiti di pianificazione terapeutica e di follow-up

L'ST-Model, descritto nella Sezione 2.2, è stato concepito come un quadro epistemologico generico, applicabile non solo alla diagnosi ma anche a due ulteriori compiti fondamentali del ragionamento medico: la pianificazione terapeutica, in cui il ciclo parte dalla diagnosi e dai dati osservati per abdurre terapie candidate, ordinarle per priorità e testarne le conseguenze attese, e il monitoraggio, in cui il ciclo verifica se la terapia in corso sta producendo gli effetti attesi, eventualmente innescando nuovi cicli di diagnosi o pianificazione.

L'implementazione descritta nei Capitoli 3 e 4 si è concentrata esclusivamente sul compito diagnostico: i nodi dell'architettura sono stati progettati e validati unicamente per la generazione e la verifica di ipotesi diagnostiche. Un'estensione del framework consisterebbe nell'implementare i due compiti rimanenti riutilizzando la medesima struttura a grafo, con adattamenti mirati agli schemi di output.

Il compito di monitoraggio, in particolare, presenta un'affinità diretta con il ciclo di retroazione post-induzione discusso nella Sezione 6.2 e con gli Approcci II e III del Capitolo 4.2: un ciclo di monitoraggio richiede l'acquisizione periodica di nuove osservazioni (esami di controllo, evoluzione dei sintomi) e la loro integrazione in un giudizio aggiornato sull'efficacia della terapia in corso, una struttura con elementi comuni a quella già esplorata sperimentalmente per l'aggiornamento delle ipotesi diagnostiche in risposta a nuove evidenze cliniche. Questo suggerisce che potrebbe essere riutilizzata, con modifiche contenute, come base per l'implementazione di questa nuova funzionalità.

Le procedure sperimentali di aggiornamento delle ipotesi potrebbero offrire un punto di partenza, ma il loro impiego nel monitoraggio richiederebbe una valutazione dedicata.

CAPITOLO 7: Conclusioni

Questa tesi affronta alcuni limiti che ostacolano l'impiego clinico degli LLM anche quando le loro prestazioni diagnostiche sono elevate: la difficoltà di ricostruire il processo che porta alla risposta, l'eccessiva sicurezza delle valutazioni e gli errori di omissione. Il lavoro ha esteso un framework che traduce il *Select-and-Test Model* di Ramoni et al. in un grafo a stati orchestrato con LangGraph, dove ogni fase inferenziale è un nodo ispezionabile con output vincolato da schemi definiti. L'estensione ha riguardato tre aspetti: l'ottimizzazione del nodo di astrazione, l'integrazione di nuove evidenze cliniche e lo sviluppo di un'interfaccia con supervisione umana (Capitolo 4).

Il primo contributo riguarda il nodo di astrazione. L'ispezione preliminare dei log ha evidenziato un'aggregazione eccessiva di condizioni pregresse e farmaci in voci generiche. Le modifiche al prompt e al campo di spiegazione hanno permesso di rappresentare separatamente questi elementi nei casi esaminati.

La valutazione condotta sui 10 casi del CPC-Bench ha mostrato una riduzione del 30.3% del numero medio di caratteri dell'output di astrazione. Questa riduzione si accompagna però a un calo dell'accuratezza Top-1 dal 47% al 36%, mentre la Top-10 aumenta dal 72% al 74%. Il risultato indica quindi un output più breve, ma non un miglioramento generale delle prestazioni diagnostiche. La lieve variazione della Top-10 non compensa automaticamente il peggioramento nelle prime posizioni.

Il secondo contributo riguarda l'aggiornamento del giudizio in presenza di nuove evidenze. Nell'Approccio I, l'aggiunta delle aspettative deduttive e degli esiti induttivi al prompt è associata a un aumento della concordanza SCT media di 2.28 punti percentuali per Gemini 2.5 Pro e di 3.15 per Gemini 2.5 Flash. Le differenze non sono statisticamente significative e non dimostrano quindi la superiorità dell'approccio *framework-aware*. Gli Approcci II e III hanno esplorato rispettivamente il riordinamento delle ipotesi esistenti e la riesecuzione del ciclo completo. Nell'unico caso clinico con omissioni iniziali, l'Approccio III non ha recuperato la diagnosi nelle esecuzioni interessate. Tuttavia, entrambi hanno sempre preservato una diagnosi già inclusa nel ranking al primo ciclo. Nel sottoinsieme incluso nell'analisi del rango, l'Approccio III ha mostrato una maggiore dispersione delle variazioni di posizione rispetto all'Approccio II.

Il terzo contributo è l'interfaccia grafica. Questa realizza concretamente il principio *Human-in-the-Loop* con due modalità di sospensione, la modifica del primo ranking da parte del clinico, la distinzione visiva tra correzione umana e aggiornamento algoritmico, la possibilità di ispezionare gli output dei nodi e il salvataggio dello stato.

Il lavoro non dimostra che il framework superi un LLM usato in modalità *zero-shot* sul piano delle prestazioni diagnostiche. Mostra invece la possibilità di rendere il ragionamento di un LLM esplicito, tracciabile e supervisionabile, e quantifica i compromessi di questa scelta. Questi risultati vanno interpretati alla luce dei limiti discussi nella Sezione 6.2: campioni di numerosità ridotta, un solo caso con omissione reale, separazione manuale delle evidenze, un solo modello per il confronto tra gli Approcci II e III e il rischio di contaminazione dei *benchmark* pubblici.

Le direzioni di sviluppo, descritte nella Sezione 6.4, discendono direttamente dai limiti emersi:

- l'integrazione dell'inserimento incrementale di nuove evidenze nell'interfaccia;
- l'integrazione di moduli RAG per integrare conoscenze esterne nelle fasi di generazione e verifica delle ipotesi;
- l'adozione di un'infrastruttura per rendere la conformità agli schemi indipendente dal modello;
- la validazione prospettica su casi non pubblicati, in ambienti di simulazione clinica;
- l'estensione del framework alla pianificazione terapeutica e al monitoraggio, già previsti dal modello di Ramoni et al.

In conclusione, il framework non è proposto come sostituto del giudizio clinico, ma come strumento sperimentale per organizzare e ispezionare gli output degli LLM e consentire interventi umani sul processo. Il suo eventuale impiego clinico richiede ulteriori verifiche di accuratezza, affidabilità e utilità per i professionisti.

Bibliografia

1. Shortliffe EH, Davis R, Axline SG, Buchanan BG, Green CC, Cohen SN. *Computer-based consultations in clinical therapeutics: explanation and rule acquisition capabilities of the MYCIN system.* Computers and Biomedical Research. 1975;8(4):303–320. doi:10.1016/0010-4809(75)90032-9.

2. Miller RA, Pople HE Jr, Myers JD. *Internist-I, an experimental computer-based diagnostic consultant for general internal medicine. New England Journal of Medicine.* 1982;307(8):468–476. doi:10.1056/NEJM198208193070803.

3. Ramoni M, Stefanelli M, Magnani L, Barosi G. *An epistemological framework for medical knowledge-based systems.* IEEE Transactions on Systems, Man, and Cybernetics. 1992;22(6):1361–1375. doi:10.1109/21.199462.

4. Patel VL, Arocha JF, Zhang J. Thinking and reasoning in medicine. In: Holyoak KJ, Morrison RG, editors. *The Cambridge Handbook of Thinking and Reasoning*. Cambridge, UK: Cambridge University Press; 2005. p. 727–750.

5. Thirunavukarasu AJ, Ting DSJ, Elangovan K, Gutierrez L, Tan TF, Ting DSW. *Large language models in medicine.* Nature Medicine. 2023;29(8):1930–1940. doi:10.1038/s41591-023-02448-8.

6. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser Ł, Polosukhin I. *Attention is all you need.* Advances in Neural Information Processing Systems. 2017;30:5998–6008.

7. Devlin J, Chang MW, Lee K, Toutanova K. BERT: p*re-training of deep bidirectional transformers for language understanding*. In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Minneapolis, MN: Association for Computational Linguistics; 2019. p. 4171–4186. doi:10.18653/v1/N19-1423.

8. Radford A, Narasimhan K, Salimans T, Sutskever I. *Improving language understanding by generative pre-training.* OpenAI; 2018. Disponibile su: https://www.cs.ubc.ca/~amuham01/LING530/papers/radford2018improving.pdf.

9. Brown T, Mann B, Ryder N, Subbiah M, Kaplan JD, Dhariwal P, Neelakantan A, Shyam P, Sastry G, Askell A, Agarwal S, Herbert-Voss A, Krueger G, Henighan T, Child R, Ramesh A, Ziegler D, Wu J, Winter C, Hesse C, Chen M, Sigler E, Litwin M, Gray S, Chess B, Clark J, Berner C, McCandlish S, Radford A, Sutskever I, Amodei D. *Language models are few-shot learners*. In: *Advances in Neural Information Processing Systems 33* (NeurIPS 2020). Curran Associates, Inc.; 2020.

10. Brodeur PG, Buckley TA, Kanjee Z, Goh E, Bin Ling E, Jain P, Cabral S, Abdulnour RE, Haimovich AD, Freed JA, Olson A, Morgan DJ, Hom J, Gallo R, McCoy LG, Mombini H, Lucas C, Fotoohi M, Gwiazdon M, Restifo D, Restrepo D, Horvitz E. *Superhuman performance of a large language model on the reasoning tasks of a physician*. arXiv preprint arXiv:2412.10849; 2024.

11. Wu D, Nateghi Haredasht F, Kumar Maharaj S, Jain P, Tran J, Gwiazdon M, et al. First, do NOHARM: *towards clinically safe large language models* [preprint]. arXiv; 2025. arXiv:2512.01241. doi:10.48550/arXiv.2512.01241.

12. Ilievski F, Hammer B, van Harmelen F, Paassen B, Saralajew S, Schmid U, Bolognesi M, Dong XL, Gashteovski K, Hellendoorn V, et al. *Aligning generalisation between humans and machines.* Nature Machine Intelligence. 2025;7(9):1378–1389. doi:10.1038/s42256-025-01109-4.

13. Charlin B, Roy L, Brailovsky C, Goulet F, van der Vleuten C. *The Script Concordance Test: a tool to assess the reflective clinician.* Teaching and Learning in Medicine. 2000;12(4):189–195. doi:10.1207/S15328015TLM1204_5.

14. McCoy LG, Swamy R, Sagar N, Wang M, Bacchi S, Fong JMN, Tan NCK, Tan K, Buckley TA, Brodeur PG, Celi LA, Manrai AK, Humbert A, Rodman A. *Assessment of large language models in clinical reasoning: a novel benchmarking study.* NEJM AI. 2025;2(10):AIdbp2500120. doi:10.1056/AIdbp2500120.

15. Rancati S, Bergomi L, Parimbelli E, Nicora G, Bellazzi R. *Epistemologically guided LLM reasoning for differential diagnosis.* In: *Proceedings of the 2026 Conference on Artificial Intelligence in Medicine (AIME 2026). Lecture Notes in Computer Science*. Cham: Springer; 2026. p. 115–124. doi:10.1007/978-3-032-30710-1_14.

16. Peirce CS. *Deduction, induction, and hypothesis*. Popular Science Monthly. 1878;13:470–482.

17. Buckley TA, Conci R, Brodeur PG, Gusdorf J, Beltrán S, Behrouzi B, Crowe B, Dockterman J, Muhammad M, Ohnigian S, Sanchez A, Diao JA, Shah AP, Restrepo D, Rosenberg ES, Lea AS, Zitnik M, Podolsky SH, Kanjee Z, Abdulnour REE, Koshy JM, Rodman A, Manrai AK. *Advancing medical artificial intelligence using a century of cases: CPC-Bench and Dr. CaBot.* arXiv preprint arXiv:2509.12194; 2025.

18. Yazdani S, Hoseini Abardeh M. *Five decades of research and theorization on clinical reasoning: a critical review. Advances in Medical Education and Practice.* 2019;10:703–716. doi:10.2147/AMEP.S213492.

19. National Library of Medicine. Unified Medical Language System (UMLS) Terminology Services. Bethesda, MD: U.S. National Library of Medicine. Disponibile su: https://uts.nlm.nih.gov/. Consultato il 10 ottobre 2026.

20. World Health Organization. International Statistical Classification of Diseases and Related Health Problems, 10th Revision (ICD-10). Geneva: World Health Organization; 2019. Disponibile su: https://icd.who.int/browse10/. Consultato il 10 ottobre 2026.

21. SCT-Bench. *sctpublic: an open-source framework for evaluating clinical reasoning in large language models using Script Concordance Tests* [software]. GitHub; 2026. Disponibile su: https://github.com/SCT-Bench/sctpublic. Consultato il 10 ottobre 2026.

22. SCT-Bench. SCT-Bench — Script Concordance Testing for evaluating clinical reasoning in large language models [Internet]. Disponibile su: https://www.concor.dance/. Consultato il 10 ottobre 2026.

23. LangChain Inc. LangGraph overview. LangChain Documentation; 2026. Consultato il 7 ottobre 2026. Disponibile su: https://docs.langchain.com/oss/python/langgraph/overview.

24. *Pydantic Documentation — Concepts: Models*. Disponibile su: https://docs.pydantic.dev/latest/concepts/models/ . Consultato il 12 ottobre 2026.

25. Ceusters W, Elkin P, Smith B. *Negative findings in electronic health records and biomedical ontologies: a realist approach. International Journal of Medical Informatics*. 2007;76(Suppl 3):S326–S333. doi:10.1016/j.ijmedinf.2007.02.003.

26. Xia K, Bareinboim E. *Causal abstraction inference under lossy representations*. In: Proceedings of the 42nd International Conference on Machine Learning (ICML 2025). PMLR; 2025.

27. Croskerry P. *The importance of cognitive errors in diagnosis and strategies to minimize them.* Academic Medicine. 2003;78(8):775–780. doi:10.1097/00001888-200308000-00003.

28. *Dash Documentation*. Plotly; Disponibile su: https://dash.plotly.com/ . Consultato il: 9 ottobre 2026.

29. Santoro, M., Vaccari, L., Mavridis, D., Smith, R., Posada Sanchez, M., & Gattwinkel, D. (2019). *Web Application Programming Interfaces (APIs): general-purpose standards, terms and European Commission initiatives.* Publications Office of the European Union, JRC118082. DOI: 10.2760/675.

30. Google. *A new era of intelligence with Gemini 3*. Google Blog; November 18, 2025. Disponibile su: https://blog.google/products-and-platforms/products/gemini/gemini-3/

31. Google Cloud. Gemini 2.5 updates: Flash/Pro GA, SFT, Flash-Lite on Vertex AI. Google Cloud Blog; June 17, 2025. Disponibile su: https://cloud.google.com/blog/products/ai-machine-learning/gemini-2-5-flash-lite-flash-pro-ga-vertex-ai

32. National Library of Medicine. SNOMED CT. Bethesda, MD: U.S. National Library of Medicine. Disponibile su: https://www.nlm.nih.gov/healthit/snomedct/index.html. Consultato il 10 ottobre 2026.

33. *Python Software Foundation. difflib — helpers for computing deltas.* Python 3 Documentation. Disponibile su: https://docs.python.org/3/library/difflib.html. Consultato il 10 ottobre 2026.

34. Ratcliff JW, Metzener DE. *Pattern matching: The Gestalt approach. Dr. Dobb's Journal.* 1988;13(7):46.

35. Jaccard P. *The distribution of the flora in the alpine zone*. New Phytologist. 1912;11(2):37–50. doi:10.1111/j.1469-8137.1912.tb05611.x.

36. Ma T, Guo L, Yan H, Wang L. *Cobind: quantitative analysis of the genomic overlaps.* Bioinform Adv. 2023 Aug 7;3(1):vbad104. doi: 10.1093/bioadv/vbad104. PMID: 37600846; PMCID: PMC10438957.

37. Reimers N, Gurevych I. *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 (EMNLP-IJCNLP)*. Hong Kong: Association for Computational Linguistics; 2019. p. 3982–3992. doi:10.18653/v1/D19-1410.

38. OpenAI. Key concepts: Tokens. OpenAI API Documentation. Disponibile su: https://developers.openai.com/api/docs/concepts. Consultato il 14 ottobre 2026.

39. Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W., Rocktäschel, T., Riedel, S., & Kiela, D. (2020). *Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. Advances in Neural Information Processing Systems*, 33.

40. Zheng L, Yin L, Xie Z, Sun C, Huang J, Yu CH, Cao S, Kozyrakis C, Stoica I, Gonzalez JE, Barrett C, Sheng Y. *SGLang: efficient execution of structured language model programs*. arXiv preprint arXiv:2312.07104; 2023. doi:10.48550/arXiv.2312.07104.

41. SGLang. *Structured outputs.* SGLang Documentation; 2026. Consultato a settembre 2026. Disponibile su: https://docs.sglang.ai/advanced_features/structured_outputs.html