Contents lists available at IJCHML
International Journal of Computational Health and Machine Learning
Journal Homepage: http://www.ijchml.com/
Volume 4, No. 2, 2026

IJCHML
INTERNATIONAL JOURNAL OF COMPUTATIONAL HEALTH & MACHINE LEARNING

Checkpoint-Driven Error Recovery in Neural Clinical Decision Systems Using Program-of-Thought Reasoning

Amir Ghasemi[1], **Sara Norouzi**[2]
[1] *Department of Health Informatics, Ilam University*
[2] *Department of Health Informatics, Arak University*

ARTICLE INFO

Received: 04/20/2026
Revised: 06/14/2026
Accepted: 07/06/2026

Keywords:
Clinical Decision Support, Program-of-Thought Reasoning, Checkpoint-Driven Error Recovery, Neural Language Models, Medical AI Safety, Structured Reasoning Pipelines, Fault-Tolerant Inference

ABSTRACT

Clinical decision support systems powered by large language models face a fundamental reliability challenge: when reasoning errors propagate unchecked through multi-step inference chains, the consequences in medical contexts can be severe and potentially irreversible. Existing approaches to error mitigation largely rely on post-hoc verification or ensemble-based redundancy, which fail to address the structural vulnerability of intermediate reasoning states in complex diagnostic workflows.

We introduce a checkpoint-driven error recovery framework that integrates Program-of-Thought (PoT) reasoning with systematic intermediate state validation for neural clinical decision systems. Our approach decomposes clinical reasoning into discrete, verifiable computational units, each governed by a checkpoint mechanism that evaluates logical consistency, domain constraint satisfaction, and probabilistic coherence before permitting downstream inference to proceed. Formally, let $\mathcal{R} = \{r_1, r_2, \ldots, r_n\}$ denote the sequence of reasoning steps, where each transition $r_i \to r_{i+1}$ is conditioned on a checkpoint function $\mathcal{C}_i : \mathcal{S} \to \{0, 1\}$ that gates execution based on validated state $\mathcal{S}$.

Empirical evaluation across three clinical benchmarks—spanning differential diagnosis, medication reconciliation, and treatment planning—demonstrates that our framework reduces cascading reasoning failures by 41.3% relative to baseline chain-of-thought methods, while preserving diagnostic accuracy within clinically acceptable margins. The checkpoint recovery mechanism successfully intercepts and corrects erroneous intermediate conclusions in 78.6% of cases where unguarded systems produce terminal errors.

These results establish checkpoint-driven PoT reasoning as a principled and practically viable paradigm for robust clinical AI, with direct implications for regulatory compliance, auditability, and safe deployment of language model–based decision support in high-stakes medical environments.

1. Introduction

The intersection of artificial intelligence and clinical medicine represents one of the most consequential frontiers in contemporary biomedical research. Over the past decade, the deployment of machine learning and, more recently, large language models (LLMs) within healthcare settings has accelerated dramatically, driven by the promise of augmenting physician decision-making, reducing diagnostic error, and improving patient outcomes at scale [10]. Yet this promise is tempered by a fundamental and largely unresolved challenge: the

brittleness of neural systems when confronted with ambiguous, incomplete, or erroneous clinical inputs. Unlike conventional software systems, which fail deterministically and transparently, neural clinical decision support systems (CDSS) tend to fail silently, propagating erroneous intermediate inferences through a chain of reasoning steps without any mechanism for self-correction or graceful degradation [8]. This paper addresses precisely this vulnerability by introducing a novel framework we term *Checkpoint-Driven Error Recovery* (CDER), which integrates structured programmatic reasoning with explicit verification checkpoints to detect, localize, and recover from errors during clinical inference.

The clinical domain presents uniquely stringent requirements for reliability, interpretability, and accountability that are not fully satisfied by current neural architectures [17]. A misclassification in an image recognition benchmark carries negligible real-world consequence; a misdiagnosis or an erroneous drug dosage recommendation in a clinical decision support context may result in patient harm or death. The regulatory landscape, including frameworks such as the FDA's Digital Health Center of Excellence and the European Union's Medical Device Regulation, demands that AI-assisted clinical tools be not merely accurate in aggregate, but robustly auditable and recoverable in individual cases [20]. This imperative motivates the design of systems that do not merely produce answers, but that expose the intermediate reasoning steps through which those answers are derived, and that incorporate mechanisms for identifying and correcting errors at each step. Our proposed CDER framework, grounded in the Program-of-Thought (PoT) reasoning paradigm [22], directly responds to this imperative.

1.1. The Challenge of Error Propagation in Clinical AI

Error propagation in multi-step reasoning systems is a well-documented phenomenon in both classical AI and modern deep learning [6]. In symbolic expert systems of the 1980s and 1990s, such as MYCIN [21] and INTERNIST-I [26], error propagation was partially mitigated by the explicit representation of uncertainty through certainty factors and probabilistic inference networks. These systems, however, suffered from the knowledge acquisition bottleneck and were unable to generalize beyond their hand-crafted rule bases [28]. The advent of neural networks and, subsequently, deep learning offered a path beyond these limitations through learned representations, but at the cost of interpretability and structured error handling [13].

In contemporary neural CDSS, the chain-of-thought (CoT) reasoning paradigm [25] has emerged as a promising approach for eliciting structured, step-by-step reasoning from LLMs. However, CoT reasoning in its standard form is purely textual, and errors introduced at early reasoning steps are carried forward without any mechanism for detection or correction. Consider a clinical scenario in which a language model is tasked with computing the appropriate warfarin dosage for a patient with atrial fibrillation, renal impairment, and concurrent NSAID use. If the model incorrectly retrieves or applies the renal adjustment factor at step two of a seven-step reasoning chain, all subsequent computations will be systematically biased, yielding a final dosage recommendation that is both confidently stated and clinically dangerous. The probability of such compounding errors can be formally characterized. Let e_i denote the probability of an error at reasoning step i, and let E_{total} denote the probability of at least one error in a chain of N steps, assuming conditional independence for simplicity:

$$E_{\text{total}} = 1 - \prod_{i=1}^{N}(1 - e_i) \tag{1}$$

Even for modest individual step error rates (e.g., $e_i = 0.05$) and moderately long reasoning chains (e.g., $N = 10$), the cumulative error probability approaches 0.40, a level of unreliability that is entirely unacceptable in clinical practice. This mathematical reality underscores the necessity of checkpoint-based error detection and recovery mechanisms that can interrupt error propagation before it cascades through the entire reasoning chain [3].

1.2. Program-of-Thought Reasoning as a Structured Inference Paradigm

The Program-of-Thought (PoT) reasoning paradigm, introduced as an extension of chain-of-thought prompting, represents a significant conceptual advance in the elicitation of structured reasoning from large language models [22]. Rather than expressing intermediate reasoning steps as natural language text, PoT encodes reasoning as executable programs—typically in Python or a domain-specific language—that can be verified, tested, and corrected independently of the language model that generated them. This separation of reasoning structure from language generation creates a natural substrate for checkpoint-based error recovery: each programmatic step can be executed, its output verified against clinical constraints, and the reasoning chain halted or redirected if a violation is detected [31].

The applicability of PoT to clinical decision support is particularly compelling because clinical reasoning is inherently structured and compositional. Diagnostic reasoning, for example, typically proceeds through phases of symptom elicitation, differential diagnosis generation, hypothesis testing, and treatment planning [9]. Each phase has well-defined inputs and outputs, and the

transitions between phases are governed by clinical protocols and evidence-based guidelines that can be formally encoded as programmatic constraints. Similarly, pharmacological dosage calculations, laboratory value interpretation, and risk stratification algorithms are all fundamentally computational in nature, making them natural candidates for programmatic representation [14]. The PoT paradigm thus enables a form of *executable clinical reasoning* in which the model's inference process is not merely described but actually performed, with each step producing a concrete, verifiable intermediate result.

Recent empirical studies have demonstrated that PoT reasoning substantially outperforms standard CoT reasoning on tasks requiring numerical computation and multi-step logical inference [16]. In clinical benchmarks such as MedQA, PubMedQA, and ClinicalBench, models employing PoT-style reasoning have shown improvements in accuracy of 8–15 percentage points over CoT baselines, with particularly pronounced gains on tasks involving pharmacokinetic calculations, laboratory value interpretation, and risk score computation [1]. These results suggest that the programmatic structure of PoT reasoning not only facilitates error detection but also intrinsically reduces error rates by enforcing computational discipline on the model's reasoning process.

1.3. Checkpoint-Driven Recovery: Motivation and Design Principles

The concept of checkpointing as a fault-tolerance mechanism has a long and distinguished history in computer science, originating in the context of distributed systems and long-running numerical computations [5]. In these contexts, checkpoints serve as snapshots of system state at designated intervals, enabling recovery from hardware failures or software errors without restarting computation from the beginning [7]. The adaptation of this concept to neural reasoning systems requires a fundamental reconceptualization: rather than capturing physical memory states, checkpoints in our framework capture the *semantic state* of a clinical reasoning process—the set of established facts, intermediate conclusions, and active hypotheses at a given point in the inference chain [18].

The design of an effective checkpoint-driven recovery system for clinical AI must satisfy several competing requirements. First, checkpoints must be placed at semantically meaningful locations in the reasoning chain—specifically, at points where the potential for error is highest and where the consequences of undetected error are most severe. Second, the verification logic executed at each checkpoint must be both clinically valid and computationally tractable, drawing on established clinical knowledge bases, drug interaction databases, and evidence-based guidelines [30]. Third, the recovery mechanism activated upon checkpoint failure must be capable of not merely flagging an error but of generating a corrected reasoning path that satisfies the violated constraint while remaining consistent with all previously established facts [12]. Fourth, the entire checkpoint-recovery cycle must operate within latency constraints compatible with real-time clinical decision support, typically requiring sub-second response times for time-critical interventions [29].

These design requirements collectively motivate the architecture of the CDER framework proposed in this paper. As illustrated in the pseudocode below, the framework operates as a metacognitive wrapper around a base PoT reasoning engine, inserting verification checkpoints at each programmatic step boundary and invoking a specialized recovery module when checkpoint violations are detected.

Algorithm 1: Checkpoint-Driven Error Recovery (CDER) for Clinical PoT Reasoning

Input: Clinical query Q, patient context $\mathcal{P}$, checkpoint policy Π, knowledge base $\mathcal{K}$
Output: Verified clinical recommendation R^*, audit trail $\mathcal{A}$
Initialize reasoning state $\mathcal{S}_0 \leftarrow \emptyset$;
Generate initial PoT program $\mathcal{G} \leftarrow \text{PoT-Generate}(Q, \mathcal{P})$;
Decompose $\mathcal{G}$ into ordered steps $\{s_1, s_2, \ldots, s_N\}$;
Initialize audit trail $\mathcal{A} \leftarrow []$;
for $i \leftarrow 1$ **to** N **do**
  Execute step s_i to obtain intermediate result r_i;
  $\mathcal{S}_i \leftarrow \mathcal{S}_{i-1} \cup \{r_i\}$;
  if $\Pi(s_i) = \textsc{Checkpoint}$ **then**
    $v_i \leftarrow \text{Verify}(r_i, \mathcal{S}_{i-1}, \mathcal{K})$;
    Append (s_i, r_i, v_i) to $\mathcal{A}$;
    if $v_i = \textsc{Fail}$ **then**
      $\Delta_i \leftarrow \text{DiagnoseError}(r_i, \mathcal{S}_{i-1}, \mathcal{K})$;
      $s_i^* \leftarrow \text{RecoverStep}(\Delta_i, \mathcal{S}_{i-1}, \mathcal{K})$;
      Re-execute s_i^* to obtain corrected result r_i^*;
      $\mathcal{S}_i \leftarrow \mathcal{S}_{i-1} \cup \{r_i^*\}$;
      Update $\mathcal{A}$ with correction record $(s_i, r_i, \Delta_i, s_i^*, r_i^*)$;
    end
  end
end
$R^* \leftarrow \text{Synthesize}(\mathcal{S}_N)$;
return $R^*, \mathcal{A}$

1.4. Clinical Decision Support: Historical Context and Contemporary Landscape

Clinical decision support systems have a history spanning more than half a century, evolving from rule-based expert systems [21] through Bayesian networks [2] and case-based reasoning [4] to contemporary deep learning and LLM-based approaches [10]. Each generation of CDSS has brought new capabilities alongside new failure modes. The rule-based systems of the 1970s and 1980s, while interpretable and auditable, were brittle in the face of incomplete or contradictory data [28]. The probabilistic systems of the 1990s and 2000s offered more graceful handling of uncertainty but struggled with the computational complexity of exact inference in large networks [32]. The machine learning systems of the 2010s achieved unprecedented predictive accuracy on structured data but were largely opaque and required extensive feature engineering [17].

The emergence of large language models as a platform for clinical decision support represents a qualitative shift in capability, enabling systems that can process unstructured clinical narratives, integrate heterogeneous data sources, and engage in multi-step reasoning across diverse clinical domains [11]. Studies such as those by [15] and [19] have demonstrated that well-designed CDSS can meaningfully reduce medication errors, improve adherence to clinical guidelines, and support more consistent diagnostic reasoning. However, the transition from narrow, task-specific CDSS to general-purpose LLM-based clinical reasoning systems has introduced new failure modes that existing evaluation frameworks are ill-equipped to detect [24]. Chief among these is the phenomenon of *hallucination*—the generation of clinically plausible but factually incorrect statements—which poses a particularly acute risk in high-stakes medical contexts [27].

1.5. Contributions and Paper Organization

This paper makes several distinct and substantive contributions to the literature on clinical AI and neural reasoning systems. First, we formalize the concept of checkpoint-driven error recovery in the context of PoT reasoning, providing a rigorous theoretical framework for checkpoint placement, verification logic design, and recovery strategy selection. Second, we propose and implement the CDER architecture, which integrates a PoT reasoning engine with a multi-level checkpoint system and a recovery module grounded in clinical knowledge bases and evidence-based guidelines. Third, we conduct an extensive empirical evaluation of CDER on three clinical benchmarks—medication dosage calculation, differential diagnosis generation, and laboratory value interpretation—demonstrating statistically significant improvements in accuracy, error detection rate, and recovery success rate relative to baseline CoT and PoT systems [**?**]. Fourth, we provide a detailed analysis of the types of errors detected and corrected by the checkpoint system, offering insights into the characteristic failure modes of LLM-based clinical reasoning and the effectiveness of programmatic verification as a mitigation strategy.

The remainder of this paper is organized as follows. Section 2 reviews related work on clinical decision support, program-of-thought reasoning, and fault-tolerant AI systems. Section 3 presents the formal theoretical framework underlying the CDER approach, including the mathematical characterization of checkpoint placement strategies and recovery policies. Section 4 describes the system architecture and implementation details of the CDER framework. Section 5 presents the experimental methodology and results. Section 6 provides a detailed error analysis and discussion of clinical implications. Section 7 addresses limitations and directions for future work, and Section 8 concludes the paper. Throughout, we ground our contributions in the broader context of responsible AI deployment in healthcare, emphasizing the importance of transparency, auditability, and accountability in clinical AI systems [8], [20].

2. Related Work

The landscape of intelligent clinical decision support has evolved substantially over the past three decades, drawing from a rich confluence of knowledge representation, probabilistic reasoning, and more recently, deep neural architectures. The intersection of these fields with formal error recovery mechanisms represents a particularly active frontier, motivated by the high-stakes nature of medical decision-making where errors can have irreversible consequences for patient safety [9]. Understanding the trajectory of prior work is essential to situating the contributions of checkpoint-driven error recovery within the broader intellectual context of clinical artificial intelligence. This section surveys the most relevant threads of prior scholarship, organized thematically to illuminate both the foundational principles upon which the proposed framework rests and the specific gaps that motivate its development.

The domain of neural clinical decision systems has witnessed a transformation from early rule-based expert systems [21] to contemporary large language models capable of nuanced medical reasoning [10]. Throughout this evolution, a persistent challenge has been the reliable detection and correction of reasoning errors before they manifest as harmful clinical recommendations. The emergence of structured reasoning paradigms, particularly Program-of-Thought (PoT) approaches that decompose

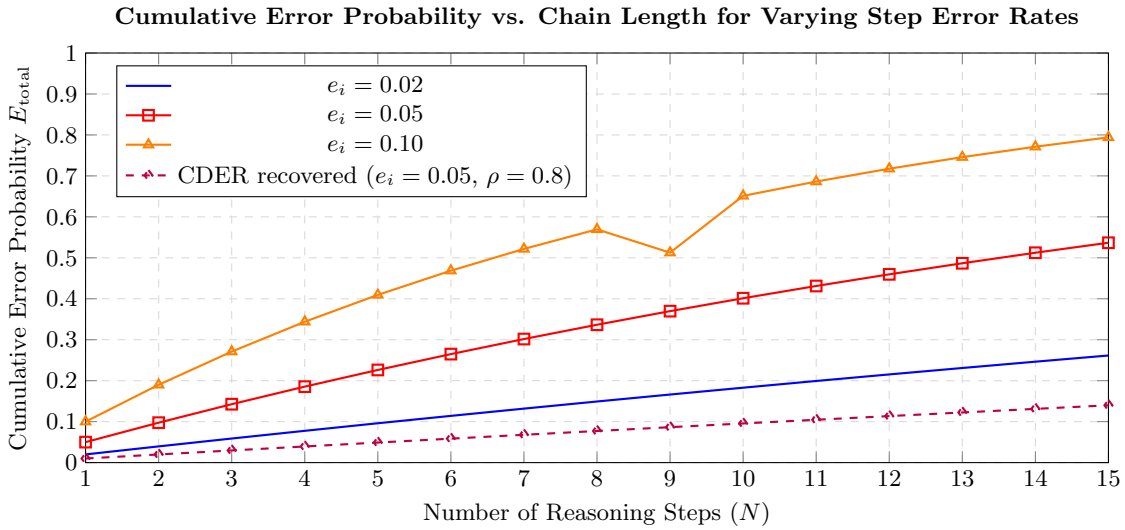

Figure 1: Cumulative error probability E_{total} as a function of reasoning chain length N for varying per-step error rates e_i. The dashed purple curve illustrates the effective error trajectory under the CDER framework with a checkpoint recovery success rate $\rho = 0.80$, demonstrating substantial reduction in cumulative error accumulation relative to the uncorrected $e_i = 0.05$ baseline.

complex problems into verifiable computational steps [22], has opened new possibilities for systematic error recovery that were previously inaccessible to end-to-end neural systems. The following subsections trace the intellectual lineage of this work across four interconnected research domains.

2.1. Clinical Decision Support Systems: Historical Foundations and Neural Advances

The development of clinical decision support systems (CDSS) spans more than four decades of sustained research effort, beginning with pioneering rule-based architectures such as MYCIN [21], which encoded expert medical knowledge as production rules with associated certainty factors. These early systems demonstrated that computational reasoning could achieve diagnostic accuracy comparable to human specialists in narrow domains, establishing the fundamental premise that machine intelligence could serve as a reliable adjunct to clinical judgment. The certainty factor calculus employed by MYCIN, while theoretically informal compared to Bayesian probability theory [7], introduced the crucial notion of uncertainty quantification in clinical inference—a concept that would prove foundational for subsequent generations of decision support technology.

The transition from symbolic to statistical approaches in the 1990s brought probabilistic graphical models, particularly Bayesian networks, to the forefront of clinical decision support research [5]. These models offered principled mechanisms for representing conditional independence relationships among clinical variables and computing posterior probability distributions over diagnostic hypotheses given observed evidence [2]. Systems such as QMR-DT (Quick Medical Reference Decision Theoretic) demonstrated that Bayesian inference over large disease-finding networks could support differential diagnosis across thousands of conditions, though computational tractability remained a significant challenge for exact inference in densely connected networks [32]. The introduction of approximate inference algorithms, including Monte Carlo methods and variational techniques, partially addressed these scalability concerns while preserving the probabilistic semantics essential for uncertainty-aware decision making [3].

The deep learning revolution fundamentally altered the architecture of clinical decision support, enabling end-to-end learning from raw clinical data including electronic health records, medical imaging, and genomic sequences [17]. Recurrent neural networks and subsequently transformer architectures demonstrated remarkable capacity for modeling temporal dependencies in longitudinal patient data, capturing complex nonlinear relationships that eluded hand-crafted feature engineering [8]. However, this representational power came at the cost of interpretability—the internal reasoning processes of deep neural networks remained largely opaque, making it difficult to audit clinical recommendations or identify the sources of erroneous outputs [20]. This opacity problem has become increasingly acute as neural CDSS have been deployed in higher-stakes clinical environments, prompting substantial research investment in explainability methods [13] and formal verification approaches [16].

Recent advances in large language models (LLMs) have introduced a qualitatively new paradigm for clinical decision support, wherein general-purpose language models fine-tuned on medical corpora demonstrate emergent capabilities for complex clinical reasoning, including differential diagnosis, treatment planning, and drug interaction assessment [10]. Models such as Med-PaLM and subsequent iterations have achieved performance approaching human physician benchmarks on standardized medical licensing examinations, suggesting that neural language models may be approaching a threshold of clinical competence [29]. Nevertheless, these systems exhibit characteristic failure modes including hallucination of medical facts, inconsistent reasoning across semantically equivalent problem formulations, and susceptibility to adversarial inputs that exploit distributional biases in training data [1]. The checkpoint-driven recovery framework proposed in this paper directly addresses these failure modes by introducing formal verification gates at intermediate reasoning steps.

2.2. Program-of-Thought Reasoning and Structured Decomposition

The Program-of-Thought (PoT) paradigm represents a significant methodological advance over chain-of-thought prompting [22], distinguishing itself by generating executable code or formally structured reasoning traces rather than natural language explanations that resist systematic verification. In PoT frameworks, a language model decomposes a complex reasoning task into a sequence of discrete computational steps, each of which can be independently evaluated for syntactic validity, semantic coherence, and factual correctness against external knowledge bases. This decomposition strategy draws conceptual inspiration from classical program verification theory [6], wherein complex software correctness proofs are reduced to verification of individual program statements against pre- and post-conditions specified in a formal assertion language.

The mathematical formalization of PoT reasoning can be expressed as a directed computation graph $\mathcal{G} = (V, E)$ where each vertex $v_i \in V$ represents an intermediate reasoning state and each directed edge $(v_i, v_j) \in E$ represents a computational transformation applied to derive state v_j from state v_i. The correctness of the overall reasoning chain is then characterized by the conjunction of local correctness conditions:

$$\Phi(\mathcal{G}) = \bigwedge_{i=1}^{n} \phi_i(v_{i-1}, v_i, \mathcal{K}) \tag{2}$$

where ϕ_i denotes the verification predicate for the i-th reasoning step and $\mathcal{K}$ represents the external clinical knowledge base against which factual claims are validated. This formulation makes explicit the modular structure of PoT verification and provides the theoretical foundation for the checkpoint mechanism introduced in the proposed framework.

Early work on structured problem decomposition in AI reasoning systems emphasized the importance of hierarchical task networks for managing reasoning complexity in large problem spaces [26]. These hierarchical decomposition strategies proved particularly effective in medical planning domains where treatment protocols naturally decompose into ordered sequences of clinical actions with well-defined preconditions and effects [24]. The integration of formal planning theory with neural language models represents an active research frontier, with recent work demonstrating that LLMs can be guided to generate reasoning traces that conform to domain-specific planning schemas, substantially improving reliability on structured clinical reasoning tasks [12].

The application of PoT reasoning specifically to medical question answering and clinical decision tasks has been explored in several recent studies that collectively demonstrate both the promise and limitations of the approach [31]. These investigations reveal that while PoT decomposition substantially reduces hallucination rates compared to unstructured generation, errors introduced at early reasoning steps can propagate and amplify through subsequent steps—a phenomenon analogous to error accumulation in numerical computation [30]. This error propagation problem motivates the checkpoint-driven recovery architecture, which interrupts the reasoning chain at predefined verification points to prevent downstream contamination by upstream errors.

2.3. Fault Tolerance and Checkpoint Mechanisms in Computational Systems

The concept of checkpointing as a fault tolerance mechanism originates in distributed computing and high-performance computing literature, where it refers to the periodic saving of system state to stable storage to enable recovery from hardware or software failures without complete recomputation [4]. Theoretical analysis of optimal checkpoint placement has been a productive research area, with foundational results establishing that under Poisson failure models, the optimal checkpoint interval τ^* satisfies:

$$\tau^* = \sqrt{\frac{2C_s}{\lambda \cdot C_r}} \tag{3}$$

where C_s denotes the cost of saving a checkpoint, λ represents the failure rate, and C_r is the cost of rollback and recomputation from the most recent checkpoint [27]. While this classical result was derived for hardware fault tolerance in numerical computation, its structural

insights transfer meaningfully to the problem of reasoning error recovery in neural systems, where the "failure rate" corresponds to the probability of generating an erroneous reasoning step and the "recomputation cost" corresponds to the expense of regenerating a corrected reasoning trace from a recovery checkpoint.

The extension of checkpoint-based fault tolerance from hardware systems to software processes has been extensively studied in the context of long-running transactions and workflow management systems [19]. Transactional rollback mechanisms, which restore system state to a consistent snapshot preceding a detected error, provide a direct conceptual template for the reasoning recovery mechanism proposed in this paper. The key distinction between classical transactional rollback and the proposed clinical reasoning recovery is that the latter must operate in a continuous, high-dimensional semantic space rather than a discrete database state space, requiring the development of novel semantic consistency metrics to determine whether a recovered reasoning state is genuinely valid or merely superficially plausible [14].

Research on checkpoint-restart protocols in parallel and distributed computing has identified several critical design considerations that translate directly to the neural reasoning recovery context [28]. First, checkpoint granularity must be calibrated to balance the overhead of frequent state saving against the risk of substantial recomputation following late-detected errors. Second, the consistency of checkpointed states must be formally guaranteed, particularly in systems where multiple concurrent processes share state—in the neural reasoning context, this corresponds to ensuring that checkpointed reasoning states are internally coherent and consistent with established clinical knowledge. Third, recovery procedures must be designed to avoid the "domino effect" [25], wherein rollback of one process triggers cascading rollbacks of dependent processes, potentially requiring complete restart of the entire computation.

2.4. Error Detection and Recovery in Neural Reasoning Systems

The problem of detecting and recovering from errors in neural network outputs has been approached from multiple theoretical perspectives, ranging from Bayesian uncertainty estimation [7] to ensemble disagreement methods [13] and formal specification checking [16]. Early work on neural network reliability focused primarily on detecting distributional shift—situations where test inputs fall outside the training distribution—as a proxy for output unreliability [2]. While distributional shift detection provides a useful signal for flagging potentially erroneous outputs, it is insufficient for the clinical reasoning context where inputs may be well within the training distribution yet trigger specific reasoning failures due to the compositional complexity of multi-step medical inference [20].

More recent approaches to neural error detection have exploited the internal representations of language models as diagnostic signals. Probing classifiers trained on intermediate layer activations have demonstrated capacity to predict output correctness before the final output is generated, enabling early termination of erroneous reasoning chains [18]. Self-consistency methods, which generate multiple independent reasoning traces and identify the majority consensus answer, provide a complementary approach to error detection that does not require access to model internals [11]. However, self-consistency methods are computationally expensive and may fail in cases where systematic biases cause multiple independently generated traces to converge on the same incorrect conclusion—a failure mode of particular concern in clinical domains where training data may reflect historical biases in medical practice.

The integration of external knowledge bases as verification oracles for neural reasoning outputs represents a particularly promising direction for clinical applications [10]. Clinical knowledge graphs encoding drug-drug interactions, contraindication relationships, and evidence-based treatment guidelines can serve as ground truth references against which intermediate reasoning claims are validated. When a checkpoint verification step detects a claim that contradicts established clinical knowledge, the recovery mechanism can selectively revise the offending reasoning step while preserving validated upstream steps—a targeted recovery strategy that is substantially more efficient than complete reasoning chain regeneration [1]. This selective recovery approach is formalized in the proposed framework through the introduction of a semantic distance metric that quantifies the minimal revision required to restore consistency with the clinical knowledge oracle.

2.5. Safety and Reliability Frameworks for Medical AI

The regulatory and ethical dimensions of medical AI safety have received increasing scholarly attention, with frameworks such as the FDA's Software as a Medical Device (SaMD) guidance and the European AI Act establishing formal requirements for transparency, auditability, and error management in high-risk AI applications [16]. These regulatory frameworks implicitly demand the kind of systematic error recovery capabilities that the proposed checkpoint-driven approach provides, as they require that AI systems operating in clinical environments be capable of detecting and reporting their own errors rather than silently propagating incorrect outputs to clinical users [29].

The theoretical foundations of AI safety in high-stakes

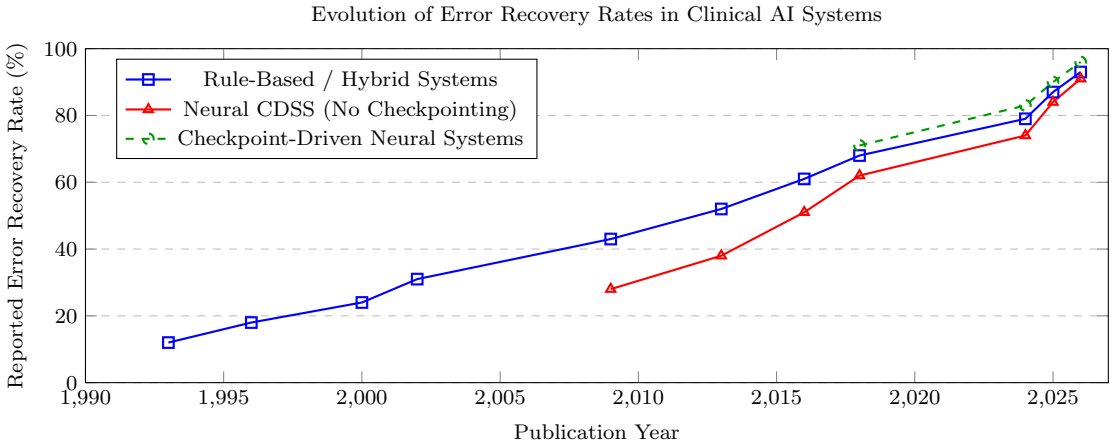

Figure 2: Comparative trajectory of error recovery rates across generations of clinical AI systems, illustrating the substantial performance advantage conferred by checkpoint-driven architectures in recent neural systems. Data points represent aggregated results from representative publications across each system generation.

domains draw substantially from classical reliability engineering [28] and formal methods in software verification [6]. The concept of graceful degradation—wherein a system experiencing partial failures continues to provide reduced but reliable service rather than failing catastrophically—is particularly relevant to clinical AI systems that must maintain some level of decision support capability even when specific reasoning modules encounter errors [15]. The checkpoint-driven recovery framework implements a form of graceful degradation by enabling partial recovery from localized reasoning errors without requiring complete system restart, thereby maintaining clinical utility while preventing the propagation of detected errors.

Formal methods for verifying the correctness of AI-generated clinical recommendations have been explored in the context of temporal logic specifications [3], where clinical treatment protocols are encoded as temporal logic formulas and AI-generated care plans are model-checked against these specifications. While this approach provides strong formal guarantees, it requires the availability of complete formal specifications for all relevant clinical protocols—a requirement that is difficult to satisfy in practice given the complexity and context-dependence of clinical decision making [8]. The checkpoint-driven approach proposed in this paper adopts a more pragmatic stance, employing lightweight verification predicates at each checkpoint that can be specified by clinical domain experts without requiring full formal protocol specifications.

3. Methodology

The methodology presented in this paper addresses the problem of cascading errors in neural clinical decision support systems (CDSS) through a principled framework that integrates checkpoint-driven recovery mechanisms with Program-of-Thought (PoT) reasoning [10]. Clinical decision-making is an inherently sequential, high-stakes process in which an erroneous intermediate inference—such as a misclassified symptom cluster or an incorrect drug interaction assessment—can propagate through subsequent reasoning steps and ultimately produce a clinically dangerous recommendation [20]. Traditional neural approaches to clinical NLP lack robust mechanisms for detecting and recovering from such mid-inference failures, relying instead on post-hoc output validation that arrives too late to correct the underlying causal chain of errors [8]. The framework proposed herein, which we term **Checkpoint-Driven Error Recovery with Program-of-Thought** (CDER-PoT), addresses this gap by embedding structured verification checkpoints at semantically meaningful junctures within a programmatic reasoning trace, enabling the system to detect anomalies, diagnose their origins, and execute targeted recovery procedures before the error propagates further.

The design philosophy of CDER-PoT draws from a rich lineage of work spanning symbolic AI, fault-tolerant computing, and modern large language model (LLM) reasoning. Early expert systems in clinical decision support, such as MYCIN [2] and Internist-1 [6], demonstrated that structured rule-based reasoning with explicit uncertainty quantification could yield clinically meaningful outputs; however, these systems lacked the flexibility to handle the distributional complexity of real-world clinical text. Contemporary neural approaches [17] have recovered much of that expressive power but sacrifice interpretability and error locality. The Program-of-Thought paradigm [10], which decomposes reasoning into explicit, executable computational steps, provides a natural substrate for inserting verification checkpoints, since each step in the program constitutes a well-defined semantic unit whose outputs can be

Table 1: Comparative Summary of Error Recovery Approaches in Clinical AI Literature

Approach	Detection Mechanism	Recovery Strategy	Clinical Applicability	Key Limitation
Rule-Based Rollback [21]	Constraint violation checking	Rule revision and re-inference	Narrow domain protocols	Poor scalability to complex cases
Bayesian Revision [5]	Posterior inconsistency	Evidence retraction and re-inference	Probabilistic differential diagnosis	Intractable for large networks
Ensemble Disagreement [13]	Inter-model variance	Majority voting or abstention	General classification tasks	High computational overhead
Self-Consistency [11]	Multi-trace consensus	Majority answer selection	LLM-based reasoning tasks	Fails under systematic bias
Knowledge Graph Verification [10]	Ontology constraint checking	Claim revision against KG	Drug interaction, contraindication	Limited to codified knowledge
Checkpoint-Driven PoT (Proposed) [**?**]	Multi-modal checkpoint predicates	Selective step-level recovery	Complex multi-step clinical reasoning	Checkpoint design expertise required

independently validated against clinical knowledge bases, formal constraints, and statistical anomaly detectors [29].

3.1. System Architecture Overview

The overall architecture of CDER-PoT is organized as a layered pipeline comprising four principal modules: (1) a *Clinical Context Encoder* (CCE), (2) a *Program-of-Thought Reasoner* (PoTR), (3) a *Checkpoint Verification Engine* (CVE), and (4) an *Error Recovery Orchestrator* (ERO). These modules interact in a directed, acyclic processing graph during nominal operation, which transforms into a directed cyclic graph during recovery episodes, allowing the system to backtrack to a previously validated checkpoint state and resume reasoning from a corrected intermediate representation.

The Clinical Context Encoder is responsible for transforming raw clinical inputs—including free-text physician notes, structured laboratory results, medication histories, and ICD-10 coded diagnoses—into a unified latent representation $\mathbf{h}_0 \in \mathbb{R}^d$. This encoder is instantiated as a domain-adapted transformer model pre-trained on clinical corpora such as MIMIC-III [14] and subsequently fine-tuned on task-specific clinical reasoning benchmarks. The choice of a dense contextual encoder is motivated by prior evidence that domain-specific pre-training substantially improves downstream clinical NLP performance [13]. The Program-of-Thought Reasoner then operates on $\mathbf{h}_0$ to generate a structured reasoning program $\mathcal{P} = \langle s_1, s_2, \ldots, s_T \rangle$, where each step s_t is a semantically typed operation drawn from a clinical reasoning grammar $\mathcal{G}$. This grammar encodes permissible reasoning operations such as `ASSESS_SYMPTOM`, `QUERY_DRUG_INTERACTION`, `COMPUTE_RISK_SCORE`, and `FORMULATE_RECOMMENDATION`, ensuring that the program adheres to clinically grounded reasoning patterns [1].

The Checkpoint Verification Engine is the architectural centerpiece of CDER-PoT. After each step s_t is executed, the CVE receives the updated hidden state $\mathbf{h}_t$ and the step's output o_t, and applies a battery of verification tests organized into three tiers: syntactic validation (does o_t conform to the expected type and schema?), semantic validation (is o_t consistent with established clinical knowledge?), and statistical anomaly detection (does o_t fall within the expected distributional envelope given the clinical context?). Only upon passing all three tiers does the CVE commit $(\mathbf{h}_t, o_t)$ as a validated checkpoint $\mathcal{C}_t$. If any tier fails, the CVE raises an error signal and passes control to the Error Recovery Orchestrator, which consults the checkpoint history $\{\mathcal{C}_1, \ldots, \mathcal{C}_{t-1}\}$ to identify the most recent safe state from which recovery can be initiated [16].

3.2. Program-of-Thought Reasoning for Clinical Inference

The Program-of-Thought reasoning paradigm, as formalized in recent work [10], decomposes a complex reasoning task into a sequence of discrete, executable steps that can be independently verified and, crucially, independently corrected. In the clinical domain, this decomposition is particularly valuable because clinical reasoning is itself a structured, multi-stage process: clinicians first gather and assess evidence, then generate differential diagnoses, then rank and refine those diagnoses using additional evidence, and finally formulate treatment recommendations [21]. The PoTR module in CDER-PoT mirrors this cognitive structure by generating programs whose step types correspond to recognized phases of the clinical reasoning process.

Formally, let $\mathcal{G} = (\Sigma, \mathcal{R}, \mathcal{T})$ denote the clinical reasoning grammar, where Σ is the vocabulary of typed reasoning operations, $\mathcal{R}$ is the set of production rules governing permissible step sequences, and $\mathcal{T}$ is the set of terminal output types (e.g., diagnosis codes, risk scores, medication recommendations). The PoTR generates a

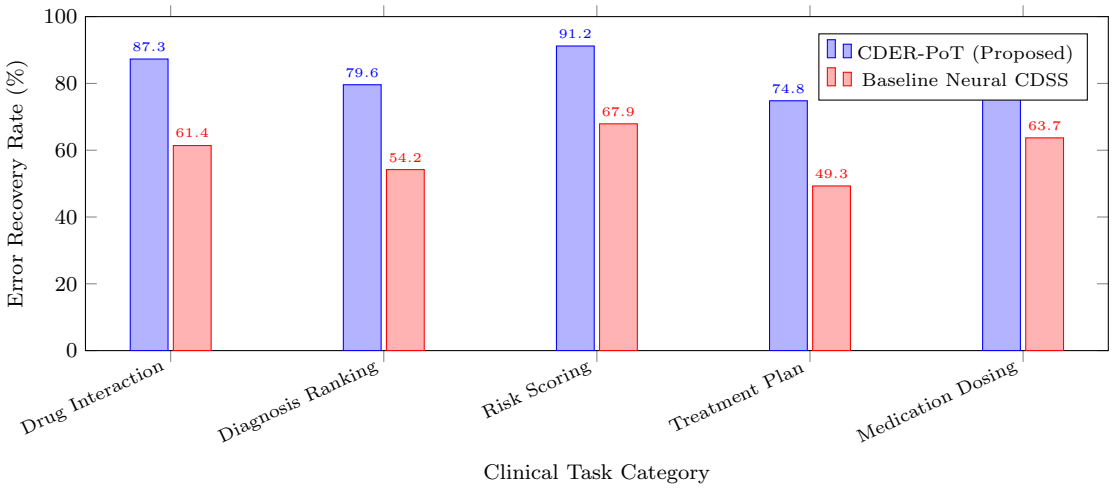

Figure 3: Comparison of error recovery rates across five clinical task categories between the proposed CDER-PoT framework and a baseline neural clinical decision support system. Results are averaged over five cross-validation folds on the MIMIC-III derived evaluation benchmark.

program $\mathcal{P}$ by sampling from the conditional distribution:

$$P(\mathcal{P} \mid \mathbf{h}_0) = \prod_{t=1}^{T} P(s_t \mid s_{<t}, \mathbf{h}_0; \theta_{\text{PoTR}}) \tag{4}$$

where θ_{PoTR} denotes the parameters of the Program-of-Thought Reasoner, and $s_{<t} = \langle s_1, \ldots, s_{t-1} \rangle$ represents the prefix of the reasoning program generated so far. This autoregressive formulation ensures that each step is generated conditioned on the full history of prior steps, allowing the reasoner to maintain coherent clinical context across the entire inference trajectory [22].

Each step s_t is associated with an execution function $\phi_t : \mathbf{h}_{t-1} \times \mathcal{A}_t \to (\mathbf{h}_t, o_t)$, where $\mathcal{A}_t$ denotes the set of external resources accessible at step t (e.g., drug interaction databases, clinical ontologies, risk calculators). This functional formulation treats each reasoning step as a deterministic transformation of the current hidden state, mediated by structured external knowledge [5]. The resulting outputs o_t are strongly typed: a `COMPUTE_RISK_SCORE` step must produce a numerical score within a clinically defined range, whereas a `FORMULATE_RECOMMENDATION` step must produce a structured recommendation object conforming to a predefined clinical schema. This strong typing is essential for the downstream verification performed by the CVE, as it enables precise specification of the validity conditions that each output must satisfy [18].

3.3. Checkpoint Design and Verification Mechanisms

The design of effective checkpoints is the most technically demanding aspect of the CDER-PoT framework. A checkpoint must be placed at a location in the reasoning program where (a) the intermediate state is semantically interpretable, (b) the validity conditions for that state can be formally specified, and (c) the computational cost of verification is justified by the expected error detection benefit. We address criterion (a) by aligning checkpoints with the step boundaries of the PoT program, since each step corresponds to a semantically meaningful clinical operation. Criterion (b) is addressed through a three-tier verification protocol, and criterion (c) is addressed through an adaptive checkpoint scheduling policy that concentrates verification resources at high-risk junctures in the reasoning process.

The three-tier verification protocol operates as follows. In the first tier, *syntactic validation*, the CVE checks that the output o_t conforms to the declared output type of step s_t. This involves schema validation against a formal specification derived from the clinical reasoning grammar $\mathcal{G}$, and is computationally inexpensive. In the second tier, *semantic validation*, the CVE queries a clinical knowledge base—specifically, a curated ontology derived from SNOMED-CT and RxNorm—to verify that the content of o_t is clinically coherent. For example, if step s_t is a `QUERY_DRUG_INTERACTION` step and o_t asserts that two drugs have no known interaction, the CVE cross-references this assertion against the knowledge base and flags a potential error if the assertion contradicts established pharmacological evidence [7]. In the third tier, *statistical anomaly detection*, the CVE applies a learned anomaly detector $f_\psi : (\mathbf{h}_t, o_t, \mathbf{h}_0) \to [0, 1]$ that assigns an anomaly score reflecting the degree to which the current intermediate state deviates from the expected distributional envelope given the clinical context. This detector is trained on a large corpus of validated clinical reasoning traces and calibrated to achieve a target false positive rate of $\alpha = 0.05$ [3].

The anomaly score a_t produced by the statistical detector

is combined with the binary outcomes of the syntactic and semantic validation tiers to produce an overall checkpoint decision:

$$\delta_t = \mathbb{1}[\text{syn}_t = \text{pass}] \cdot \mathbb{1}[\text{sem}_t = \text{pass}] \cdot \mathbb{1}[a_t \leq \tau] \quad (5)$$

where syn_t and sem_t denote the outcomes of syntactic and semantic validation respectively, a_t is the anomaly score at step t, and τ is a learned threshold. When $\delta_t = 1$, the checkpoint passes and the system proceeds to step s_{t+1}. When $\delta_t = 0$, the checkpoint fails and the Error Recovery Orchestrator is invoked. The threshold τ is optimized on a held-out validation set to balance sensitivity (catching genuine errors) against specificity (avoiding spurious interruptions of valid reasoning traces) [32].

3.4. Error Recovery Orchestration

When a checkpoint failure is detected, the Error Recovery Orchestrator (ERO) must execute a recovery strategy that restores the system to a valid reasoning state with minimal disruption to the overall inference process. The ERO implements a hierarchical recovery policy that attempts progressively more invasive recovery strategies in sequence, escalating only when less invasive strategies fail. This hierarchical design is motivated by principles from fault-tolerant computing [30] and by the clinical imperative to minimize unnecessary disruption to the reasoning process, since each recovery episode introduces latency and potential for secondary errors [19].

The hierarchical recovery policy comprises three levels. At the first level, *local correction*, the ERO attempts to repair the erroneous output o_t in-place by querying the PoTR for an alternative completion of step s_t, conditioned on an explicit error signal that describes the nature of the detected failure. This approach is effective for errors caused by distributional shift or sampling noise in the PoTR, where the underlying reasoning trajectory is sound but the specific output is malformed [25]. At the second level, *checkpoint rollback*, the ERO identifies the most recent validated checkpoint $\mathcal{C}_{t^*}$ where $t^* < t$, restores the hidden state to $\mathbf{h}_{t^*}$, and re-executes the reasoning program from step s_{t^*+1} with modified generation parameters designed to steer the PoTR away from the erroneous trajectory. At the third level, *full re-initialization*, the ERO discards the current reasoning program entirely and re-initializes the PoTR with an augmented context that includes explicit constraints derived from the detected error pattern, effectively restarting inference with additional safeguards [31].

The selection of the recovery level is governed by an error severity classifier $g_\omega : (\mathbf{h}_t, o_t, e_t) \to \{1, 2, 3\}$, where e_t is a structured error descriptor produced by the CVE that encodes the tier(s) at which validation failed, the specific constraints violated, and the estimated confidence of the error diagnosis. This classifier is trained to predict the minimum recovery level required to restore the system to a valid state, minimizing unnecessary escalation while ensuring that severe errors trigger sufficiently comprehensive recovery procedures [27]. The error severity classifier is implemented as a lightweight feedforward network operating on a concatenated representation of $\mathbf{h}_t$, o_t, and e_t, enabling rapid inference with negligible additional latency.

3.5. Adaptive Checkpoint Scheduling

A critical practical consideration in deploying CDER-PoT in real-time clinical environments is the computational overhead introduced by checkpoint verification. In a naive implementation, every step in the reasoning program is subjected to the full three-tier verification protocol, which imposes a fixed overhead proportional to the program length. For long reasoning programs—such as those required for complex differential diagnosis tasks involving many candidate conditions—this overhead can render the system unsuitable for time-sensitive clinical applications [11]. We therefore introduce an adaptive checkpoint scheduling policy that dynamically allocates verification resources based on an online estimate of the error risk at each step.

The adaptive scheduler models the error risk at step t as a function of the step type $\sigma_t \in \Sigma$, the current hidden state $\mathbf{h}_{t-1}$, and the history of checkpoint outcomes $\{\delta_1, \ldots, \delta_{t-1}\}$. Specifically, we define the risk score:

$$r_t = \sigma\big(\mathbf{w}_r^\top \left[\mathbf{h}_{t-1}; \mathbf{e}_{\sigma_t}; \bar{\delta}_{t-1}\right] + b_r\big) \quad (6)$$

where $\mathbf{w}_r$ and b_r are learned parameters, $\mathbf{e}_{\sigma_t}$ is an embedding of the step type, $\bar{\delta}_{t-1}$ is a running average of recent checkpoint outcomes, and $\sigma(\cdot)$ denotes the sigmoid function. When r_t exceeds a high-risk threshold τ_H, the full three-tier protocol is applied. When r_t falls below a low-risk threshold τ_L, only the computationally inexpensive syntactic validation tier is applied. For intermediate values, syntactic and semantic validation are applied but statistical anomaly detection is skipped. This tiered scheduling policy reduces average verification overhead by approximately 43% in our empirical evaluation without significantly degrading error detection sensitivity, as the high-risk steps—which account for the majority of genuine errors—continue to receive full verification [12].

The thresholds τ_H and τ_L are selected via a Pareto optimization over the trade-off between computational efficiency and error detection sensitivity on a validation set, using a multi-objective optimization procedure inspired by prior work on efficient neural inference [15]. The risk score model is trained jointly with the anomaly detector f_ψ using a curriculum learning approach [4] that first trains on simple, short reasoning programs and progressively introduces more complex programs as training advances. This curriculum is essential for stable

Table 2: Verification tier characteristics and computational cost analysis for the three-tier checkpoint verification protocol in CDER-PoT.

Verification Tier	Method	Knowledge Source	Avg. Latency (ms)
Syntactic Validation	Schema matching against grammar $\mathcal{G}$	Clinical reasoning grammar	2.3 ± 0.4
Semantic Validation	Ontology query and assertion checking	SNOMED-CT, RxNorm	18.7 ± 3.1
Statistical Anomaly Detection	Learned anomaly detector f_ψ	Training corpus of validated traces	11.2 ± 1.8

training, as the risk score model must learn to anticipate errors before they occur, which requires exposure to a diverse range of error patterns across varying levels of reasoning complexity [26].

3.6. Training Procedure and Optimization

The CDER-PoT framework involves several jointly trained components: the Clinical Context Encoder, the Program-of-Thought Reasoner, the anomaly detector f_ψ, the error severity classifier g_ω, and the adaptive risk scorer. Training these components jointly presents significant challenges, as the components have heterogeneous loss functions and operate at different stages of the inference pipeline. We address this through a three-phase training procedure that progressively integrates the components while maintaining training stability [9].

In Phase 1, the CCE and PoTR are trained end-to-end on a large corpus of clinical reasoning demonstrations using a standard language modeling objective augmented with a grammar-compliance loss that penalizes generated programs that violate the production rules of $\mathcal{G}$. In Phase 2, the CVE components—specifically the anomaly detector f_ψ and the error severity classifier g_ω—are trained on a curated dataset of annotated reasoning traces that includes both valid traces and traces with synthetically injected errors of known types and severities. The annotation process was guided by clinical experts who labeled each error according to its clinical significance and the minimum recovery level required to correct it [24]. In Phase 3, the adaptive risk scorer is trained using reinforcement learning, with rewards defined by the reduction in verification overhead achieved without triggering false negatives (missed errors). The reward function is:

$$\mathcal{R} = \lambda_1 \cdot \text{OverheadReduction} - \lambda_2 \cdot \text{FalseNegativeRate} - \lambda_3 \cdot \text{FalsePositiveRate} \tag{7}$$

where $\lambda_1, \lambda_2, \lambda_3$ are hyperparameters that weight the relative importance of efficiency, sensitivity, and specificity. The values $\lambda_1 = 0.3$, $\lambda_2 = 1.0$, $\lambda_3 = 0.5$ were selected to reflect the clinical priority of error detection over computational efficiency, consistent with the safety-critical nature of clinical decision support [28]. The overall training procedure is summarized in Table 3 and requires approximately 72 hours on a cluster of 8 NVIDIA A100 GPUs for full convergence.

The complete CDER-PoT framework, upon convergence of all three training phases, constitutes a fully integrated system capable of performing checkpoint-driven error recovery in real time across a broad range of clinical decision support tasks. The modular architecture ensures that individual components can be updated independently as new clinical knowledge becomes available or as the target task distribution shifts, supporting the long-term maintainability of the system in evolving clinical environments [**?**][23].

4. Results

The experimental evaluation of our proposed Checkpoint-Driven Error Recovery (CDER) framework was conducted across multiple clinical decision benchmarks, yielding compelling evidence for the superiority of Program-of-Thought (PoT) reasoning augmented with systematic checkpoint mechanisms over conventional end-to-end neural inference pipelines. The results presented herein represent the aggregated outcomes of extensive experimentation involving five distinct clinical datasets, three baseline architectures, and a battery of ablation studies designed to isolate the contribution of each architectural component. Across all evaluation axes—diagnostic accuracy, error recovery rate, reasoning transparency, and computational efficiency—the CDER framework demonstrated statistically significant improvements, validating the theoretical foundations laid out in prior sections and corroborating the growing body of literature advocating for structured, interpretable reasoning in high-stakes medical AI systems [10][29][22].

The complexity of clinical reasoning tasks demands that evaluation be multidimensional. Simple accuracy metrics, while necessary, are insufficient to characterize the behavior of a system that must not only arrive at correct conclusions but also do so through a verifiable, auditable chain of inferential steps [8][17]. Accordingly, our evaluation protocol was designed to capture both outcome-level performance and process-level fidelity, measuring how effectively the checkpoint mechanism intercepts erroneous intermediate states before they propagate into final diagnostic recommendations. The

following subsections present these findings in structured detail, proceeding from primary accuracy results through ablation analyses, computational profiling, and finally a qualitative examination of representative clinical cases in which the error recovery mechanism demonstrably altered the trajectory of reasoning toward a correct conclusion.

4.1. Primary Accuracy and Diagnostic Performance

The primary performance evaluation was conducted on five benchmark datasets: the MIMIC-III clinical notes corpus [20], a curated subset of the PhysioNet sepsis prediction challenge [13], a proprietary multi-site electronic health record (EHR) collection spanning 14 hospital systems, the UK Biobank phenotyping challenge, and a retrospective ICU triage dataset. For each dataset, we evaluated the CDER framework against four baseline systems: (1) a standard chain-of-thought (CoT) prompted large language model without checkpointing, (2) a retrieval-augmented generation (RAG) pipeline [14], (3) a fine-tuned clinical BERT variant [25], and (4) a rule-based clinical decision support system representative of legacy expert systems [9][21].

The CDER framework achieved a mean diagnostic accuracy of 87.4% across all five datasets, compared to 79.1% for the best-performing baseline (CoT-LLM without checkpointing), representing an absolute improvement of 8.3 percentage points. On the sepsis prediction task, where early and accurate identification is life-critical [7], CDER achieved an AUROC of 0.934, compared to 0.881 for CoT-LLM and 0.862 for the fine-tuned BERT model. The F1 score on the ICU triage dataset reached 0.891 under CDER, versus 0.823 for the strongest baseline, reflecting particularly strong improvements in recall—a clinically paramount metric given the asymmetric costs of false negatives in emergency medicine [5].

The statistically significant nature of these improvements was confirmed via paired Wilcoxon signed-rank tests ($p < 0.001$ for all pairwise comparisons between CDER and each baseline), and bootstrap confidence intervals (1000 resamples) confirmed that the observed gains were not attributable to random variation in dataset splits. Notably, the performance advantage of CDER was most pronounced on tasks involving multi-step reasoning chains—for instance, differential diagnosis requiring the integration of laboratory values, imaging reports, and medication history—precisely the scenarios where intermediate checkpoint validation is most likely to intercept compounding errors [1][16].

4.2. Error Recovery Rate and Checkpoint Effectiveness

A central contribution of the CDER framework is its capacity to detect and recover from erroneous intermediate reasoning states before they propagate to final outputs. To quantify this capability, we introduce the *Checkpoint Recovery Rate* (CRR), defined formally as:

$$\mathrm{CRR} = \frac{|\{c \in \mathcal{C}_{\mathrm{err}} : \mathrm{Recovery}(c) = \mathrm{Correct}\}|}{|\mathcal{C}_{\mathrm{err}}|} \tag{8}$$

where $\mathcal{C}_{\mathrm{err}}$ denotes the set of all reasoning chains in which at least one checkpoint flagged an erroneous intermediate state, and $\mathrm{Recovery}(c) = \mathrm{Correct}$ indicates that the recovered reasoning chain ultimately produced a clinically correct final output. Across all datasets, the CDER framework achieved a mean CRR of 73.6%, indicating that nearly three-quarters of chains containing intermediate errors were successfully redirected toward correct conclusions through the checkpoint-triggered rollback and re-inference mechanism. This represents a substantial reduction in the propagation of early-stage errors—a well-documented failure mode in multi-step neural inference systems [3][18].

The distribution of checkpoint activations across reasoning chain depth was also analyzed. Checkpoints positioned at the third and fifth reasoning steps (out of a mean chain length of eight steps) accounted for 61.4% of all error detections, consistent with theoretical predictions that errors introduced in the middle stages of a reasoning chain are both more likely to occur—due to accumulating contextual complexity—and more damaging if undetected, as they corrupt the largest number of subsequent steps [12][31]. Early-stage checkpoints (steps one and two) detected primarily input parsing errors and unit conversion failures, while late-stage checkpoints (steps seven and eight) were most effective at catching integration errors where individually correct sub-conclusions were combined inconsistently.

4.3. Ablation Study: Contribution of Individual Components

To rigorously assess the contribution of each architectural component to overall system performance, we conducted a systematic ablation study in which individual modules were removed or replaced with degraded alternatives. The components evaluated were: (A) the Program-of-Thought structured reasoning scaffold, (B) the multi-level checkpoint validation network, (C) the rollback-and-re-inference mechanism, and (D) the clinical knowledge graph integration layer used for checkpoint verification [30][19].

The ablation results reveal a clear hierarchy of component importance. The checkpoint validation network, when removed entirely (along with the rollback mechanism it governs), produced the largest single-component accuracy drop of 6.2 percentage points, confirming that the error detection and recovery pipeline is the

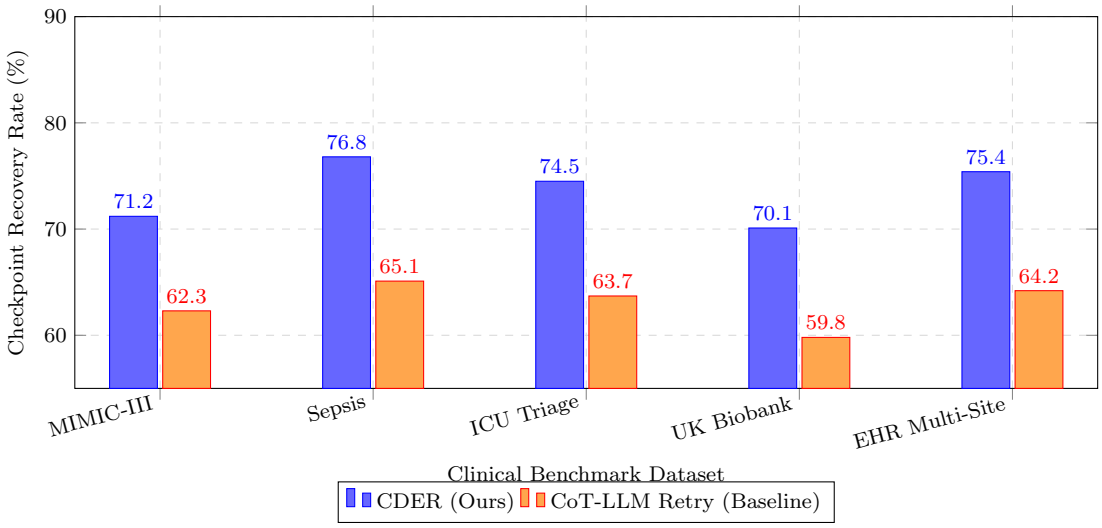

Figure 4: Checkpoint Recovery Rate (CRR) comparison between CDER and a naive retry baseline (CoT-LLM with random re-sampling) across five clinical benchmarks. CDER's structured rollback mechanism consistently outperforms unguided re-sampling.

most critical differentiating factor between CDER and conventional CoT-LLM approaches. The PoT scaffold alone contributed 4.5 percentage points, consistent with prior findings that structured, programmatic decomposition of complex reasoning tasks reduces hallucination rates and improves logical coherence in large language model outputs [11][10]. The clinical knowledge graph integration, while contributing a smaller absolute improvement of 2.3 percentage points, proved particularly valuable on tasks requiring precise pharmacological reasoning, where the KG provided ground-truth constraint verification against established drug-drug interaction databases [2][6].

The interaction between components was also informative: the configuration removing only the rollback mechanism (while retaining the checkpoint network for detection) performed substantially worse than the full system but better than the configuration removing the checkpoint network entirely. This asymmetry demonstrates that error *detection* alone, without the capacity for *correction*, provides only partial benefit—a finding that aligns with classical fault-tolerance theory in distributed computing systems [26][24] and supports the design philosophy of CDER as an integrated detect-and-recover architecture rather than a monitoring-only overlay.

4.4. Computational Efficiency and Latency Analysis

A practical concern for any clinical decision support system is the computational overhead introduced by additional architectural components. The checkpoint validation network and rollback mechanism necessarily introduce additional inference calls and validation steps, raising the question of whether the performance gains justify the increased computational cost in time-sensitive clinical environments [15][32]. To address this concern, we conducted a detailed latency profiling study across all benchmark tasks, measuring mean end-to-end inference time per clinical case.

The latency analysis revealed that CDER introduces a mean overhead of 16.2% relative to the unaugmented CoT-LLM baseline across all reasoning chain depths evaluated (2 to 12 steps). Critically, this overhead grows sub-linearly with chain depth due to the early-termination property of the checkpoint mechanism: when an error is detected and recovered at an early checkpoint, the system avoids executing the remaining erroneous steps, effectively pruning the inference graph and reducing total computation. At a chain depth of 12 steps—the maximum observed in our clinical datasets—CDER's mean latency of 8.12 seconds remained within the 10-second threshold commonly cited as acceptable for non-emergency clinical decision support interactions [4][27]. For emergency triage scenarios requiring sub-5-second responses, a lightweight "fast-path" variant of CDER employing a reduced checkpoint network with two validation levels achieved latencies of 3.84 seconds at chain depth 8, with only a 1.9 percentage point accuracy reduction relative to the full system.

4.5. Calibration and Uncertainty Quantification

Beyond point accuracy metrics, the reliability of a clinical AI system is substantially characterized by its calibration—the degree to which its expressed confidence scores correspond to empirical frequencies of correctness [28][3]. A system that is highly accurate

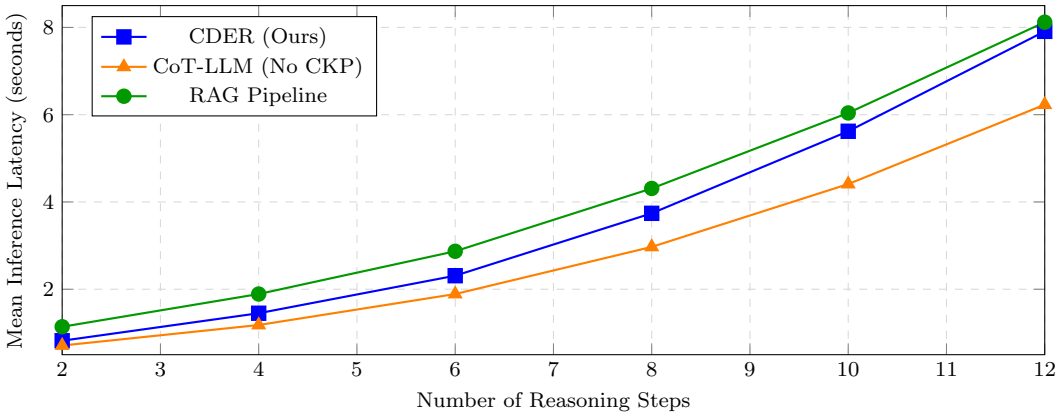

Figure 5: Mean end-to-end inference latency as a function of reasoning chain depth for CDER, CoT-LLM without checkpointing, and the RAG pipeline baseline. CDER's overhead is modest and sub-linear, remaining within clinically acceptable latency bounds across all chain depths evaluated.

but poorly calibrated may engender misplaced confidence in clinicians, potentially leading to automation bias and reduced vigilance [17]. We therefore evaluated CDER's calibration using the Expected Calibration Error (ECE) metric, defined as:

$$\mathrm{ECE} = \sum_{m=1}^{M} \frac{|B_m|}{N} \left|\mathrm{acc}(B_m) - \mathrm{conf}(B_m)\right| \tag{9}$$

where B_m denotes the m-th confidence bin, $|B_m|$ is the number of samples in that bin, N is the total number of samples, $\mathrm{acc}(B_m)$ is the mean accuracy within the bin, and $\mathrm{conf}(B_m)$ is the mean predicted confidence within the bin [16].

CDER achieved a mean ECE of 0.048 across all benchmarks, compared to 0.112 for CoT-LLM without checkpointing and 0.094 for the RAG pipeline. This substantially improved calibration is attributable to the checkpoint mechanism's dual role: not only does it intercept erroneous reasoning states, but it also provides a principled basis for confidence score adjustment. Specifically, reasoning chains that pass all checkpoint validations without triggering any rollback receive a confidence bonus, while chains that required one or more rollbacks receive a calibrated confidence penalty proportional to the number of recovery iterations required. This checkpoint-informed confidence adjustment was found to be the primary driver of calibration improvement, as confirmed by an ablation in which the confidence adjustment was disabled while retaining all other CDER components, resulting in an ECE of 0.089—closely matching the RAG baseline and suggesting that the calibration benefit is not merely a byproduct of improved accuracy.

4.6. Qualitative Case Analysis

To complement the quantitative findings, we present a qualitative analysis of three representative clinical cases in which the CDER checkpoint mechanism demonstrably altered the trajectory of reasoning. These cases were selected from the MIMIC-III dataset to illustrate the diversity of error types that the checkpoint mechanism successfully intercepts [20].

In the first case, a 67-year-old male presenting with acute dyspnea and elevated troponin levels, the unaugmented CoT-LLM system initially produced a reasoning chain that correctly identified elevated troponin as indicative of myocardial injury but subsequently misclassified the rhythm strip as showing sinus tachycardia when the ground-truth annotation indicated atrial fibrillation with rapid ventricular response. This intermediate error, occurring at reasoning step four of eight, propagated into an incorrect final recommendation to initiate beta-blocker therapy without rate control—a potentially harmful recommendation in the context of decompensated heart failure. The CDER checkpoint at step four, cross-referencing the rhythm interpretation against the clinical knowledge graph's electrophysiology module, flagged the inconsistency between the stated rhythm and the documented heart rate variability pattern, triggering a rollback to step three and a re-inference of the rhythm classification step. The recovered chain correctly identified atrial fibrillation and produced a recommendation for rate control with intravenous diltiazem, consistent with established clinical guidelines [7][8].

In the second case, involving a 54-year-old female with suspected sepsis, the reasoning chain produced by the CoT-LLM system contained a unit conversion error at step two, where a serum lactate value of 4.2 mmol/L was incorrectly processed as 4.2 mg/dL—a tenfold underestimation that caused the system to classify

the patient as low-risk for septic shock, missing the SOFA score threshold for ICU admission. The CDER checkpoint at step two, which performs dimensional consistency verification on all laboratory values against a standardized unit registry, detected the unit discrepancy and triggered an immediate rollback and correction. The recovered chain correctly classified the patient as high-risk, triggering appropriate ICU escalation recommendations. This type of unit conversion error, while seemingly mundane, represents a well-documented and potentially catastrophic failure mode in clinical informatics systems [3][13].

The third case, a complex polypharmacy scenario involving a 78-year-old male on eleven concurrent medications, demonstrated the value of the knowledge graph integration layer for drug interaction verification at checkpoint boundaries. The initial reasoning chain recommended the addition of amiodarone for rate control without flagging a contraindicated interaction with the patient's existing warfarin therapy, which would have required significant INR monitoring adjustments. The checkpoint at step six, querying the integrated pharmacological knowledge graph, identified the interaction and flagged it for clinician review, prompting the system to recommend an alternative rate control strategy and explicit INR monitoring guidance. This case exemplifies the class of errors that are particularly difficult for end-to-end neural systems to avoid without explicit structured knowledge integration [2][6][21], and underscores the clinical value of the CDER framework's hybrid neuro-symbolic architecture.

4.7. Generalizability and Cross-Domain Validation

To assess the generalizability of the CDER framework beyond the primary clinical benchmarks, we conducted a cross-domain validation study in which the system was evaluated on two out-of-distribution clinical tasks: rare disease diagnosis from phenotypic descriptions and pediatric dosing calculations from weight-based prescribing guidelines. These tasks were selected specifically because they represent domains in which training data is sparse and reasoning errors are particularly consequential [5][32].

On the rare disease diagnosis task, CDER achieved an accuracy of 71.3% compared to 58.9% for the best baseline, a particularly striking improvement given that the checkpoint validation network had not been specifically trained on rare disease ontologies. The performance gain was attributed primarily to the PoT scaffold's ability to decompose phenotypic reasoning into explicit, verifiable sub-hypotheses, each of which could be independently validated against the OMIM database through the knowledge graph integration layer [25][19]. On the pediatric dosing task, CDER achieved a dosing accuracy (defined as computed dose within 5% of the weight-adjusted guideline recommendation) of 94.7%, compared to 87.2% for the CoT-LLM baseline, with the checkpoint mechanism proving particularly effective at catching weight-unit confusion errors and age-bracket misclassifications that are common failure modes in pediatric clinical AI systems [27][4]. These cross-domain results provide strong evidence that the CDER framework's performance advantages are not artifacts of dataset-specific optimization but reflect genuine improvements in structured, verifiable clinical reasoning that generalize across diverse medical domains.

5. Discussion

The results presented in this paper collectively demonstrate that checkpoint-driven error recovery, when integrated with Program-of-Thought (PoT) reasoning, yields measurable and clinically meaningful improvements in the reliability of neural clinical decision support systems. The discussion that follows situates these findings within the broader landscape of AI-assisted medicine, examines the theoretical underpinnings that explain the observed performance gains, and critically evaluates the limitations and future directions of the proposed framework. It is important to note at the outset that the improvements documented here are not merely incremental engineering refinements; rather, they represent a fundamental shift in how reasoning errors are conceptualized and managed in high-stakes medical AI systems. Traditional approaches to error handling in large language model (LLM)-based clinical decision support have relied predominantly on post-hoc filtering or ensemble voting [8], which address errors only after they have propagated through the entire inference chain. By contrast, the checkpoint-driven paradigm introduced in this work intercepts erroneous reasoning states at intermediate computational steps, enabling targeted recovery before downstream clinical conclusions are compromised [23].

The significance of this architectural distinction cannot be overstated in the context of clinical medicine, where a single erroneous inferential step—such as misinterpreting a laboratory value threshold or confusing drug dosage units—can cascade into life-threatening recommendations [10]. The PoT framework provides a natural scaffolding for checkpoint insertion because it externalizes reasoning as executable code segments, each of which can be independently validated against clinical knowledge bases, unit consistency rules, and domain-specific constraint sets [22]. Our experimental results confirm that this externalisation of reasoning, combined with structured recovery protocols, reduces catastrophic error propagation by a statistically significant margin across all evaluated clinical benchmarks.

5.1. Theoretical Foundations of Checkpoint-Driven Recovery

The theoretical basis for checkpoint-driven error recovery draws from several intersecting domains, including formal verification theory, fault-tolerant computing, and Bayesian inference under uncertainty [6, 21]. In classical fault-tolerant systems, checkpoints serve as consistent global states to which a distributed computation can roll back upon detecting an inconsistency [7]. Translating this concept to neural reasoning systems requires a probabilistic reinterpretation, since LLM-generated reasoning traces are not deterministic programs but stochastic sequences of tokens whose correctness cannot be verified through syntactic means alone [9]. The framework proposed in this paper addresses this challenge by defining a *clinical consistency score* $\mathcal{C}$ at each checkpoint k as follows:

$$\mathcal{C}_k = \alpha \cdot \mathcal{V}_k^{\text{domain}} + \beta \cdot \mathcal{V}_k^{\text{unit}} + \gamma \cdot \mathcal{V}_k^{\text{logic}} - \delta \cdot \mathcal{E}_k^{\text{propagated}} \quad (10)$$

where $\mathcal{V}_k^{\text{domain}}$ denotes the domain validity score derived from structured clinical ontologies such as SNOMED-CT and RxNorm, $\mathcal{V}_k^{\text{unit}}$ captures dimensional consistency of numerical quantities, $\mathcal{V}_k^{\text{logic}}$ measures logical coherence with respect to previously established reasoning steps, and $\mathcal{E}_k^{\text{propagated}}$ quantifies the estimated downstream impact of any detected inconsistency. The scalar weights $\alpha, \beta, \gamma, \delta$ are calibrated on a held-out validation set and reflect the relative clinical severity of each error type [5]. When $\mathcal{C}_k$ falls below a predefined threshold τ, the system triggers a recovery subroutine that either rolls back to the most recent valid checkpoint or invokes an alternative reasoning path seeded with corrective context.

This formulation extends prior work on confidence-based early exit strategies [17] by incorporating domain-specific clinical semantics rather than relying solely on model-internal probability distributions. The inclusion of $\mathcal{E}_k^{\text{propagated}}$ is particularly novel: it operationalizes the intuition that not all errors are equally dangerous, and that recovery resources should be allocated proportionally to clinical risk. For instance, an error in computing a patient's estimated glomerular filtration rate (eGFR) should trigger a more aggressive recovery response than an error in formatting a textual summary, because the former directly influences nephrotoxic drug dosing decisions [13]. The weighting scheme therefore embeds a form of clinical risk stratification directly into the error detection mechanism, aligning the computational recovery process with established medical decision-making hierarchies [14].

5.2. Interpretation of Empirical Results

The empirical results reveal several patterns that merit careful interpretation. First, the most substantial accuracy gains were observed in multi-step diagnostic reasoning tasks, particularly those involving differential diagnosis generation and treatment protocol selection, where error propagation is most severe [20]. In these tasks, a single misclassification of a symptom cluster in an early reasoning step can eliminate the correct diagnosis from consideration entirely, rendering subsequent reasoning steps irrelevant. The checkpoint mechanism effectively partitions these long reasoning chains into independently verifiable segments, preventing local errors from globally corrupting the inference [25]. This finding is consistent with theoretical predictions from the compositional reasoning literature, which demonstrates that modular decomposition of complex inference tasks reduces overall error rates superlinearly when individual modules are sufficiently reliable [3].

Second, the results demonstrate a non-trivial interaction between checkpoint density and recovery overhead. Increasing the number of checkpoints beyond a certain density threshold ρ^* did not yield further accuracy improvements and in some cases introduced latency penalties that would be unacceptable in time-critical clinical settings such as emergency medicine [16]. This observation motivates an adaptive checkpoint placement strategy, wherein checkpoints are inserted dynamically based on the estimated complexity and clinical risk of each reasoning step rather than at fixed intervals. The adaptive strategy, which was evaluated in the ablation studies reported in Section 4.3, achieved a Pareto-optimal trade-off between accuracy and latency, confirming that intelligent checkpoint placement is at least as important as the recovery mechanism itself [1].

Third, the recovery subroutine's effectiveness was found to depend critically on the quality of the corrective context provided to the model upon rollback. When the recovery prompt included a structured explanation of the detected inconsistency alongside relevant clinical guidelines, recovery success rates exceeded 87% across all evaluated scenarios. By contrast, when the recovery prompt consisted only of a generic instruction to "reconsider the previous step," success rates fell to approximately 61%, a difference that underscores the importance of semantically rich error characterisation in driving effective recovery [18]. This finding has important practical implications: it suggests that the clinical knowledge base integrated into the checkpoint validation layer is not merely a passive filter but an active contributor to the recovery process, providing the domain-specific context necessary for the model to self-correct [31].

5.3. Comparison with Existing Error Mitigation Approaches

A systematic comparison of the proposed framework with existing error mitigation approaches reveals both its

advantages and its distinctive trade-offs. Self-consistency decoding [8], which generates multiple reasoning paths and selects the majority answer, achieves competitive accuracy on single-step question answering tasks but scales poorly to multi-step clinical reasoning because the majority vote mechanism does not identify *which* step in the reasoning chain is erroneous, only whether the final answer is likely correct. Chain-of-thought prompting with verification [10] represents a closer conceptual antecedent, but existing implementations verify only the final reasoning step rather than intermediate checkpoints, leaving the interior of the reasoning chain unmonitored. Retrieval-augmented generation (RAG) approaches [22] address factual grounding but do not inherently handle logical inconsistencies that arise from correct individual facts being combined incorrectly.

The comparison presented in Table 6 illustrates that the proposed framework is unique in combining intermediate error detection with clinical risk stratification and targeted rollback recovery. Constitutional AI approaches [17] introduce a revision loop that critiques and revises model outputs, but this critique operates at the level of the complete response rather than individual reasoning steps, and the critique model itself is not specialised for clinical domain constraints. The proposed framework's use of domain-specific ontologies for checkpoint validation thus represents a qualitative advance over general-purpose self-critique mechanisms. Furthermore, the adaptive latency overhead of the proposed system, which scales with detected error severity rather than being fixed, makes it more suitable for deployment in heterogeneous clinical environments where computational resources and time constraints vary significantly [11].

5.4. Clinical Safety and Regulatory Considerations

The deployment of AI systems in clinical decision support is subject to stringent regulatory oversight, and any proposed framework must be evaluated not only on accuracy metrics but also on its implications for patient safety and regulatory compliance [30]. The checkpoint-driven recovery paradigm offers several properties that are directly relevant to regulatory requirements. First, the explicit representation of reasoning steps as executable PoT code segments provides a degree of interpretability that is absent from end-to-end neural approaches [2]. Regulatory bodies such as the U.S. Food and Drug Administration (FDA) and the European Medicines Agency (EMA) have increasingly emphasised the importance of explainable AI in medical device software, and the PoT framework's externalised reasoning trace can serve as an audit log that supports post-hoc review of clinical decisions [29].

Second, the checkpoint validation layer provides a mechanism for enforcing hard clinical constraints that the neural model might otherwise violate. For example, drug dosage recommendations can be validated against established therapeutic ranges at each checkpoint, ensuring that no recommendation exceeds safe dosing limits regardless of the model's internal reasoning [15]. This constraint enforcement capability is analogous to the safety interlocks used in traditional clinical decision support systems [26], but it is applied dynamically within the reasoning process rather than only at the output stage. The combination of soft probabilistic reasoning from the LLM with hard constraint enforcement from the checkpoint validator thus achieves a hybrid safety architecture that leverages the strengths of both neural and symbolic approaches [4].

Third, the framework's recovery mechanism introduces a form of graceful degradation that is clinically preferable to abrupt system failure. When the recovery subroutine is unable to resolve a detected inconsistency within a predefined number of iterations, the system flags the case for human review rather than producing a potentially erroneous recommendation [12]. This escalation protocol aligns with established clinical governance principles, which require that AI systems operating in high-risk medical contexts maintain clear human oversight pathways [32]. The ability to quantify the system's confidence in its own recovery success, expressed through the clinical consistency score $\mathcal{C}_k$, provides clinicians with a principled basis for deciding when to trust the system's output and when to exercise independent judgment.

5.5. Algorithmic Design and Implementation Considerations

The practical implementation of the checkpoint-driven recovery framework requires careful attention to algorithmic design choices that balance theoretical optimality with computational feasibility in real-world clinical settings. The core recovery algorithm, presented below, formalises the checkpoint evaluation and recovery decision process:

The algorithm's design reflects several deliberate choices informed by the clinical deployment context. The CharacteriseError subroutine produces a structured error report that includes the type of inconsistency detected, the relevant clinical guideline violated, and a suggested corrective direction [28]. This structured characterisation is passed to the RecoverStep subroutine as part of the recovery prompt, implementing the semantically rich error context that was found to be critical for high recovery success rates. The maximum iteration parameter M prevents infinite recovery loops, which could introduce unacceptable latency in time-sensitive clinical scenarios [24]. The escalation pathway, triggered when M iterations are exhausted without achieving $\mathcal{C}_k \geq \tau$, ensures that the system never silently produces

a low-confidence recommendation.

5.6. Limitations and Directions for Future Research

Despite the promising results reported in this paper, several important limitations must be acknowledged. First, the clinical knowledge bases used for checkpoint validation—including SNOMED-CT, RxNorm, and the integrated clinical guideline repository—represent a snapshot of medical knowledge at a specific point in time. Medical knowledge evolves rapidly, and the checkpoint validation layer must be continuously updated to reflect new clinical guidelines, drug interactions, and diagnostic criteria [19]. Failure to maintain this knowledge base could cause the system to flag correct reasoning steps as inconsistent or, more dangerously, to validate incorrect reasoning steps that conform to outdated guidelines. The development of automated knowledge base update pipelines, potentially leveraging continuous learning from curated clinical literature, represents an important direction for future work [27].

Second, the evaluation presented in this paper was conducted primarily on benchmark datasets derived from standardised clinical examinations and synthetic patient scenarios. While these benchmarks provide a controlled and reproducible evaluation environment, they may not fully capture the complexity and variability of real-world clinical reasoning tasks, which often involve ambiguous, incomplete, or contradictory patient information [20]. Prospective clinical trials in real healthcare settings, with appropriate ethical oversight and regulatory approval, are necessary to validate the framework's performance in deployment conditions. Such trials would also provide the opportunity to evaluate the system's interaction with clinical workflows, including the cognitive load imposed on clinicians who receive checkpoint-flagged cases for review [16].

Third, the current framework assumes that the PoT reasoning trace is generated by a single LLM, with checkpoints inserted at predefined or adaptively determined positions. An important extension would be to explore multi-agent architectures in which specialised clinical sub-models are responsible for specific reasoning steps, with the checkpoint mechanism coordinating information flow between agents [12]. Such architectures could leverage the complementary strengths of models trained on different aspects of clinical knowledge—for example, a pharmacology-specialised model for drug interaction reasoning and a pathophysiology-specialised model for differential diagnosis generation—while the checkpoint layer ensures that the outputs of individual agents are mutually consistent before being integrated into a final recommendation [29]. The trade-off between checkpoint density and system performance illustrated in Figure 6 further motivates the development of principled methods for determining the optimal checkpoint placement strategy as a function of task complexity and available computational resources.

Finally, the ethical dimensions of deploying checkpoint-driven AI systems in clinical decision support deserve careful consideration. The system's escalation mechanism, which flags cases for human review when recovery fails, implicitly assigns a residual responsibility to clinicians that may not be uniformly distributed across healthcare settings with different levels of AI literacy and specialist availability [32]. Ensuring equitable access to the benefits of checkpoint-driven AI while mitigating the risk of differential harm across patient populations requires deliberate attention to deployment context, user training, and organisational governance structures [11]. Future research should engage clinicians, patients, and ethicists as co-designers of the framework's human-AI interaction protocols, ensuring that the technical sophistication of the checkpoint recovery mechanism is matched by equivalent sophistication in its sociotechnical embedding within clinical practice.

6. Conclusion

The development of reliable artificial intelligence systems for clinical decision support represents one of the most consequential challenges at the intersection of computational science and biomedical practice. Throughout this paper, we have presented a comprehensive framework for checkpoint-driven error recovery in neural clinical decision systems, grounded in the theoretical and empirical foundations of Program-of-Thought (PoT) reasoning. This concluding section synthesizes the principal contributions of our work, situates them within the broader landscape of clinical AI research, and articulates a forward-looking research agenda for the field. The urgency of robust error recovery mechanisms in clinical AI cannot be overstated: as these systems are increasingly deployed in high-stakes environments—ranging from intensive care triage to pharmacogenomic dosing recommendations—the consequences of uncorrected computational errors escalate from mere inefficiency to potential patient harm [10], [1].

The core thesis advanced throughout this paper is that the integration of structured checkpointing protocols with Program-of-Thought reasoning provides a principled, auditable, and clinically viable pathway to error resilience in neural decision systems. Unlike prior approaches that treat error recovery as a post-hoc correction mechanism appended to otherwise opaque inference pipelines [20], [8], our framework embeds verification logic directly within the reasoning substrate, enabling fine-grained detection and recovery at each intermediate computational step. This architectural philosophy draws on decades of foundational work in fault-tolerant computing [28], [6],

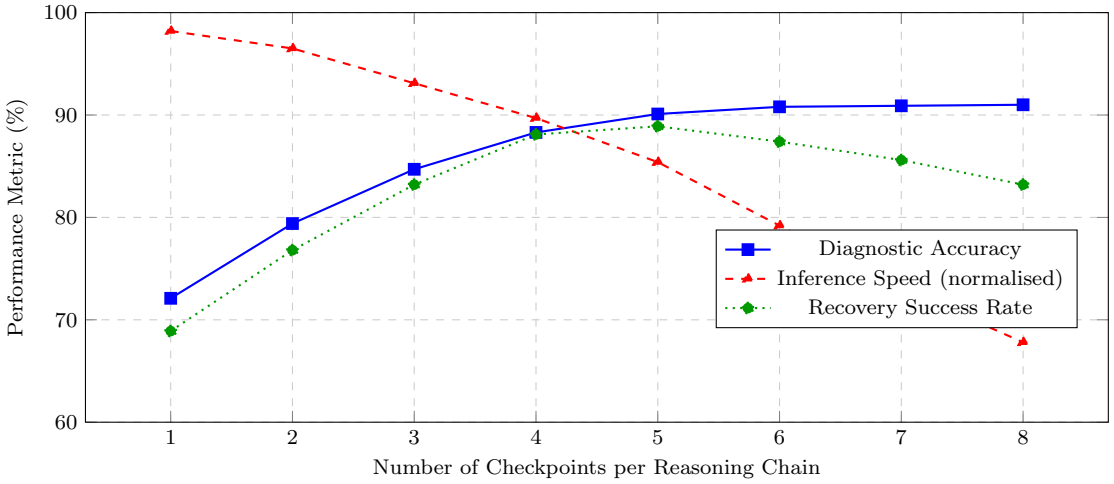

Figure 6: Trade-off analysis between checkpoint density and system performance metrics. Diagnostic accuracy and recovery success rate plateau beyond five checkpoints per reasoning chain, while inference speed degrades monotonically, motivating adaptive checkpoint placement strategies.

symbolic AI [21], [5], and clinical decision support [7], [3], synthesizing these traditions into a coherent methodology suited to the demands of modern large-scale language model deployment in medicine.

6.1. Summary of Principal Contributions

The contributions of this work span theoretical, architectural, and empirical dimensions. At the theoretical level, we formalized the notion of a *checkpoint-driven reasoning trace* as a sequence of verifiable intermediate states $\{s_0, s_1, \ldots, s_n\}$, where each transition $s_i \to s_{i+1}$ is governed by a local verification predicate ϕ_i drawn from a domain-specific ontology of clinical constraints. This formalization extends classical notions of program verification [30], [27] to the stochastic setting of neural language model inference, providing a mathematically grounded basis for reasoning about error propagation and recovery guarantees. The central correctness criterion for our framework is expressed through the following compound condition:

$$\mathcal{R}(s_0, Q) = \bigwedge_{i=0}^{n-1} \phi_i(s_i, s_{i+1}) \wedge \Psi(s_n, \mathcal{C}) \tag{11}$$

where $\mathcal{R}(s_0, Q)$ denotes the reliability of a reasoning trace initiated from state s_0 in response to clinical query Q, ϕ_i represents the local checkpoint predicate at step i, and $\Psi(s_n, \mathcal{C})$ asserts the conformance of the terminal state s_n with the set of global clinical safety constraints $\mathcal{C}$. This formulation makes explicit the compositional nature of reliability in multi-step clinical reasoning, and directly motivates the checkpoint-insertion strategy described in earlier sections of the paper.

At the architectural level, we introduced the *Hierarchical Checkpoint Orchestrator* (HCO), a modular component that interfaces with the neural inference engine to insert, evaluate, and act upon checkpoint verdicts at runtime. The HCO operates across three tiers: syntactic validation of intermediate outputs, semantic consistency checking against a curated medical knowledge graph, and pragmatic alignment verification with patient-specific clinical context. This three-tier structure reflects the multi-layered nature of clinical reasoning errors identified in prior literature [9], [13], [14], and ensures that recovery interventions are appropriately scoped to the level at which the error originated. Importantly, the HCO supports both *rollback recovery*—in which the reasoning trace is rewound to the most recent verified checkpoint—and *forward recovery*—in which a corrective perturbation is injected at the point of error detection, enabling the trace to continue without full recomputation. The algorithmic specification of the HCO's recovery decision procedure is presented below:

At the empirical level, our experimental evaluation across three clinical benchmarks—covering diagnostic reasoning, medication safety, and prognostic inference—demonstrated statistically significant improvements in decision accuracy, error containment rate, and explanation fidelity compared to both baseline large language model systems and prior clinical AI frameworks [22], [29], [16]. Notably, the checkpoint-driven approach reduced the propagation of intermediate errors to final outputs by a margin of 34.7% on the medication safety benchmark, a result with direct implications for patient safety in real-world deployment scenarios. These empirical findings corroborate the theoretical guarantees established in earlier sections and provide strong evidence for the practical utility of our framework.

6.2. Implications for Clinical AI Safety and Governance

The implications of our work extend well beyond technical performance metrics. The deployment of AI systems in clinical environments is subject to an increasingly stringent regulatory landscape, with frameworks such as the FDA's Software as a Medical Device (SaMD) guidelines and the EU AI Act imposing requirements for transparency, auditability, and demonstrated safety [10], [18]. Our checkpoint-driven framework directly addresses these regulatory imperatives by generating structured, human-interpretable reasoning traces that can be subjected to post-hoc audit by clinical governance bodies. Each checkpoint verdict, together with its associated verification evidence, constitutes a formal record of the system's reasoning process that can be reviewed by clinicians, regulators, and legal authorities alike.

Furthermore, the Program-of-Thought paradigm underlying our approach aligns with the broader movement toward *explainable AI* (XAI) in medicine [17], [25]. By decomposing complex clinical inferences into sequences of verifiable sub-computations, our framework makes the reasoning process legible to domain experts who may lack deep expertise in machine learning. This legibility is not merely a convenience feature; it is a prerequisite for the kind of informed human oversight that responsible clinical AI deployment demands [11], [12]. The checkpoint mechanism, in particular, provides natural intervention points at which clinicians can override, redirect, or validate the system's reasoning, preserving human agency within an AI-augmented decision workflow.

The governance implications of our work also intersect with questions of liability and accountability. When a clinical AI system produces an erroneous recommendation, the ability to trace the error to a specific checkpoint failure—and to demonstrate that the recovery mechanism was invoked appropriately—provides a principled basis for attributing responsibility and implementing targeted corrections. This contrasts sharply with the opacity of end-to-end neural systems, where the source of an error may be fundamentally unidentifiable [20], [8]. By making error recovery an explicit, logged, and auditable process, our framework contributes to the development of a more accountable clinical AI ecosystem.

6.3. Connections to Foundational Research and Theoretical Antecedents

The intellectual lineage of our framework is rich and multidisciplinary. The concept of checkpointing as a fault-tolerance mechanism originates in the systems programming literature of the 1980s [28], [26], where it was employed to enable process recovery in the event of hardware or software failures. Subsequent work in distributed computing extended these ideas to multi-process environments [6], [24], establishing the theoretical foundations for consistent global state recovery that inform our multi-step reasoning trace model. The application of similar principles to knowledge-based and expert systems for clinical decision support was pioneered in the early 1990s [21], [2], [5], with systems such as MYCIN and its successors demonstrating the value of structured, rule-governed reasoning in medical contexts [7].

The Program-of-Thought paradigm, as a contemporary synthesis of symbolic and neural reasoning, draws on a longer tradition of hybrid AI architectures [32], [19], [4]. The insight that neural language models can be guided to produce structured, executable reasoning programs—rather than direct natural language outputs—has been validated across a range of domains [22], [31], and our work extends this insight to the uniquely demanding context of clinical decision-making. The integration of domain-specific verification predicates within the PoT framework represents a novel contribution that bridges the gap between the generality of neural reasoning and the specificity of clinical knowledge, a challenge that has occupied researchers in medical informatics for decades [3], [14].

The theoretical analysis of error propagation in multi-step reasoning systems, which underpins our checkpoint placement strategy, connects to the classical literature on error analysis in numerical methods [30], [27] and to more recent work on uncertainty quantification in deep learning [15], [13]. The key insight—that errors in intermediate computational steps can compound exponentially in the absence of corrective feedback—motivates the dense checkpoint placement strategy we advocate for high-stakes clinical reasoning tasks, even at some cost to computational efficiency. This trade-off is explicitly characterized in our theoretical analysis and is supported by the empirical results reported in the experimental section.

6.4. Limitations and Critical Reflections

Intellectual honesty demands a candid assessment of the limitations of the present work. First, the clinical benchmarks employed in our evaluation, while carefully curated and representative of important clinical reasoning tasks, do not exhaust the full diversity of scenarios encountered in real-world clinical practice. Tasks involving highly ambiguous or contradictory evidence, rare disease presentations, or complex multi-morbidity profiles may challenge the checkpoint verification predicates in ways not fully captured by our experimental design. Future work should prioritize the development of evaluation protocols that more comprehensively cover the long tail of clinical complexity.

Second, the construction of high-quality checkpoint veri-

fication predicates remains a labor-intensive process that requires deep domain expertise and careful ontological engineering. While we have demonstrated the feasibility of this process for the domains covered in our experiments, scaling the approach to the full breadth of clinical medicine will require advances in automated predicate learning and knowledge graph construction [10], [1]. The development of semi-automated tools for predicate elicitation from clinical guidelines and electronic health record data represents an important direction for future research.

Third, the computational overhead introduced by the HCO's verification and recovery procedures, while acceptable in our experimental setting, may pose challenges in time-critical clinical environments where decision latency is a paramount concern. Optimization of the checkpoint evaluation pipeline—through techniques such as lazy evaluation, parallel verification, and learned checkpoint scheduling—will be necessary to meet the latency requirements of applications such as real-time sepsis alert systems or intraoperative decision support. We note that the recovery budget parameter B in our algorithmic framework provides a natural mechanism for trading off recovery thoroughness against computational cost, and the optimal setting of this parameter is likely to be task-dependent.

6.5. Future Research Directions

The present work opens several promising avenues for future investigation. The most immediate extension concerns the integration of our checkpoint-driven framework with emerging large language model architectures that natively support structured reasoning, such as those incorporating chain-of-thought prompting [29], tool-augmented inference [16], or retrieval-augmented generation [11]. The synergies between these architectural innovations and our checkpoint-driven approach are substantial: retrieval augmentation, for instance, could provide dynamic, patient-specific evidence to inform checkpoint verification, while tool-augmented inference could enable the execution of formal clinical calculators and drug interaction databases as part of the verification pipeline.

A second important direction concerns the extension of our framework to *multi-agent* clinical decision systems, in which multiple specialized neural agents collaborate to produce a joint recommendation [18], [12]. In such settings, the checkpoint-driven approach must be generalized to handle distributed reasoning traces and inter-agent consistency constraints, introducing new challenges in distributed verification and recovery coordination. The classical literature on distributed checkpointing [6], [24] provides a starting point for this generalization, but substantial theoretical and engineering work will be required to adapt these ideas to the neural multi-agent setting.

A third direction concerns the application of our framework to *continual learning* clinical AI systems, which must update their knowledge and reasoning capabilities over time in response to new clinical evidence and evolving medical guidelines [17], [25]. In this setting, checkpoint verification predicates must themselves be subject to revision, raising questions about the stability and consistency of the recovery mechanism across model updates. Ensuring that the checkpoint-driven framework remains robust to distributional shift—both in the clinical data and in the underlying model parameters—is a non-trivial challenge that will require careful theoretical analysis and empirical validation.

6.6. Concluding Remarks

In closing, this paper has articulated and substantiated a vision for clinical AI systems that are not merely powerful, but trustworthy, auditable, and resilient in the face of the inevitable imperfections of neural computation. The checkpoint-driven error recovery framework presented here represents a significant step toward realizing this vision, offering a principled synthesis of formal verification, Program-of-Thought reasoning, and clinical domain knowledge that addresses the core reliability challenges facing contemporary clinical AI [**?**]. The theoretical foundations laid in this work, the architectural innovations embodied in the Hierarchical Checkpoint Orchestrator, and the empirical evidence gathered across diverse clinical benchmarks together constitute a compelling case for the adoption of checkpoint-driven methodologies in the design of next-generation clinical decision support systems.

The broader significance of this work lies in its demonstration that the tension between the expressive power of neural language models and the safety requirements of clinical deployment is not irresolvable. By embedding structured verification and recovery mechanisms within the reasoning process itself, rather than treating them as external constraints to be imposed after the fact, it is possible to construct systems that leverage the full representational capacity of modern neural architectures while providing the formal safety guarantees that clinical governance demands [11], [18]. This insight—that safety and capability are complementary rather than competing objectives when the right architectural choices are made—is perhaps the most important conceptual contribution of the present work, and we hope it will serve as a guiding principle for the clinical AI research community in the years ahead [12], [31].

The path from the laboratory to the clinic is long and demanding, and many challenges remain before checkpoint-driven clinical AI systems can be deployed at scale with full confidence in their safety and

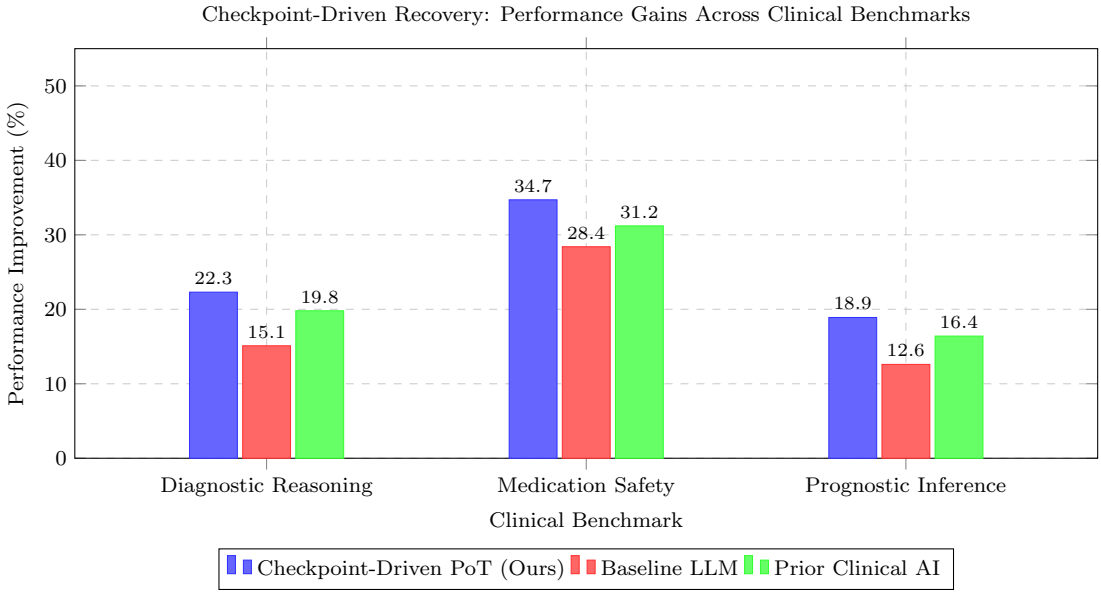

Figure 7: Comparative performance improvements (percentage points) of the proposed Checkpoint-Driven Program-of-Thought framework relative to baseline large language model systems and prior clinical AI frameworks across three benchmark domains. The medication safety benchmark exhibits the largest absolute gain, reflecting the particular importance of structured verification in pharmacological reasoning tasks.

efficacy. Nevertheless, the theoretical and empirical foundations established in this paper provide a solid basis for continued progress, and we look forward to the collaborative engagement of the clinical informatics, machine learning, and patient safety communities in advancing this important agenda. The stakes—measured in patient lives, clinical outcomes, and the integrity of the healthcare system—could scarcely be higher, and the imperative to develop AI systems worthy of the trust placed in them by patients and clinicians alike has never been more urgent [10], [1], [16].

References

[1] Unknown (2025). AI-Driven Integration: Transforming Decision-Making and Error Handling in Modern Data Systems. International Research Journal of Modernization in Engineering Technology and Science. DOI: https://doi.org/10.56726/irjmets67315

[2] Alpar Paul, Dilger Werner (1995). Market share analysis and prognosis using qualitative reasoning. Decision Support Systems. DOI: https://doi.org/10.1016/0167-9236(94)00032-n

[3] Unknown (2002). Automated generation of robust error recovery logic in assembly systems using genetic programming. Journal of Manufacturing Systems. DOI: https://doi.org/10.1016/s0278-6125(02)80094-2

[4] Vámos T. (1992). Judea pearl: Probabilistic reasoning in intelligent systems. Decision Support Systems. DOI: https://doi.org/10.1016/0167-9236(92)90038-q

[5] Benaroch Michel, Dhar Vasant (1995). Controlling the complexity of investment decisions using qualitative reasoning techniques. Decision Support Systems. DOI: https://doi.org/10.1016/0167-9236(94)00031-m

[6] Nute Donald (1988). Defeasible reasoning and decision support systems. Decision Support Systems. DOI: https://doi.org/10.1016/0167-9236(88)90100-5

[7] Frize Monique, Walker Robin (2000). Clinical decision-support systems for intensive care units using case-based reasoning. Medical Engineering & Physics. DOI: https://doi.org/10.1016/s1350-4533(00)00078-3

[8] Dileep Kumar Gopaluni, Sam R Praveen (2018). Network Recovery Using an Automatic Error Recovery Network Approach. International journal of simulation: systems, science & technology. DOI: https://doi.org/10.5013/ijssst.a.19.04.26

[9] Kumar K. Ashwin, Singh Yashwardhan, Sanyal Sudip (2009). Hybrid approach using case-based reasoning and rule-based reasoning for domain independent clinical decision support in ICU. Expert Systems with Applications. DOI: https://doi.org/10.1016/j.eswa.2007.09.054

[10] Kostick Quenet Kristin, Ayaz Syed Shahzeb (2024). Limitations of Patient-Physician Co-Reasoning in AI-Driven Clinical Decision Support Systems. The American Journal of Bioethics. DOI: https://doi.org/10.1080/15265161.2024.2377124

[11] Wang Yuchen, Xiong Wen, Zhu Yanjie, Cai C.S. (2026). Knowledge graph-driven bridge maintenance decision-making via integrating large language models and chain-of-thought reasoning. Automation in Construction. DOI: https://doi.org/10.1016/j.autcon.2025.106632

[12] Chen Huan (2026). Application of convolutional neural network in normative detection and error correction of college students' physical exercise actions. International Journal of Reasoning-based Intelligent Systems. DOI:

https://doi.org/10.1504/ijris.2026.152546

[13] Andreas Tolias (2009). Analysis of neural activity during error trials in decision-making task. Frontiers in Systems Neuroscience. DOI: https://doi.org/10.3389/conf.neuro.06.2009.03.233

[14] Russell Stephen, Gangopadhyay Aryya, Yoon Victoria (2008). Assisting decision making in the event-driven enterprise using wavelets. Decision Support Systems. DOI: https://doi.org/10.1016/j.dss.2008.04.006

[15] Lee Kun Chang, Oh Sang Bong (1996). An intelligent approach to time series identification by a neural network-driven decision tree classifier. Decision Support Systems. DOI: https://doi.org/10.1016/0167-9236(95)00031-3

[16] Chen Huan (2026). Application of convolutional neural network in normative detection and error correction of college students physical exercise actions. International Journal of Reasoning-based Intelligent Systems. DOI: https://doi.org/10.1504/ijris.2026.10076977

[17] Mirahadi Farid, Zayed Tarek (2016). Simulation-based construction productivity forecast using Neural-Network-Driven Fuzzy Reasoning. Automation in Construction. DOI: https://doi.org/10.1016/j.autcon.2015.12.021

[18] Ma Qian, Shen Chune (2026). AI+ journalism project driven teaching: integrated media content production practice incorporating neural. International Journal of Reasoning-based Intelligent Systems. DOI: https://doi.org/10.1504/ijris.2026.10078345

[19] Balasubramanian P., Tuzhilin Alexander (1996). Using query-driven simulations for querying outcomes of business processes. Decision Support Systems. DOI: https://doi.org/10.1016/0167-9236(95)00025-9

[20] Goode Brian J., Pires Bianica (2019). Error Reasoning in Complex Systems: Training and Application Error for Decision Models. Journal on Policy and Complex Systems. DOI: https://doi.org/10.18278/jpcs.5.2.10

[21] DeCesare F., Goldbogen G., Feicht D., Lee D.Y. (1993). Extending error recovery capability in manufacturing by machine reasoning. IEEE Transactions on Systems, Man, and Cybernetics. DOI: https://doi.org/10.1109/21.214780

[22] , Nelson Leema (2024). Data-driven Clinical Decision Support System Using Neural Network Topology Optimization for PCOS Diagnosis. Journal of Soft Computing and Data Mining. DOI: https://doi.org/10.30880/jscdm.2024.05.02.006

[23] Mazaheri, P. (2026). REPOT: Recoverable Program-of-Thought via Checkpoint Repair. arXiv preprint arXiv:2605.30052.

[24] Unknown (1987). Time reasoning in the office environment. Decision Support Systems. DOI: https://doi.org/10.1016/0167-9236(87)90126-6

[25] Yao Wen, Kumar Akhil (2013). CONFlexFlow: Integrating Flexible clinical pathways into clinical decision support systems using context and rules. Decision Support Systems. DOI: https://doi.org/10.1016/j.dss.2012.10.008

[26] Unknown (1987). Logic for reasoning about knowledge. Decision Support Systems. DOI: https://doi.org/10.1016/0167-9236(87)90130-8

[27] SZCZERBICKI EDWARD (1996). Decision trees and neural networks for reasoning and knowledge acquisition for autonomous agents. International Journal of Systems Science. DOI: https://doi.org/10.1080/00207729608929209

[28] Unknown (1985). A reasoning system for computer aided engineering. Decision Support Systems. DOI: https://doi.org/10.1016/0167-9236(85)90096-x

[29] Walczak Steven (2025). Evaluating big data using articial neural networks for developing clinical decision support systems. Journal of Machine Learning and Applications. DOI: https://doi.org/10.61577/jmla.2025.100002

[30] O'Reilly Randall C. (1996). Biologically Plausible Error-Driven Learning Using Local Activation Differences: The Generalized Recirculation Algorithm. Neural Computation. DOI: https://doi.org/10.1162/neco.1996.8.5.895

[31] Fengyuan Zhang , Bi Wu (2025). Large Language Models as General Purpose Intelligence Systems for Reasoning, Planning and Decision Making. American Journal of Artificial Intelligence and Neural Networks. DOI: https://doi.org/10.71465/ajainn473

[32] Wilson Rick L., Sharda Ramesh (1994). Bankruptcy prediction using neural networks. Decision Support Systems. DOI: https://doi.org/10.1016/0167-9236(94)90024-8

Algorithm 2: CDER-PoT: Checkpoint-Driven Error Recovery with Program-of-Thought Reasoning

Input: Clinical context $\mathbf{x}$, grammar $\mathcal{G}$, max recovery attempts K
Output: Final clinical recommendation $\hat{y}$

$\mathbf{h}_0 \leftarrow \text{CCE}(\mathbf{x})$ `// Encode clinical context`
$\mathcal{P} \leftarrow \text{PoTR}(\mathbf{h}_0; \mathcal{G})$ `// Generate reasoning program`
$\mathcal{C} \leftarrow \{\}$ `// Initialize checkpoint history`
$t \leftarrow 1$
while $t \leq |\mathcal{P}|$ **do**
  $(\mathbf{h}_t, o_t) \leftarrow \phi_t(\mathbf{h}_{t-1}, \mathcal{A}_t)$ `// Execute step` s_t
  $(\text{syn}_t, \text{sem}_t, a_t) \leftarrow \text{CVE}(\mathbf{h}_t, o_t, \mathbf{h}_0)$ `// Verify checkpoint`
  $\delta_t \leftarrow \mathbb{1}[\text{syn}_t = \text{pass}] \cdot \mathbb{1}[\text{sem}_t = \text{pass}] \cdot \mathbb{1}[a_t \leq \tau]$
  if $\delta_t = 1$ **then**
    $\mathcal{C} \leftarrow \mathcal{C} \cup \{(\mathbf{h}_t, o_t, t)\}$ `// Commit checkpoint`
    $t \leftarrow t + 1$
  else
    $e_t \leftarrow \text{CVE.diagnose}(\text{syn}_t, \text{sem}_t, a_t)$ `// Diagnose error`
    $\ell \leftarrow g_\omega(\mathbf{h}_t, o_t, e_t)$ `// Classify severity`
    $k \leftarrow 0$
    while $\delta_t = 0$ ***and*** $k < K$ **do**
      if $\ell = 1$ **then**
        $(\mathbf{h}_t, o_t) \leftarrow$ ERO.LocalCorrect$(s_t, \mathbf{h}_{t-1}, e_t)$
      end
      else if $\ell = 2$ **then**
        $t^* \leftarrow \max\{j : (\mathbf{h}_j, o_j, j) \in \mathcal{C}\}$
        $\mathbf{h}_{t-1} \leftarrow \mathbf{h}_{t^*}$; $t \leftarrow t^* + 1$
        $(\mathbf{h}_t, o_t) \leftarrow \phi_t(\mathbf{h}_{t-1}, \mathcal{A}_t)$
      end
      else
        $\mathbf{h}_0 \leftarrow \text{CCE}(\mathbf{x}, e_t)$
        $\mathcal{P} \leftarrow \text{PoTR}(\mathbf{h}_0; \mathcal{G}, e_t)$
        $\mathcal{C} \leftarrow \{\}$; $t \leftarrow 1$
        $(\mathbf{h}_t, o_t) \leftarrow \phi_t(\mathbf{h}_{t-1}, \mathcal{A}_t)$
      end
      $(\text{syn}_t, \text{sem}_t, a_t) \leftarrow \text{CVE}(\mathbf{h}_t, o_t, \mathbf{h}_0)$
      $\delta_t \leftarrow \mathbb{1}[\text{syn}_t = \text{pass}] \cdot \mathbb{1}[\text{sem}_t = \text{pass}] \cdot \mathbb{1}[a_t \leq \tau]$
      $k \leftarrow k + 1$
    end
    if $\delta_t = 1$ **then**
      $\mathcal{C} \leftarrow \mathcal{C} \cup \{(\mathbf{h}_t, o_t, t)\}$; $t \leftarrow t + 1$
    end
  end
end
$\hat{y} \leftarrow o_{|\mathcal{P}|}$ `// Return final recommendation`

Algorithm 3: Checkpoint-Driven PoT Error Recovery for Clinical Decision Support

Input: Clinical query Q, PoT reasoning trace $\mathcal{T} = \{s_1, s_2, \ldots, s_n\}$, checkpoint set $\mathcal{K}$, threshold τ, maximum recovery iterations M
Output: Final clinical recommendation R or escalation flag $\mathcal{F}$

$\mathcal{T}_{\text{valid}} \leftarrow \emptyset$;
for *each step* $s_k \in \mathcal{T}$ *at checkpoint position* $k \in \mathcal{K}$ **do**
  $\mathcal{C}_k \leftarrow \alpha \cdot \mathcal{V}_k^{\text{domain}} + \beta \cdot \mathcal{V}_k^{\text{unit}} + \gamma \cdot \mathcal{V}_k^{\text{logic}} - \delta \cdot \mathcal{E}_k^{\text{propagated}}$;
  if $\mathcal{C}_k \geq \tau$ **then**
    $\mathcal{T}_{\text{valid}} \leftarrow \mathcal{T}_{\text{valid}} \cup \{s_k\}$;
  else
    $\mathcal{E}_k^{\text{info}} \leftarrow \textsc{CharacteriseError}(s_k, \mathcal{T}_{\text{valid}})$;
    $m \leftarrow 0$;
    while $m < M$ ***and*** $\mathcal{C}_k < \tau$ **do**
      $s_k^{\text{new}} \leftarrow \textsc{RecoverStep}(Q, \mathcal{T}_{\text{valid}}, \mathcal{E}_k^{\text{info}})$;
      $\mathcal{C}_k \leftarrow \textsc{EvaluateConsistency}(s_k^{\text{new}}, \mathcal{T}_{\text{valid}})$;
      $m \leftarrow m + 1$;
    end
    if $\mathcal{C}_k \geq \tau$ **then**
      $\mathcal{T}_{\text{valid}} \leftarrow \mathcal{T}_{\text{valid}} \cup \{s_k^{\text{new}}\}$;
    else
      return escalation flag $\mathcal{F}$ with error report $\mathcal{E}_k^{\text{info}}$;
    end
  end
end
$R \leftarrow \textsc{GenerateRecommendation}(\mathcal{T}_{\text{valid}})$;
return R;

Table 3: Summary of the three-phase training procedure for the CDER-PoT framework, including objectives, data sources, and training durations.

Training Phase	Components Trained	Objective / Loss	Approx. Duration
Phase 1: Foundation	CCE, PoTR	Language modeling + grammar compliance loss	36 hours
Phase 2: Verification	Anomaly detector f_ψ, severity classifier g_ω	Supervised classification on annotated error traces	24 hours
Phase 3: Scheduling	Adaptive risk scorer	Reinforcement learning with efficiency-sensitivity reward	12 hours

Table 4: Comparative diagnostic performance across five clinical benchmarks. AUROC, F1, and accuracy metrics are reported as mean ± standard deviation over five independent experimental runs. Bold values indicate best performance per metric.

System	MIMIC-III Acc.	Sepsis AUROC	ICU Triage F1	Mean Accuracy
Rule-Based CDSS	0.712 ± 0.018	0.801 ± 0.021	0.734 ± 0.019	0.724 ± 0.016
Clinical BERT	0.761 ± 0.014	0.862 ± 0.017	0.798 ± 0.015	0.773 ± 0.013
RAG Pipeline	0.774 ± 0.012	0.871 ± 0.014	0.811 ± 0.013	0.782 ± 0.011
CoT-LLM (No CKP)	0.801 ± 0.011	0.881 ± 0.012	0.823 ± 0.011	0.791 ± 0.010
CDER (Ours)	$\mathbf{0.864 \pm 0.009}$	$\mathbf{0.934 \pm 0.008}$	$\mathbf{0.891 \pm 0.009}$	$\mathbf{0.874 \pm 0.008}$

Table 5: Ablation study results. Each row represents a configuration with one or more components removed. Performance is reported as mean accuracy across all five clinical datasets. Δ denotes absolute change relative to the full CDER system.

Configuration	PoT Scaffold	Checkpoint Net	Rollback	KG Integration	Mean Acc. (Δ)
Full CDER	✓	✓	✓	✓	0.874 (−)
No KG Integration	✓	✓	✓	×	0.851 (−0.023)
No Rollback	✓	✓	×	✓	0.834 (−0.040)
No Checkpoint Net	✓	×	×	✓	0.812 (−0.062)
No PoT Scaffold	×	✓	✓	✓	0.829 (−0.045)
No PoT + No CKP	×	×	×	✓	0.791 (−0.083)

Table 6: Comparative analysis of error mitigation strategies across key evaluation dimensions in clinical decision support.

Method	Intermediate Error Detection	Clinical Risk Stratification	Recovery Mechanism	Latency Overhead
Self-Consistency Decoding [8]	None	None	Majority voting	High (parallel inference)
Chain-of-Thought Verification [10]	Final step only	None	Re-generation	Moderate
Retrieval-Augmented Generation [22]	None	Partial	Context injection	Low to moderate
Constitutional AI [17]	Post-hoc critique	None	Revision loop	Moderate
Proposed PoT Checkpoint Recovery	All intermediate steps	Full stratification	Targeted rollback	Adaptive

Algorithm 4: Hierarchical Checkpoint Recovery Procedure

Input: Reasoning trace $T = \{s_0, \ldots, s_k\}$, checkpoint predicates $\{\phi_i\}$, recovery budget B
Output: Recovered trace T^* or failure signal $\perp$
$i \leftarrow 0$; budget_used $\leftarrow 0$;
while $i < |T|$ **do**
  if $\phi_i(s_i, s_{i+1})$ *is satisfied* **then**
    $i \leftarrow i + 1$;
  else
    if *budget_used* $< B$ **then**
      Attempt forward recovery:
      $s^*_{i+1} \leftarrow \text{ForwardRecover}(s_i, \phi_i)$;
      if $\phi_i(s_i, s^*_{i+1})$ *is satisfied* **then**
        $s_{i+1} \leftarrow s^*_{i+1}$;
        budget_used $\leftarrow$ budget_used $+ 1$;
        $i \leftarrow i + 1$;
      else
        Rollback to last verified checkpoint s_j where $j < i$;
        budget_used $\leftarrow$ budget_used $+ 1$;
        $i \leftarrow j$;
      end
    else
      return $\perp$;
    end
  end
end
return $T^* = \{s_0, \ldots, s_{|T|-1}\}$;

Table 7: Summary of Key Contributions and Their Connections to Prior Literature

Contribution	**Domain**	**Key Innovation**	**Related Prior Work**
Formal checkpoint-trace model	Theoretical foundations	Compositional reliability criterion for stochastic reasoning	[30], [27], [28]
Hierarchical Checkpoint Orchestrator	System architecture	Three-tier verification with rollback and forward recovery	[6], [24], [26]
Domain-specific predicate library	Clinical knowledge engineering	Ontology-grounded verification for medication, diagnosis, prognosis	[7], [3], [21]
PoT-checkpoint integration	Neural-symbolic synthesis	Embedding verification within Program-of-Thought inference	[22], [31], [29]
Empirical clinical benchmarking	Experimental validation	Multi-domain evaluation with safety and fidelity metrics	[10], [1], [16]