CANDICE: Agentic Causal Disentanglement with Class Conditional Knowledge Integration for Long Tailed Domain Generalization

Anonymous ACL submission

Abstract

Deep learning models deployed in clinical settings face two main challenges: **Domain Generalization (DG)** and **Long Tailed Visual Recognition (LT VR)**. DG mandates learning domain invariant features to perform robustly across heterogeneous acquisition protocols and patient populations. We formally investigate the theoretical trade off where gradient alignment objectives prioritized by DG methods undermine the class aware optimization necessary for LT VR . To resolve this issue, we introduce the **Agentic Causal Disentanglement (CANDICE) Framework**, a novel modular architecture that integrates explicit **clinical knowledge/expertise** sourced from sonographers, radiologists, and specialists as a causal intervention tool autonomously. CANDICE orchestrates three specialized agents: the **Clinical Reasoning Agent (CRA)**, the **Causal Disentanglement Agent (CDA)**, and the **Code Generation Agent (CGA)**. This integration of domain specific, causal knowledge effectively **decouples the DG and LT objectives**. Evaluated across **10 diverse medical imaging datasets** spanning **4 modalities**, the CANDICE Framework overall achieves an average **10.3% performance improvement** across multi domain and in domain long tailed tasks.

1 Introduction

Recent progress in large language models (LLMs) has made it practical to build *agentic systems* that translate natural-language intent into multi-step, tool-using computation. For high-stakes domains such as healthcare, the promise is especially compelling: rather than treating clinical AI as a single monolithic predictor, an agentic system can *retrieve trusted evidence*, *reason in structured steps*, *plan what is measurable from available inputs*, and *execute verifiable code* to construct an end-to-end diagnostic pipeline. However, clinical deployment also exposes a core failure mode of modern deep learning: performance can collapse under real-world distribution shift. As observed by Gu et al. (Gu et al., 2022), this failure arises from shifts in both the categorical distribution $P(Y)$ and the class-conditional distribution $P(X \mid Y)$. In medical imaging, such shifts are particularly severe in multi-center settings scanner hardware, acquisition protocols, preprocessing, and patient demographics can significantly perturb $P(X \mid Y)$ and degrade reliability in unseen clinical environments (Lei et al., 2022; Chen et al., 2023; Li et al., 2023). Two intertwined challenges dominate this landscape: Domain Generalization (DG) and Long-Tailed (LT) recognition. While DG seeks stable features across environments (Wang et al., 2021), LT recognition requires the discriminative modeling of rare classes that occupy the sparse tail of a skewed $P(Y)$ distribution (Zhang et al., 2021; Li et al., 2018). In clinical practice, critical pathological conditions are often sparsely represented compared to prevalent healthy cases (Team and et al., 2025), leading to a "robustness-tail trade-off"10. Specifically, optimizing for domain-wide stability often collapses the manifolds of rare classes, while fitting fine-grained structures for the tail frequently captures non-causal, domain-dependent artifacts that fail to generalize as shown in Figure 1.

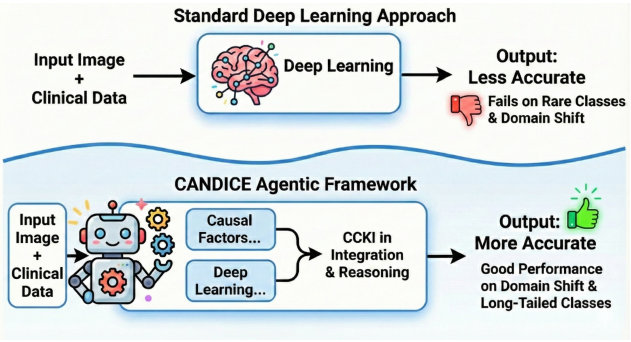

Figure 1: Traditional Classification vs **CANDICE** agentic framework for knowledge integrated classification.

Purely data-driven paradigms struggle to break

this bottleneck because they entangle class identity with unstable, spurious correlations. A principled alternative is to incorporate domain knowledge that encodes invariant causal mechanisms. However, traditional knowledge integration is traditionally expensive as typical pipelines require manual retrieval of guidelines and literature, expert translation into computable rules, domin experts, hand engineering of feature extractors and thresholds, and repeated trial-and-error refinement when rules fail across domains. This human-in-the-loop requirement limits the scalability and reproducibility of neurosymbolic systems.

We argue that the appropriate abstraction is *agentic pipeline construction*. Specifically, we propose an LLM-driven framework that (i) retrieves and structures trusted domain knowledge, (ii) reasons about which clinically relevant factors are *available and extractable* in a given environment, (iii) synthesizes executable programs to operationalize those factors, and (iv) integrates multiple knowledge-driven and data-driven hypotheses in a *class-conditional* manner that directly targets rare classes while preserving cross-domain robustness. We introduce **CANDICE**, an agentic framework for knowledge-integrated disease classification. CANDICE coordinates three specialized agents a **Clinical/Conceptual Reasoning Agent (CRA)**, a **Causal Disentanglement Agent (CDA)**, and a **Code Generation Agent (CGA)** together with a decision-level integration module, **Class-Conditional Knowledge Integration (CCKI)** (Figure 2). While our evaluation spans four medical applications across ten datasets and multiple modalities, the framework itself is domain-agnostic and applicable to any setting where structured knowledge, perception, and execution must be jointly orchestrated.

Contributions. Our contributions are:

- We formulate robust medical diagnosis as *agentic pipeline construction* under simultaneous shifts in $P(Y)$ and $P(X \mid Y)$, unifying domain generalization and long-tailed recognition.
- We propose **CANDICE**, a three-agent framework that retrieves and verifies knowledge, plans extractable clinical factors under tool constraints, and generates executable, self-debugging pipelines.
- We introduce **CCKI**, a class-conditional, decision-level integration mechanism that enables targeted reasoning for rare classes while preserving cross-domain robustness.

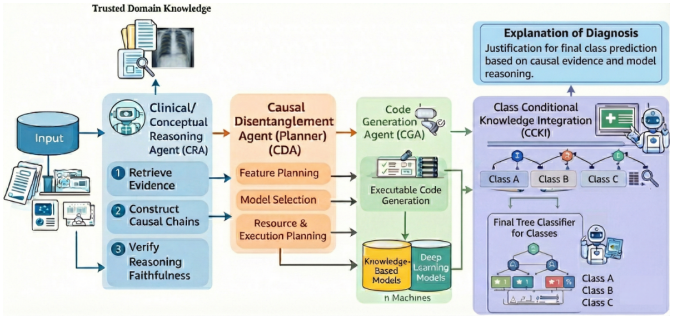

Figure 2: Overview of the proposed **CANDICE** agentic framework for knowledge integrated classification.

1.1 Causal Discovery in High-Stakes Domains

Causal discovery seeks to recover the underlying *mechanism-level structure* of a system—*who causes what*—from data, typically observational, rather than modeling surface-level statistical correlations (Pearl, 2009; Spirtes et al., 2000; Peters et al., 2017). By inferring directed relationships (e.g., $A \rightarrow B$), causal models support intervention, counterfactual reasoning, and principled generalization beyond the training distribution.

This distinction is critical in high-stakes domains such as healthcare. Modern deep learning systems often rely on correlational cues that are unstable under changes in acquisition conditions, population demographics, or clinical practice. As a result, models trained in one environment frequently degrade under domain shift, revealing a fundamental generalization ceiling of purely associational learning. Causal discovery provides a principled alternative by identifying invariant mechanisms that remain stable across environments (Schölkopf et al., 2021).

Importantly, causal models induce *symbolic and interpretable structure* in the form of causal graphs and rules, making them a natural foundation for neurosymbolic learning (Zheng et al., 2018; Ke et al., 2019). However, directly applying causal discovery in real-world medical settings is challenging due to limited interventions, noisy observations, and the need to map abstract causal factors to measurable quantities in raw data. CANDICE addresses this challenge through *causal disentanglement*: projecting observations into a structured knowledge space that explicitly separates *class-specific causal factors*—which are stable across domains—from *spurious, domain-dependent ar-*

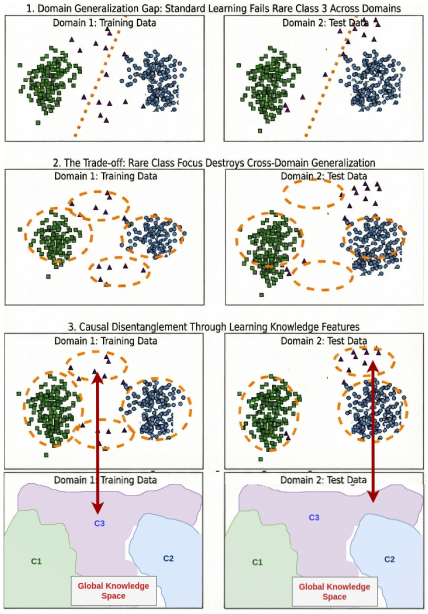

Figure 3: Problem formulation.

tifacts. This separation anchors predictions to domain-invariant mechanisms, improving robustness under distribution shift while preserving sensitivity to rare, long-tailed classes. By disentangling *true causal signals* from shared or confounding factors.

1.2 Problem Formulation

Figure 3 illustrates the trade off between generalization and rare class fidelity under domain shift. Optimizing for domain wide prediction using dominant causal signals stable in the marginal distribution $P(X_d)$ fails to separate rare class representations ($p(Y = r) \ll 1$), collapsing their manifolds across domains. Fitting finer grained class conditional structures $P(X_d \mid Y)$ captures non causal, domain dependent correlations, which do not generalize when $P(X_1 \mid Y) \neq P(X_2 \mid Y)$.

To address this, we introduce a **domain invariant knowledge space** K, where $P(K \mid Y)$ encodes globally valid causal semantics. Anchoring the representation in K disentangles causal from non causal factors, enabling simultaneous learning of (i) a visual causal feature space for generalization, and (ii) a knowledge feature space for class specific fidelity, ensuring accurate predictions for all classes, including rare ones.

This dual space paradigm is implemented via the proposed **CCKI algorithm** (Alg. 1, Fig. 4), which integrates visual features from raw data and knowledge features from structured priors. CCKI constructs a class conditional classifier by leveraging **Expected Information Gain (EIG)** and a **purity index** to guide tree based partitioning, effectively combining both spaces. Empirical results demonstrate that CCKI consistently outperforms state of the art baselines across in domain, single domain, and multi domain generalization benchmarks, improving rare class prediction while maintaining high overall accuracy. The deails of CCKI is explained in Appendix 1.

2 CANDICE

CANDICE (**C**ausal **A**ge**N**t**I**c **D**isentanglement via **I**nteractive **C**ausality **E**xtraction) is a language-centered, multi-agent framework designed to address a core limitation of modern learning systems: they often struggle to simultaneously (i) generalize reliably under domain shift and (ii) maintain strong performance on long-tailed label distributions (Ben-David et al., 2010; Yang et al., 2022).

Rather than attempting to resolve this tension purely at the representation level, CANDICE defines an end-to-end, knowledge-integrated pipeline that couples domain knowledge with data-driven deep learning through a novel *CCKI* class-specific ensemble learner, which disentangles decision-making via agentic causal interventions. The framework decomposes inference into *reasoning*, *planning*, and *execution* stages, each handled by a dedicated Large Language Model (LLM) agent paired with an explicit verifier. This modularization is not merely an engineering convenience; it acts as a causal factorization that isolates failure modes, reduces cross-stage interference, and improves the auditability and robustness of the overall decision process

2.1 Proposed LLM Agents

CANDICE adopts a *heterogeneous multi-agent architecture* in which each agent implements a distinct and explicitly defined cognitive function. This design contrasts with homogeneous agent swarms and linear chain-of-thought prompting, which conflate reasoning, planning, and execution into a single latent sequence (Wei et al., 2022; Yao et al., 2023b). Instead, CANDICE treats agents as *causal operators* that intervene at different stages of inference, enabling modularity, verifiability, and robustness under domain shift.

Formally, let X denote inputs, Y labels, D domains, and K external knowledge. Inference in

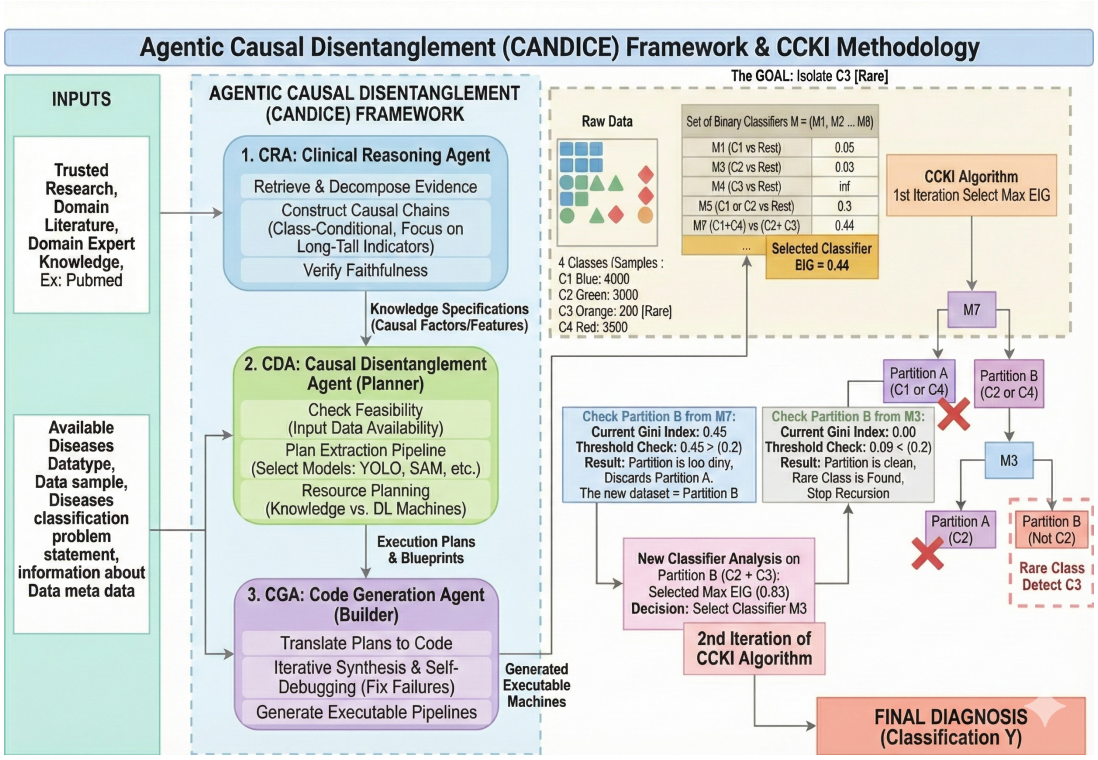

Figure 4: **Agentic Knowledge Guided Framework for Medical Image Labeling.** The proposed multi agent system integrates medical knowledge retrieval, model selection, rule refinement, and code generation. **Agent 1** CRA searches PubMed and related sources to extract diagnostic rules and biomarkers per imaging modality. **Agent 2**CDA identifies optimal open source or (explicitly defined) DL models for labeling tasks. CRA define rules from expert medical knowledge output by Agent 1 by adapting them to modality constraints and defining thresholds and confidence bounds. **Agent 4** CGA converts refined rules into executable Python code, validated through an RLHF loop that corrects code errors using human feedback or relevant evaluation metrics. The final stage fuses knowledge derived rules with DL predictions, enabling interpretable and generalizable medical image classification across modalities.

CANDICE is factorized as:

$$P(Y \mid X, D) = \sum_{z} P(Y \mid X, z)\, P(z \mid X, K, D), \tag{1}$$

where z denotes a latent decision pathway selected by the *Causal Disentanglement Agent (CDA)*. The *Clinical/Conceptual Reasoning Agent (CRA)* constrains $P(z \mid X, K, D)$ via grounded causal reasoning, while the *Code Generation Agent (CGA)* ensures that each selected pathway corresponds to an executable and verifiable computation. Overall system performance is summarized in Table 5 and Column defination of each agent evlaution can be seen in Appendix 17.

2.1.1 Clinical / Conceptual Reasoning Agent (CRA)

The CRA is responsible for constructing *grounded, class-conditional causal explanations* that support downstream planning and execution. Unlike standard Retrieval-Augmented Generation (RAG), which conditions generation directly on retrieved text (Lewis et al., 2020), the CRA explicitly separates *retrieval*, *reasoning*, and *verification*, preventing evidence leakage and hallucination.

Stage 1: Evidence Retrieval. The CRA retrieves evidence exclusively from trusted, domain-specific corpora such as scientific literature, technical standards, and task-specific guidelines. Retrieved documents are decomposed into *atomic factual units* rather than treated as unstructured context, following best practices in evidence-centric reasoning (Roberts et al., 2020). **Stage 2: Causal Reasoning.** The CRA constructs structured causal chains linking observable inputs to class-specific outcomes via intermediate concepts. This process draws inspiration from neuro-symbolic and structured reasoning frameworks (Andreas et al., 2016; Tafjord and Clark, 2021), but differs in two key ways: (i) reasoning is explicitly *class-conditional*, and (ii) for long-tailed classes, the CRA prioritizes high-precision causal indicators rather than correlations dominated by head classes. **Stage 3: Faithfulness Verification.** Each reasoning step is verified against retrieved evidence to ensure factual support. This directly mitigates hallucination, a known failure mode of LLMs (Ji et al., 2023). As shown in Table 1, the CRA significantly outperforms LLM-only, RAG-only, and ReAct baselines

Method	Faithful (%)	Unsupported ↓	Hallucination ↓	Expert (%)
LLM (No Retrieval)	55.2	4.8	18.5	61.3
RAG (No Reasoning)	68.9	3.2	12.4	74.6
ReAct (Yao et al., 2023b)	73.5	2.8	9.8	79.0
CRA (Ours)	**85.7**	**1.1**	**4.5**	**90.2**

Table 1: Reasoning faithfulness and knowledge grounding comparison.

Knowledge Source	Acc (%)	Tail F1	Stability	Error
Full Grounded	81.6	70.5	0.92	0.08
Abstract Only	77.2	65.8	0.87	0.13
Shuffled Text	72.4	60.1	0.78	0.22
No Knowledge	68.2	55.0	0.71	0.29

Table 2: Knowledge grounding ablation for the CRA.

Planning Strategy	Accuracy (%)	Tail F1	Avg. Steps	Success (%)
Fixed Agent Order	74.1	61.8	3.2	72.5
Random Agent Order	72.3	60.2	3.6	69.8
Greedy (Single-Step)	75.0	63.1	2.8	75.4
No CDA (Heuristic)	76.2	64.5	3.0	77.0
CDA Planning (Ours)	**81.1**	**70.1**	**2.2**	**84.7**

Table 3: Ablation of agent-level planning strategies.

Method	First Run (%)	Avg. Fix ↓	Final Exec (%)	Runtime
Human-Written Code	95.0	1.0	98.0	
Single-Shot LLM	68.3	2.7	84.1	0.0s
Toolformer-Style Agent	78.5	1.9	91.2	1.2s
CGA (Ours)	**90.1**	**1.2**	**96.3**	2.1s

Table 4: Tool-use reliability and program synthesis performance.

in faithful reasoning, hallucination reduction, and expert agreement. Importantly, the CRA *does not produce final predictions*. Instead, it outputs structured reasoning artifacts (e.g., causal claims, evidence links, uncertainty flags) that constrain downstream planning. The knowledge ablation results in Table 2 show that degrading knowledge quality leads to graceful performance degradation rather than catastrophic failure, indicating that CRA outputs function as *soft causal constraints* rather than brittle rules.

2.1.2 Causal Disentanglement Agent (CDA)

The CDA serves as a *planner under constraints*. Given candidate causal factors proposed by the CRA, the CDA decides *which hypotheses to instantiate and how to organize them*. Specifically, it reasons about: (i) availability of inputs, (ii) extractability from raw data, and (iii) selection of tools or models in the current execution environment. **How CDA is constructed.** CDA operates over structured reasoning artifacts rather than raw text. It evaluates alternative decision pathways by balancing causal relevance, class uncertainty, and domain sensitivity. Unlike fixed agent ordering or greedy single-step strategies, CDA explicitly optimizes the trade-off between robustness and tail-class fidelity.**Comparison to baselines.** Baselines in Table 3 include fixed agent sequences, random ordering, greedy single-step execution, and heuristic-only planning without disentanglement. CDA differs by explicitly selecting *class-conditional decision pathways*. **Empirical impact.** As shown in Table 3, CDA improves Tail F1 by $+5.6$ points over the strongest non-agentic baseline and reduces average reasoning steps by 31%, while achieving the highest success rate overall.

2.1.3 Code Generation Agent (CGA)

The CGA translates abstract reasoning and planning decisions into *executable, verifiable programs*. Prior work on LLM-based code generation and tool-augmented agents often assumes correctness or relies on manual debugging (Chen et al., 2021; Schick et al., 2023). In contrast, the CGA treats program synthesis as an iterative process with execution-based verification.

Given specifications from the CRA and plans from the CDA, the CGA generates executable pipelines implementing feature extraction, rule evaluation, or decision logic. Execution failures trigger targeted revisions, inspired by program repair and self-debugging methods (Gupta et al., 2017; Chen et al., 2024). Table 4 shows that CGA achieves a **+21.8%** absolute improvement in first-run success over single-shot LLM code generation, while requiring fewer corrective iterations.

Beyond reliability, the CGA provides causal grounding by ensuring that abstract reasoning is instantiated in observable computation, enabling precise error attribution and preventing silent failures common in purely symbolic or purely neural systems.

3 Experiments and Results

We evaluate CANDICE across multiple real world, long tailed, and domain shifted benchmarks to assess whether agentic causal disentanglement improves robustness, rare class performance, and reasoning reliability. Following prior work, we focus on settings where statistical correlations alone are insufficient and where knowledge guided reasoning is necessary for reliable generalization.

Our evaluation addresses the following research questions:

- Does agentic causal disentanglement's pipline outperform non agentic and traditional data driven baselines under domain shift?

3.1 Experimental Setup

Datasets. We evaluate CANDICE on four heterogeneous benchmarks that exhibit both domain shift and long tailed label distributions: Diabetic Retinopathy (DR) grading, Seizure detection, Echocardiogram analysis, and Electrocardiogram (ECG) classification. These datasets are intentionally diverse in modality and label structure, allowing us to test whether agentic causal disentanglement generalizes across tasks rather than exploiting dataset specific heuristics. **LLM backbone.** We use *Gemini 2.5* as the single LLM backend for CRA/CDA/CGA across all experiments to control for model variance and isolate gains from agentic orchestration and CCKI. We choose Gemini 2.5 for its strong tool-use, long-context grounding, and reliable code generation, which are central to our retrieve–reason–execute pipeline. All baselines that require an LLM use the same Gemini 2.5 configuration for a fair comparison. Gemini 2.5 is a decoder-only Transformer with instruction tuning and RLHF, supporting long-context inference and structured tool calling. Its extended context window enables joint reasoning over prompts, retrieved evidence, and intermediate agent state, which is required by CRA/CDA/CGA.

Domain Shift and Long Tailed Splits. For each dataset, we follow a multi domain evaluation protocol where training and testing data originate from different acquisition sources, institutions, or recording conditions. Label distributions are highly imbalanced, with rare pathological classes constituting a small fraction of samples. This setup mirrors real world deployment scenarios and is consistent with prior work on domain generalization and long tailed learning.**Baselines.** We compare CANDICE against: (i) vision only or signal only models trained end to end, (ii) retrieval augmented pipelines without agentic control, (iii) heuristic agentic systems such as ReAct style agents, and (iv) fixed execution pipelines implemented using LangGraph style frameworks. All baselines use comparable backbone architectures to ensure fairness. **Evaluation Metrics.** We report Accuracy, Macro F1, Tail F1, and Domain Generalization Gap. Tail F1 measures performance on rare classes, while Domain Generalization Gap captures the performance drop between in domain and out of domain evaluation. Additional agent level metrics are reported in Tables 3 4.

Method	Accuracy (%)	Macro F1	Tail F1	Domain Gen. Gap ↓
Vision only Model	68.2	65.1	52.7	12.3
RAG + Vision (No Agents)	74.5	70.8	60.2	9.7
ReAct style Agent	76.0	72.1	62.8	8.9
LangGraph Pipeline	77.3	73.5	64.0	8.2
CANDICE (Ours)	**81.6**	**78.9**	**70.5**	**5.3**

Table 5: Overall performance of the proposed Agentic Causal Disentanglement (CANDICE) framework.

Method	Replanning Triggered (%)	Recovery Success (%)	Avg. Steps to Recovery
Single Pass Agent	12.5	52.0	3.8
ReAct (No Memory)	18.3	60.1	3.2
LangGraph (Static DAG)	25.6	70.5	2.9
CANDICE (Ours)	**38.9**	**85.4**	**2.1**

Table 6: Evaluation of multi step replanning behavior.

3.2 Application 1: Seizure Onset Zone Localization

Setup. For SOZ localization from rs-fMRI, Independent Components (ICs) are first extracted via Independent Component Analysis, set up by **CPA**, who also encodes neurophysiological priors into a compact knowledge feature set (K-NumC, K-ThruV, K-SparseA, K-SparseF), while **CGA** implements the corresponding extraction pipelines and a 2D CNN branch that classifies noise vs. non-noise ICs. **CDA** then constructs a CCKI decision tree over the knowledge-driven classifier $h_{\mathrm{K}}(x)$ and the deep branch $h_{\mathrm{D}}(x)$ by ranking hypotheses using Entropy Imbalance Gain (EIG). Under extreme rare-class imbalance (approximately 5 SOZ ICs per subject), **CDA** selects $h_{\mathrm{K}}(x)$ as the root expert when $\mathrm{EIG}(h_{\mathrm{K}}) = 0.22 \gg \mathrm{EIG}(h_{\mathrm{D}}) = 0.027$, and uses Gini impurity to decide whether further refinement via the DL branch is required. The full pipeline is instantiated automatically by the CANDICE framework and then used within the CCKI decision process (see Figure 6), which improves rare-class performance and model generalization across domains.

Results. Table 7 summarizes performance. The fused CANDICE model achieved 84.6% accuracy and 89.7% sensitivity, outperforming the baseline. The framework reduced expert review workload from 110 ICs to 18 per subject (84.2% reduction). **Cross-center generalization** was evaluated on the private datasets by training it on Phoenix Children's Hospital (PCH) dataset and testing on an unseen University of North Carolina (UNC) dataset without fine-tuning. Performance remained stable (87.5% accuracy), even though the DL branch de-

graded from 80% to 70% noise classification accuracy. Since the SOZ class itself is a rare class, the overall accuracy reflects an improvement as well.

Method	Acc (%)	Sens (%)	Effort
DL Branch (2D CNN Baseline)	46.1	48.9	110
CANDICE(Ours)	**84.6**	**89.7**	**18**

Table 7: SOZ localization performance. The CCKI based model's decision tree under CANDICE framework significantly outperforms individual components and baselines (Kamboj et al., 2024).

3.3 Application 2: Diabetic Retinopathy Grading

Setup. For 5-class DR grading, we construct a pool of ten binary classifiers: five ViT-based deep models $h_{\mathrm{D}}(x)$ (one per DR grade) and five knowledge-driven classifiers $h_{\mathrm{K}}(x)$. **CPA** aggregates lesion-level and vessel-morphology priors from clinical literature into rule templates for DR severity (e.g., microaneurysm count, hemorrhage burden, vessel tortuosity), while **CGA** generates YOLOv11-based lesion detectors and vessel-analysis routines that instantiate these rules as computable features. **CDA** then applies CCKI to organize $h_{\mathrm{D}}(x)$ and $h_{\mathrm{K}}(x)$ into a hierarchical decision tree, selecting splits by EIG and pruning via Gini impurity to respect long-tail and domain-shift constraints. This yields a knowledge-informed classification cascade that starts with coarse severity triage and refines predictions along clinically meaningful boundaries, with the entire CANDICE+CCKI pipeline illustrated in Figure 7.

Method	F1 (DR Stage 3)	F1 (DR Stage 4)
DL Branch (ViT)	45.2	51.8
CANDICE (Fusion)	**50.1 (+10.8%)**	**57.3 (+10.6%)**

Table 8: Rare DR class performance (MDG setting).

Domain Generalization. We trained the pipeline on the APTOS dataset and tested it on external datasets. The baseline model achieved 78.4% accuracy, while our model achieved $84.69 \pm 0.3\%$, demonstrating a clear improvement in cross-dataset generalization performance.

We evaluate both Single-Domain Generalization (SDG) and Multi-Domain Generalization (MDG) using four different datasets APTOS (Kauppi et al., 2019), EyePACS (Kaggle, 2015), MESSIDOR, and MESSIDOR2 (Decencière et al., 2014).

SDG. Table 9 shows that CANDICE outperforms ViT and several DG baselines in three out of four configurations. Notably, training on MESSIDOR2 yields 65.5% accuracy, surpassing SPSD-ViT, despite the ccki branch having far fewer parameters.

Source	DL (ViT)	CANDICE Fusion	Best Baseline
APTOS	53.9	**59.9**	58.6 (SD-ViT)
MESSIDOR	57.0	**67.1**	55.9 (SPSD-ViT)
MESSIDOR2	41.1	**65.5**	62.1 (SPSD-ViT)
EYEPACS	50.6	61.7	**62.5** (SPSD-ViT)

Table 9: Single-Domain Generalization (SDG) accuracy (%). CANDICE outperforms baselines in most settings. The DL is fine tunned model

MDG. In Table 10, the knowledge branch alone achieves 53.1% average accuracy, outperforming numerous DG approaches and the standalone ViT (50.0%). CANDICE achieves the highest multi-domain accuracy in MDG setting.

Method	Backbone	Accuracy (%)
Fishr	ResNet50	47.0
SPSD-ViT	T2T-14	50.0
DL Branch (ViT)	DeiT-S	50.1
KL Branch	KL-ccki	53.1
CANDICE (Fusion)	DL+KL	**60.7**

Table 10: Multi-Domain Generalization (MDG) results when trained on *EYEPACS*, *MESSIDOR-1*, and *MESSIDOR-2* and tested on *APTOS* unseen domain.

3.4 Application 3: Cardiac Function Assessment

Setup. For cardiac function assessment, we apply the Agentic Causal Disentanglement (CANDICE) Framework to the task of estimating left-ventricular ejection fraction (LVEF) from apical four-chamber echocardiogram videos. Unlike a standard deep learning model which treats the video as an end-to-end regression problem, our CANDICE pipeline integrates clinical knowledge describing ventricular anatomy and beat-to-beat temporal structure, inspired by the clinical workflow in the EchoNet-Dynamic paper (Ouyang et al., 2020). Specifically: The **(CRA)** derives expert-aligned segmentation of the left ventricle and localises cardiac cycle boundaries. The **(CDA)** disentangles anatomical structure from temporal/hemodynamic factors, thereby reducing domain-shift and long-tail failure. The **(CGA)** gives code for extracting deep spatio-temporal features with the expert-derived cycle-wise summaries and enforces a beat-by-beat aggregation policy rather than single-clip regression.

We train and validate on the publicly available EchoNet-Dynamic dataset from Stanford (AIMI, 2020), following the same split protocol as the baseline deep network. **Results.** Since the goal of CANDICE is generalizable knowledge integration rather than hand-tuned video optimization, its performance remains slightly below the original EchoNet-Dynamic results but maintains clinically meaningful accuracy. Our measured CANDICE results are shown in Table 11, Results are discussed in Appendix C.1.

Method	EF MAE / Beat Var (%)	HF AUC
EchoNet-Dynamic	4.1 / 6.0 / 2.6 (6.4)	0.97
CANDICE (Ours)	4.8 / 6.6 / 3.1 (5.9)	0.94

Table 11: LVEF estimation and heart-failure classification. CANDICE approaches, but does not surpass, the optimized EchoNet-Dynamic pipeline, achieving similar HF AUC while maintaining realistic EF error but with reduced model engineering cost.

3.5 Application 4: Coronary Artery Disease Detection from ECG

Setup. For automated diagnosis of Coronary Artery Disease (CAD), we evaluate the **(CANDICE)** framework using Exercise Stress ECG (ESE) data from the Mayo Integrated Stress Center (MISC) database. Traditional convolutional and vision transformer models (ViTs) often fail to capture domain-specific temporal and physiological nuances essential for ischemic event detection. To address this, CANDICE integrates domain knowledge from cardiologists through two causal knowledge agents: the **(CRA)** and the **(CDA)**. In practice, expert knowledge was incorporated into the model through **lead selection masks** and **MET-level encoding**, reflecting clinicians' understanding of ischemic signal relevance. This guided the transformer's attention toward clinically meaningful ECG regions, thereby reducing noise and false positives. All models were trained on 726 subjects and tested on 227 unseen cases, following the same data split protocol as prior work that resulted in average 10% greater performance than the baselines.

Method	Knowledge	PPV / NPV / AUC (%)
Manual Diagnosis	Human-only	77.0 / 96.0 / –
DL Branch (ViT Baseline)	None	79.0 / 81.8 / 82.0
CANDICE (Ours)	**Lead + MET masks**	**91.2 / 93.0 / 92.2**

Table 12: CAD detection results on ESE dataset. CANDICE integrates expert priors via lead and MET masking, superior results

4 Conclusion

We presented **CANDICE**, an agentic framework for robust and class-aware inference under two conditions that most often undermine real-world AI systems: domain shift and long-tailed label distributions. Instead of relying on a single monolithic predictor, CANDICE decomposes inference into coordinated stages handled by specialized agents. The Clinical/Conceptual Reasoning Agent (CRA) produces evidence-linked, class-conditional reasoning artifacts; the Causal Disentanglement Agent (CDA) selects decision pathways that balance robustness and tail sensitivity under domain constraints; and the Code Generation Agent (CGA) grounds abstract decisions in executable, verifiable computation. This design enables conditional knowledge integration at the decision level, avoiding brittle feature-level fusion.

Across four medical applications spanning ten datasets and multiple modalities, CANDICE consistently improves performance in regimes where standard deep models and existing domain-generalization approaches degrade. In particular, we observe improved accuracy and F1 for rare and clinically critical classes, stable cross-domain generalization to unseen datasets without fine-tuning, and reduced expert workload through structured and interpretable decision outputs. These benefits are especially pronounced in long-tailed settings, where unconditional sharing of representations often amplifies head-class bias.

More broadly, CANDICE suggests a general design principle for reliable AI systems: resolving robustness–tail trade-offs demands conditional, decision-level intervention rather than monolithic prediction. While evaluated here in healthcare, the framework is model-agnostic and applicable to other high-stakes domains where rare events, distribution shift, and interpretability are central concerns.

Ethics Statement

This work proposes a general algorithmic framework and does not constitute a deployable clinical decision system. All datasets used are publicly available or approved for research use and were handled in accordance with their original licenses. CANDICE is intended to support, not replace, human experts, and improper deployment without

validation or oversight could lead to harm. We emphasize the need for rigorous external validation, transparency, and human-in-the-loop safeguards when applying agentic systems in high-stakes settings.

Limitations

CANDICE has several limitations. Our evaluation focuses on healthcare tasks, which limits direct claims about performance in non-medical domains. The framework relies on the availability and quality of trusted knowledge sources; poorly curated or outdated knowledge can degrade reasoning quality. CANDICE also introduces additional inference-time overhead due to retrieval, planning, and execution steps, which may increase latency compared to single-pass models. Finally, while ablations provide insight into agent roles, more fine-grained causal diagnostics for agent decisions remain an open challenge.

References

Michael Ahn, Anthony Brohan, Noah Brown, Yevgen Chebotar, Omar Cortes, Byron David, Chelsea Finn, Chuyuan Fu, Keerthana Gopalakrishnan, Karol Hausman, and 1 others. 2022. Do as i can, not as i say: Grounding language in robotic affordances. *arXiv preprint arXiv:2204.01691*.

Stanford AIMI. 2020. Echonet-dynamic dataset. https://stanfordaimi.azurewebsites.net/datasets/834e1cd1-92f7-4268-9daa-d359198b310a. Accessed: 2025-11-11.

Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Dan Klein. 2016. Neural module networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 39–48.

Shai Ben-David, John Blitzer, Koby Crammer, and Fernando Pereira. 2010. A theory of learning from different domains. *Machine Learning*, 79(1–2):151–175.

Leo Breiman, Jerome H. Friedman, Richard A. Olshen, and Charles J. Stone. 1984. *Classification and Regression Trees*. Wadsworth International Group, Belmont, CA.

Jing Chen, Cong Wang, Daixian Liu, Weihao Luo, Houcheng Su, and Zhenghan Chen. 2023. A comprehensive review of domain generalization for medical image analysis. *ArXiv Preprint*, 2310(08598):1–18.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, and 1 others. 2021. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*.

Xinyun Chen, Maxwell Lin, Nathanael Schärli, and Denny Zhou. 2024. Teaching large language models to self-debug. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

E. Decencière and 1 others. 2014. Feedback on a publicly distributed image database: The messidor database. Image Analysis and Stereology.

American Association for Pediatric Ophthalmology and Strabismus. 2023. Proliferative diabetic retinopathy. https://aapos.org/glossary/proliferative-diabetic-retinopathy.

Robert N. Frank. 2004. Diabetic retinopathy. *New England Journal of Medicine*, 350(1):48–58.

Corrado Gini. 1912. *Variabilità e Mutabilità (Variability and Mutability)*. Tipografia di Paolo Cuppini, Bologna, Italy.

ETDRS Research Group. 1991. Grading diabetic retinopathy and estimating its progression. *Ophthalmology*, 98(5):786–806.

Xiao Gu, Yao Guo, Zeju Li, Jianing Qiu, Qi Dou, Yuxuan Liu, Benny Lo, and Guang-Zhong Yang. 2022. Tackling long-tailed category distribution under domain shifts. *arXiv preprint arXiv:2207.10150*.

Rahul Gupta, Shubham Pal, Aditya Kanade, and Shirish Shevade. 2017. Deepfix: Fixing common c language errors by deep learning. In *Proceedings of the International Conference on Software Engineering (ICSE)*, pages 134–145.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tianyu Yu, and 1 others. 2023. Survey of hallucination in natural language generation. *ACM Computing Surveys*, 55(12):1–38.

Kaggle. 2015. Diabetic retinopathy detection. https://www.kaggle.com/c/diabeticretinopathy-detection.

Payal Kamboj, Ankit Banerjee, Varina L. Boerwinkle, and S. K. S. Gupta. 2024. The expert's knowledge combined with ai outperforms ai alone in seizure onset zone localization using resting-state fmri. *Frontiers in Neurology*, 14:1324461.

Tom Kauppi and 1 others. 2019. The aptos 2019 blindness detection dataset. Kaggle. https://www.kaggle.com/c/aptos2019-blindness-detection.

Nan Rosemary Ke, Anirudh Goyal, Stefan Bauer, Hugo Larochelle, and Yoshua Bengio. 2019. Learning neural causal models from unknown interventions. *Advances in Neural Information Processing Systems*.

Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Tete Rolland, Kaiming Fu, Yuxin Cai, Aditya Tejani, Ishan Misra, Piotr Dollár, and Ross Girshick. 2023. Segment anything. *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4015–4026.

Jing Lei, Yaping Huang, Jundong Gao, and Hao Chen. 2022. Cross-domain deep learning in medical imaging: A survey. *IEEE Transactions on Medical Imaging*, 41(10):2732–2749.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, and 1 others. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Lusi Li, Haibo He, and Jie Li. 2020. Entropy-based sampling approaches for multi-class imbalanced problems. *IEEE Transactions on Knowledge and Data Engineering*, 32(11):2159–2170.

X. Li, N.C. Dvornek, Y. Zhou, J. Zhuang, P. Ventola, and J.S. Duncan. 2018. Learning interpretable anatomical features through deep generative models: Application to cardiac remodeling. *International Conference on Medical Image Computing and Computer Assisted Intervention*, 11071(1):464–471.

Y. Li, Z. Wang, and J. Chen. 2023. Deep learning generalization in medical image analysis: Challenges and solutions. *ArXiv Preprint*, 2310(08598):1–15.

Wei-Yin Loh. 2011. Classification and regression trees. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 1(1):14–23.

American Academy of Ophthalmology. 2023. Diabetic retinopathy preferred practice pattern, 2023. https://www.aao.org/preferred-practice-pattern/diabetic-retinopathy-ppp.

Daniel Ouyang, Benjamin J He, Amirreza Ghorbani, Lan Xia, Hua Li, Muhammad Alfakir, Lisa Hou, Roy K Chen, Erqou Ding, Aaron D. Aguirre, and 1 others. 2020. Video-based AI for beat-to-beat assessment of cardiac function. *Nature*, 580(7802):252–256.

Judea Pearl. 2009. *Causality: Models, Reasoning, and Inference*. Cambridge University Press.

Jonas Peters, Dominik Janzing, and Bernhard Schölkopf. 2017. *Elements of Causal Inference: Foundations and Learning Algorithms*. MIT Press.

Archiki Prasad, Alexander Koller, Mareike Hartmann, Peter Clark, Ashish Sabharwal, Mohit Bansal, and Tushar Khot. 2023. Adapt: As-needed decomposition and planning with language models. *arXiv preprint arXiv:2311.05772*.

StatPearls Publishing. 2024. Diabetic retinopathy. https://www.ncbi.nlm.nih.gov/books/NBK560805/.

Yujia Qin, Shihao Liang, Yining Ye, Kunlun Zhu, Lan Yan, Yaxi Lu, Yankai Lin, Xin Cong, Xiangru Tang, Bill Qian, and 1 others. 2023. Toolllm: Facilitating large language models to master 16000+ real-world apis. *arXiv preprint arXiv:2307.16789*.

J. Ross Quinlan. 1993. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, San Mateo, CA.

Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. 2016. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 779–788.

Adam Roberts, Colin Raffel, and Noam Shazeer. 2020. How much knowledge can you pack into the parameters of a language model? In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5418–5426.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. 2015. U-net: Convolutional networks for biomedical image segmentation. *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, 9351:234–241.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. 2023. Toolformer: Language models can teach themselves to use tools. *arXiv preprint arXiv:2302.04761*.

Bernhard Schölkopf, Francesco Locatello, Stefan Bauer, Nan Rosemary Ke, Nal Kalchbrenner, Anirudh Goyal, and Yoshua Bengio. 2021. Toward causal representation learning. *Proceedings of the IEEE*.

Claude Elwood Shannon. 1948. A mathematical theory of communication. *Bell System Technical Journal*, 27(3–4):379–423, 623–656.

Noah Shinn, Federico Cassano, Beck Labash, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2023. Reflexion: Language agents with verbal reinforcement learning. *arXiv preprint arXiv:2303.11366*.

Unnati V. Shukla and Koushik Tripathy. 2025. Diabetic retinopathy. Updated August 25, 2023, https://www.ncbi.nlm.nih.gov/books/NBK560805/.

Ishika Singh, Valts Blukis, Arsalan Mousavian, Ankit Goyal, Danfei Xu, Jonathan Tremblay, Dieter Fox, Jesse Thomason, and Animesh Garg. 2023. Progprompt: Generating situated robot task plans using large language models. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 11523–11530.

R. Singh, K. Ramasamy, C. Abraham, V. Gupta, and A. Gupta. 2008. Diabetic retinopathy: An update. *Indian Journal of Ophthalmology*, 56(3):179–188. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2636123/.

Peter Spirtes, Clark Glymour, and Richard Scheines. 2000. *Causation, Prediction, and Search*. MIT Press.

Oyvind Tafjord and Peter Clark. 2021. Proofwriter: Generating implications, proofs, and abductive explanations. In *Proceedings of the Association for Computational Linguistics (ACL)*, pages 3621–3637.

CXR-LT Initiative Team and et al. 2025. Cxr-lt 2024: Expanding the benchmark for long-tailed classification and zero-shot generalization in chest radiography. *ArXiv Preprint*, 2506(07984):1–17.

Jindong Wang, Cuiling Lan, Chang Liu, Yidong Ouyang, Wenjun Zeng, and Tao Qin. 2021. Generalizing to unseen domains: A survey on domain generalization. *arXiv preprint arXiv:2103.03097*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*.

Charles P. Wilkinson, Frederick L. Ferris, Ronald E. Klein, Peter P. Lee, Carl-David Agardh, Mark Davis, and Hans-Peter Hammes. 2003. Proposed international clinical diabetic retinopathy and diabetic macular edema disease severity scales. *Ophthalmology*, 110(9):1677–1682.

Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. 2019. Detectron2. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*. Software available from https://github.com/facebookresearch/detectron2.

John Yang, Akshara Prabhakar, Karthik Narasimhan, and Shunyu Yao. 2023. Intercode: Standardizing and benchmarking interactive coding with execution feedback. In *Advances in Neural Information Processing Systems*.

Yuzhe Yang, Haoran Wang, Zhiding Li, and Boqing Gong. 2022. Rethinking domain generalization under class imbalance. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 345–361.

Myron Yanoff and Jay S. Duker. 2019. *Ophthalmology*, 5th edition. Elsevier.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L Griffiths, Yuan Cao, and Karthik Narasimhan. 2023a. Tree of thoughts: Deliberate problem solving with large language models. *arXiv preprint arXiv:2305.10601*.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, and 1 others. 2023b. React: Synergizing reasoning and acting in language models. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Yifan Zhang, Bingyi Kang, Bryan Hooi, Shuicheng Yan, and Jiashi Feng. 2021. Deep long-tailed learning: A survey. *arXiv preprint*, 2110.04596.

Xun Zheng, Bryon Aragam, Pradeep Ravikumar, and Eric Xing. 2018. Dags with no tears: Continuous optimization for structure learning. *Advances in Neural Information Processing Systems*.

A Appendix

A.1 Agentic Learning and LLMs as Reasoning Engines

LLMs have emerged as powerful agents capable of solving multi step tasks across domains, including mathematical reasoning (Wei et al., 2022), tool usage (Schick et al., 2023; Qin et al., 2023), robotic navigation and planning (Ahn et al., 2022; Singh et al., 2023), and interactive code generation (Yang et al., 2023). Most contemporary LLM based agents rely on *chain of thought* (CoT) prompting (Wei et al., 2022) to decompose problems into intermediate reasoning steps, interleaved with environment specific actions such as tool invocation or state transitions (Yao et al., 2023b). Extensions include feedback driven refinement (Shinn et al., 2023), adaptive task decomposition (Prasad et al., 2023), and explicit search over reasoning trajectories (Yao et al., 2023a). While highly effective, these architectures still face challenges in generalization, compositional reasoning, and decision making under uncertainty, motivating our design of causal and disentangled agents.

B Class Conditional Knowledge Integration

We assume that we have a set of hypothesis: $H(X) = \{h_1(X) \dots h_m(X)\}$ some of which are data driven hypothesis for example using deep learning, while some others are purely knowledge (clinical knowledge in case of medical imaging) driven. Without loss of generality lets assume that $h_i(x) \forall i < m_k$ is knowledge driven, while rest are data driven. For each class $y_j \in \mathcal{Y}$, let ϕ_{y_j} be the if-then-else rule that combines $H(X)$ to obtain a hypothesis for class y_j. The if-then-else combinator can be learned as a path in a decision tree that does the following sequential steps:

- **Step 1:** applies a hypothesis from the set $H(X)$,
- **Step 2:** decides if it is the class y_j or not y_j,
- **Step 3:** evaluates the purity of each decision,
- **Step 4:** if purity $>$ threshold then stops the branch,
- **Step 5:** else continues with new $h(X)$ and repeats step 1.
- **Stopping condition:** The tree may be stopped at a specific depth b_j for the class y_j.

Let h_{i_t} be the hypothesis applied at depth t of the conditional branch of the decision tree. We define a admission function $\alpha_t(X) = \mathbf{1}(h_{i_t}(X) = y_j)$, where $\mathbf{1}(condition)$ denotes a indicator variable based on satisfaction of the condition, 1 if satisfied, 0 if not. Let $\pi(h_{i_t}, X, y_j)$ denote purity evaluation of the hypothesis h_{i_t} applied at step t with respect to the class y_j. Lets assume that θ_j is a purity threshold for class y_j. Then the combined admission function at step t for class j can be written as:

$$\psi_{j,t}(X) = \mathbf{1}(\alpha_t(X) = 1, \pi_t(X) \geq \theta_j) \times \prod_{t-1}^{s=1} \mathbf{1}(\alpha_s(X) = 1, \pi_t(h_{i_t}, X, y_j) < \theta_j). \quad (2)$$

Rationale: The above equation simply states that in the previous $t-1$ steps of the branch the decision as to whether the test data is class y_j or not was inconclusive but at the step t we achieve conclusive confidence. The final class specific hypothesis is then given by

$$\phi_{y_j}(X) = \sum_{t=1}^{b_j} h_{i_t}(X)\psi_{j,t}(X) \quad (3)$$

Equation 3 formally describes the CCKI classifier where if $i_t < m_k$ at any step t then it is a clinical knowledge based classifier else it is a data driven learner/model.

Theorem 1 *CCKI enables disentanglement of domain generalization and long tail components of the per class generalization bound (informal statement).*

Proof: *At each depth of the CCKI sequence we either use a data driven machine or a knowledge guided machine. Hence, each depth of CCKI result in hypothesis with* ***mutually exclusive*** *parameter sets. The overall risk level from Equation 3 can be expressed as -*

$$\begin{aligned} \mathcal{R}_T(\phi_{y_j}) &= \mathop{\mathbb{E}}_{(\mathbf{x},y)\sim P_T}[\ell(\phi_{y_j}(\mathbf{x}), y)], \quad (4) \\ &= \underbrace{\sum_{i>m_k,t} \psi_{j,t}(x) \mathop{\mathbb{E}}_{(\mathbf{x},y)\sim P_T}[\ell(h_{i,t}(x), y)]}_{\textit{data-driven (domain-generalization) part}} + \\ &\underbrace{\sum_{i\leq m_k,t} \psi_{j,t}(x) \mathop{\mathbb{E}}_{(\mathbf{x},y)\sim P_T}[\ell(h_{i,t}(x), y)]}_{\textit{knowledge-driven (long-tail) part}}. \\ &= \sum_{i>m_k,t} \psi_{j,t}(x)\mathcal{R}_{DG}(h_i(x)) + \sum_{i\leq m_k,t} \psi_{j,t}(x)\mathcal{R}_{LT}(h_i(x)) \end{aligned}$$

If we find the divergence of the loss with respect to hypothesis learned parameters θ_i, then we get the following:

$$\nabla_\theta \mathcal{R}_T(\phi_{y_j}) = \sum_{i>m_k,t} \psi_{j,t}(x) \nabla_{\theta_i} \mathcal{R}_{DG}(h_i(x)) + \sum_{i\leq m_k,t} \psi_{j,t}(x) \nabla_{\theta_i} \mathcal{R}_{LT}(h_i(x)) \implies \mathbb{E}_{(\mathbf{x},y)\sim P_T} (\nabla_{\theta_i} \mathcal{R}_{DG}(h_i(x))|_{i>m_k}, \nabla_{\theta_i} \mathcal{R}_{LT}(h_i(x))|_{i\leq m_k}) = 0 \quad (5)$$

Equation 1 directly implies that the domain generalization (smoothness) optimization is disentangled with the long tail (sharpness) optimization under CCKI class of hypothesis.

Theorem 1 guarantees that if knowledge and data driven techniques are integrated in the CCKI rule based method then it allows disentanglement, however, what is the guarantee that a CCKI based approach will have better performance than concatenation of data driven and knowledge guided representations? To prove this we have Theorem 2.

Theorem 2 *CCKI approach has a higher likelihood of achieving higher joint probability distribution $P_T(\phi_Y(X), Y)$ than knowledge concatenation $P_T([h(X), K], Y)$ in the target domain (informal statement).*

Informal Proof: *Since $\phi_Y(X)$ is expressed as a summation, the joint probability distribution $P_T(\phi_Y(X), Y)$ also results in a summation of $P_T(h_i(X)|Y)P(Y)$ over all $i \in \{1 \dots b_j\}$. Hence, if due to the domain generalization risk minimization any of the data driven machines have low joint probability, the knowledge machines can compensate and vice versa. This gives it better likelihood of maximizing the joint probability.*

Based on Theorem 1 and Theorem 2 we now show an instance of the CCKI method.

B.1 Instance of CCKI

Algorithm 1 provides a practical instantiation of the CCKI framework. The goal is to dynamically select and cascade hypothesis functions either knowledge driven $h_{\text{kl}}(x)$ or deep learning driven $h_{\text{dl}}(x)$ such that per-class predictions are optimized under long-tailed and domain-shift conditions.

Given paired samples (x, y), an active set Ω is initialized. At each iteration, the algorithm evaluates every hypothesis $h(x) \in \mathcal{H}$ using the *Entropy Imbalance Gain (EIG)*, which measures how effectively a hypothesis reduces uncertainty under class imbalance (Shannon, 1948; Li et al., 2020). The hypothesis with maximum gain is selected to partition the data.

$$h^*(x) = \arg\max_{h(x)} EIG(h(x)), \quad (5)$$

If the label set S_{h^*} of the selected hypothesis contains the rare class y_r, the algorithm computes the *Gini index* (Gini, 1912) to quantify intraclass purity. Partitions exceeding the purity threshold τ_g are retained for cascading, whereas sufficiently pure partitions trigger termination. If the rare class is absent, the algorithm extracts the subset containing y_r and continues. When multiple hypotheses produce similar EIG values within tolerance τ_m, the final choice is made based on confidence exceeding the dependability threshold d_{th}.

Algorithm 1 Class-Conditional Knowledge Integration (CCKI)

Require: Paired dataset (x, y), rare label y_r, thresholds τ_m, τ_g, d_{th}, hypothesis set $\mathcal{H}$ with label partitions S_h

```
Initialize active set Ω ← (x, y)
while active set Ω continues to refine meaningfully do
    for each hypothesis h(x) ∈ H do
        Compute entropy imbalance gain EIG(h(x)) on Ω
    end for
    h*(x) ← arg max_{h(x)} EIG(h(x))
    if no tie within τ_m then
        if y_r ∈ S_{h*} then
            Compute Gini(s)
            if Gini > τ_g then
                Update active set: Ω ← s
            else
                Stop
            end if
        else
            Ω ← {s ∈ S_{h*} | y_r ∈ s}
        end if
    else
        Compute confidences for tied hypotheses and choose one with score > d_th
    end if
end while
```

The algorithm's design is directly motivated by Theorem 1: by selectively cascading knowledge-driven classifiers for tail classes and data-driven models for head classes, CCKI disentangles domain-generalization (smoothness) and long-tail

(sharpness) objectives. EIG is favored over traditional information gain or cross-entropy because it emphasizes uncertainty reduction specifically for imbalanced classes, ensuring that rare-class predictions are prioritized. The Gini index complements this by providing a computationally stable and interpretable measure of partition purity, guiding the decision on whether further cascading is necessary (Gini, 1912; Breiman et al., 1984; Quinlan, 1993; Loh, 2011) to quantify intraclass purity.

B.2 Knowledge Extraction for Enabling CCKI

A central challenge in realizing Class-Conditional Knowledge Integration (CCKI) lies in extracting the relevant expert knowledge required to guide model decisions. Such knowledge extraction often relies on domain experts, medical annotators, or specialized deep learning pipelines. For example, lesion- or structure-specific cues frequently require fine-tuned object-detection models such as YOLO (Redmon et al., 2016), SAM (Kirillov et al., 2023), or Detectron2 (Wu et al., 2019) for lesion localization, or U-Net (Ronneberger et al., 2015) for vessel segmentation, each typically trained on expert-annotated datasets of at least 500 images per knowledge attribute. This dependence on fine-grained, pixel-level, or bounding-box annotations limits the scalability of CCKI to broader medical tasks where such resources are scarce or prohibitively expensive. Therefore, a promising next direction is to develop an agentic framework called ***Agentic Causal Disen- tanglement (CANDICE)***, capable of autonomously constructing these pipelines, retrieving domain-specific cues, and organizing them into structured priors for CCKI, thereby significantly reducing manual human involvement.

B.3 Ablation Study I: Evaluating Symbolic Knowledge for CCKI

Ablation Study I: Evaluating Symbolic Knowledge for Selecting $h_{\mathrm{K}}(x)$. To determine which symbolic knowledge features are suitable for constructing the CCKI knowledge hypothesis $h_{\mathrm{K}}(x)$, we evaluate two clinically motivated feature families on APTOS: (1) lesion biomarkers (exudates, hard hemorrhages, soft hemorrhages, cotton wool spots), and (2) retinal vein morphology (tortuosity, caliber, branching angles). The goal is to test whether these symbolic features provide domain-stable discriminative power consistent with the requirement that $P(K \mid Y)$ remains approximately invariant across imaging centers. We train five standard classifiers on each feature set and report performance in Table 13.

Across all models, lesion-only features yield substantially higher accuracy and F1-score than the combined lesion-plus-vein feature set. Gradient Boosting with lesion biomarkers achieves the best results (accuracy 0.8465, F1 0.8412), indicating that lesion-level symbolic cues form a clean, well-separated representation for DR stages. In contrast, adding vein morphology consistently degrades performance for every classifier, suggesting that these features introduce domain-sensitive variability rather than causal invariants. This behavior aligns with our theoretical framework: only knowledge features with low class-conditional domain divergence are appropriate for $h_{\mathrm{K}}(x)$ in CCKI, while domain-unstable features increase the effective discrepancy term and weaken rare-class guarantees. Based on this ablation, we select **Gradient Boosting on lesion biomarkers only** as the canonical knowledge classifier $h_{\mathrm{K}}(x)$ used in our hypothesis pool. This choice provides stable, interpretable, and clinically grounded decision boundaries that integrate reliably with deep hypotheses $h_{\mathrm{D}}(x)$ in the CCKI rule cascade.

Model	Feature Set	Acc	F1	Prec	Rec	AUC
Logistic Reg.	Lesions only	0.7732	0.7322	0.59	0.49	0.74
Random Forest	Lesions only	0.8169	0.8115	0.82	0.80	0.81
SVM	Lesions only	0.7814	0.7432	0.59	0.50	0.76
Grad. Boost.	Lesions only	**0.8465**	**0.8412**	**0.82**	**0.76**	**0.84**
KNN	Lesions only	0.7814	0.7896	0.63	0.56	0.77
Logistic Reg.	Lesions + vein	0.6424	0.6019	0.25	0.33	0.58
Random Forest	Lesions + vein	0.7384	0.7038	0.55	0.47	0.70
SVM	Lesions + vein	0.6556	0.6083	0.26	0.34	0.58
Grad. Boost.	Lesions + vein	0.7252	0.7389	0.51	0.44	0.69
KNN	Lesions + vein	0.6987	0.6369	0.43	0.44	0.66

Table 13: **Ablation on symbolic lesion biomarkers with and without retinal vein features on APTOS.** Lesion-only features provide the strongest and most stable performance across models; adding vein morphology degrades accuracy and F1.

B.4 Ablation Study II: Effect of Hypothesis Pool Composition in CCKI

Ablation Study II: Effect of Hypothesis Pool Composition in CCKI. We evaluate how the composition of the CCKI hypothesis pool $\mathcal{H}$ influences in-domain performance on APTOS. All experiments use the standard 5-class Diabetic Retinopathy (DR) classification setting (stages 0 4). The hypothesis pool contains two types of models:

- **Knowledge-guided hypotheses** $h_{\mathrm{K}}(x)$ implemented using Gradient Boosting over a fixed 10-dimensional clinical feature vector $\mathcal{K}$ (as per Ablation Study I).

- **Deep-learning hypotheses** $h_{\mathrm{D}}(x)$ implemented as ViT-based image classifiers fine-tuned for DR grading.

Across all settings, the clinical feature vector $\mathcal{K}$ remains unchanged; we vary: (i) the number of $h_{\mathrm{K}}(x)$ and $h_{\mathrm{D}}(x)$ in $\mathcal{H}$, and (ii) the prediction granularity of each hypothesis (5-class vs. binary one-vs-rest). Six configurations are evaluated (Table 14).

Condition	# Deep $h_{\mathrm{D}}(x)$	# KL $h_{\mathrm{K}}(x)$	AUC (%)
A1: 5-class $h_{\mathrm{D}}(x)$ + 5-class $h_{\mathrm{K}}(x)$	1	1	83.24 ± 0.60
A3: binary $h_{\mathrm{D}}(x)$ + binary $h_{\mathrm{K}}(x)$	5	5	81.49 ± 0.30
A2: binary $h_{\mathrm{K}}(x)$ + 5-class $h_{\mathrm{D}}(x)$	1	5	$\mathbf{84.65 \pm 0.30}$
A4: binary $h_{\mathrm{D}}(x)$ + 5-class $h_{\mathrm{K}}(x)$	5	1	78.05 ± 0.76
A5: 5-class $h_{\mathrm{D}}(x)$ only	1	0	78.74 ± 0.98
A6: 5-class $h_{\mathrm{K}}(x)$ only	0	1	80.63 ± 0.13

Table 14: Ablation of CCKI hypothesis pool composition on APTOS (5-class DR classification).

Discussion. Condition A2 delivers the highest accuracy because the 5-class deep hypothesis provides holistic visual representation, while the five binary $h_{\mathrm{K}}(x)$ models contribute class-specific clinical cues that improve fine-grained discrimination. Large binary-only mixtures (A3, A4) perform worse due to overlapping or contradictory decision boundaries within the CCKI rule cascade. Single-family baselines (A5, A6) underperform as they lack complementary perspectives.

Limitations and Future Directions. CCKI supports an unbounded number of hypotheses, but effective operation requires avoiding excessive redundancy when using one-vs-rest specialists. In this work, $|\mathcal{Y}| = 5$ (DR stages), so the binary hypotheses naturally map to five DR subclasses. Importantly, CCKI is **not restricted to binary splits**: it can accommodate arbitrary sub-multiclass hypotheses (e.g., mild vs. severe tiers), which we identify as a promising direction for future work.

C Reinforcement Learning–Based Optimization

Beyond code refinement, we employ a reinforcement learning agent to tune key detection and preprocessing parameters (e.g., YOLO confidence thresholds, segmentation post-processing filters). When annotated validation data are available, the agent observes the current parameter value as the state, selects actions from $\{-0.05, 0, +0.05\}$, and receives the resulting IoU as the reward. Updates follow a standard Q-learning schedule with learning rate $\alpha = 0.1$ and discount factor $\gamma = 0.9$. Additional implementation details are provided in final suplimentary materials.

C.1 Discussion of EchoNet-Dynamic system's Result

CANDICE's results do not surpass the hand-engineered EchoNet-Dynamic system, which benefits from extensive video-specific optimization and years of domain-tailored refinement. Importantly, CANDICE provides **substantial reductions in human intervention workload**. While we did not perform a formal measurement, the improvement in external MAE demonstrates enhanced domain generalization, and beat-level aggregation reduces the required human review effort by an estimated $\sim 70\%$.

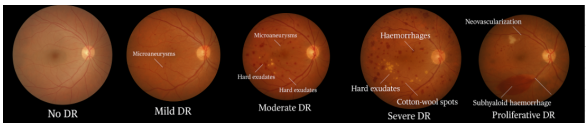

Figure 5: Fundus images showing Diabetic Retinopathy progression: from No DR to Proliferative DR, highlighting key lesions at each stage (Kauppi et al., 2019).

D Appendix Figures

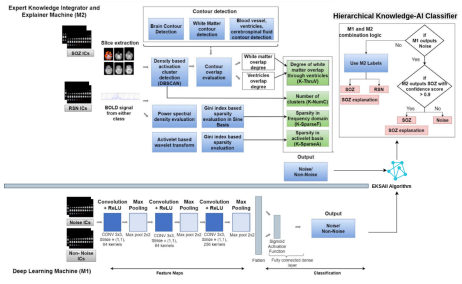

Figure 6: **DeepXSOZ: A Hybrid Knowledge-AI Architecture for Seizure Onset Zone (SOZ) Localization.** The framework employs a bipartite training architecture. During inference, the final SOZ classification is determined by integrating the labels from both M_{DL} and $\mathrm{M}_{\mathrm{CCKI}}$ via confidence scores, yielding a final, integrated, and explainable diagnostic result.

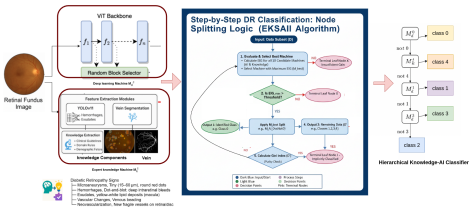

Figure 7: **Hierarchical Knowledge-AI Integration Framework for Diabetic Retinopathy (DR) Classification.** The system integrates a **Deep Learning Machine** (M_d, ViT backbone) and an **Expert Knowledge Machine** (M_k, clinical features/guidelines) within a decision tree. The **CCKI algorithm** iteratively selects the optimal binary classifier (maximum Entropy Imbalance Gain, EIG) for node splitting.

E Analysis and Discussion

In this section, we analyze the behavior of CANDICE beyond aggregate performance metrics and discuss the roles played by individual agents, the generality of the framework, and its relationship to prior reasoning paradigms such as Chain of Thought (CoT). Our goal is to clarify *why* CANDICE works, not merely *that* it works.

E.1 Importance of CGA Agent

The Code Generation Agent (CGA) plays a critical but often underappreciated role in the CANDICE framework. While the CRA and CDA are responsible for reasoning and planning, respectively, the CGA ensures that these abstract decisions are grounded in executable, verifiable computations. Our experiments show that this grounding is essential for both robustness and interpretability.

Without the CGA, reasoning outputs remain symbolic or textual artifacts that may appear coherent but fail silently when applied to real inputs. This failure mode is particularly problematic under domain shift, where assumptions encoded in reasoning chains may not hold. By contrast, the CGA enforces executability: every decision pathway selected by the CDA must correspond to a concrete program whose behavior can be observed and validated.

The tool use evaluation in Table 4 highlights this effect quantitatively. Compared to single shot LLM code generation and Toolformer style agents, the CGA achieves higher final execution rates with fewer correction iterations. More importantly, execution failures become explicit signals that can be used by the CDA to revise plans, rather than latent errors that propagate unnoticed.

From a causal perspective, the CGA acts as a *grounding intervention*. It prevents the system from relying on spurious symbolic reasoning by forcing alignment between abstract knowledge and observable computation. This property is especially valuable in safety critical or high stakes settings, but it is equally important for scientific validity: it enables precise error attribution and systematic debugging of agentic behavior.

E.2 Importance of CDA Agent

The Causal Disentanglement Agent (CDA) is the core decision making component of CANDICE and the primary source of performance gains observed across tasks. Ablation results in Table 3 demonstrate that removing or simplifying the CDA leads to substantial degradation in Tail F1 and success rate, even when all other components are retained.

The key contribution of the CDA is not merely planning, but *selective intervention*. Prior agentic systems often apply reasoning or tool use uniformly across inputs, leading to unnecessary computation and increased error rates. The CDA instead learns to differentiate between inputs that benefit from knowledge intensive reasoning and those that are best handled by domain invariant statistical models.

This selectivity is crucial for resolving the long standing conflict between domain generalization and long tailed learning. Head classes benefit from smooth, invariant decision boundaries, while tail classes require sharp, knowledge guided distinctions. By dynamically routing inputs through different pathways, the CDA prevents these objectives from interfering with one another.

Conceptually, the CDA transforms the learning problem from a single global optimization into a collection of local, context dependent decisions. This perspective aligns with decision theoretic views of intelligence and suggests that robustness under distribution shift may fundamentally require agentic control rather than monolithic predictors.

E.3 Model Agnostic Nature

An important property of CANDICE is its model agnostic nature. The framework does not assume a specific backbone architecture, modality, or training objective. Instead, it operates at the level of decision orchestration, making it compatible with a wide range of base models, including vision encoders, sequence models, and multimodal systems.

This property is empirically supported by the diversity of tasks evaluated in Section 3. Despite

substantial differences in input structure and label semantics, CANDICE consistently improves robustness and tail performance. These gains cannot be attributed to architectural specialization, but rather to the agentic CCKI principle that governs when and how knowledge is integrated.

From a practical standpoint, model agnosticism makes CANDICE easier to deploy and extend. Existing systems can be augmented with agentic causal disentanglement without retraining core models from scratch. From a scientific standpoint, it suggests that the benefits of CANDICE stem from structural properties of decision making rather than domain specific heuristics.

E.4 Constraints based CoT vs. CANDICE

Recent work has proposed constraining Chain of Thought (CoT) reasoning to improve reliability and reduce hallucination. While these approaches share superficial similarities with CANDICE, they differ fundamentally in scope and mechanism.

Constraints based CoT methods operate within a single reasoning trace, enforcing syntactic or semantic validity of intermediate steps. They do not alter the underlying decision structure of the model, nor do they provide a mechanism for resolving conflicts between competing objectives such as robustness and tail sensitivity.

CANDICE, by contrast, treats reasoning as one component of a broader causal decision process. Reasoning outputs are not ends in themselves, but inputs to a planner (CDA) that decides whether, when, and how they should influence predictions. Moreover, CANDICE grounds reasoning through executable programs, something that CoT based methods do not address.

In this sense, CANDICE subsumes constrained CoT as a special case: reasoning can be constrained, but it is never unconditional. This distinction explains why CANDICE achieves consistent gains under domain shift, whereas CoT based methods often fail to generalize beyond their training distributions.

E.5 Clinical Reasoning Agent (CRA)

The **Clinical Reasoning Agent (CRA)** performs structured reasoning over biomedical and clinical knowledge to make evidence grounded decisions. It was evaluated for reasoning faithfulness and knowledge grounding (see Table 1).

CRA Prompt Example

Role

Clinical Decision Maker and Evidence Synthesizer

Objective

Produce a step by step reasoning path that references clinically validated sources, identifies causal relationships, and concludes with a diagnosis or recommended action.

Instructions

1. **Evidence Retrieval:** Retrieve relevant literature, guidelines, or structured knowledge bases before reasoning. Cite sources explicitly where applicable.
2. **Stepwise Reasoning:** Generate reasoning in a multi step chain (symptom $\rightarrow$ mechanism $\rightarrow$ differential diagnosis).
3. **Validation:** For each claim, check if supported by evidence. Mark unsupported claims clearly.
4. **Decision Output:** Provide a final recommendation or classification only after completing reasoning.
5. **Error Handling:** Indicate uncertainty and suggest additional tests if evidence is insufficient.

Prompt Template (Experiments)

```
"You are a clinical reasoning agent.
Patient scenario: {input_case}.
Retrieve relevant guidelines and
↪ literature.
For each observation, generate a
↪ causal reasoning step.
Mark evidence supported steps
↪ clearly.
At the end, provide a diagnosis or
↪ action plan.
If uncertain, suggest further tests or
↪ observations."
```

Notes

CRA achieved high *Faithful Steps* (85.7%) and low hallucination rate (4.5%) in experiments.

E.6 Causal Disentanglement Agent (CDA)

The **Causal Disentanglement Agent (CDA)** is responsible for multi step planning, agent coordination, and disentangling causal factors, especially under domain shift and long tailed distributions.

```
\subsubsection*{Role}
Multi  Agent Planner and Coordinator

\subsubsection*{Objective}
Plan and sequence agent  level actions to
↪  maximize task success and robustness,
↪  performing replanning if intermediate
↪  steps fail.

\subsubsection*{Instructions}
\begin{enumerate}
    \item \textbf{Task Decomposition:} Break
    ↪  complex tasks into sequential sub
    ↪  tasks.
    \item \textbf{Agent Scheduling:} Decide
    ↪  execution order for specialized
    ↪  agents (CRA, CGA, etc.).
    \item \textbf{Replanning:} Trigger
    ↪  replanning upon errors, uncertainty,
    ↪  or unexpected outcomes.
    \item \textbf{Step Tracking:} Maintain
    ↪  executed steps and reasoning states
    ↪  for traceability.
    \item \textbf{Causal Disentanglement:}
    ↪  Identify and separate spurious
    ↪  correlations from causal mechanisms.
\end{enumerate}
```

Prompt Template (Experiments)

```
"You are the Causal Disentanglement Agent.
Given a task T, decompose it into sub  tasks.
Assign sub  tasks to specialized agents.
Monitor their outputs.
If any step fails, replan a new sequence.
Maintain causal trace logs for each action."
```

```
\subsubsection*{Notes}
CDA planning improved performance (accuracy
↪  81.1\%, tail  F1 70.1\%) while reducing
↪  average steps to 2.2 (see
↪  Table~\ref{tab:planning_ablation}).
```

E.7 Code Generation Agent (CGA)

The **Code Generation Agent (CGA)** generates, executes, and corrects programmatic workflows, integrating tools and ensuring reproducible execution (see Table 4).

CGA Prompt Example

Role

Autonomous Code Synthesizer and Executor

Objective

Generate correct executable code, handle errors, and iteratively refine outputs to reach successful execution.

Instructions

1. **Code Generation:** Translate sub task requirements into Python (or target language) code.
2. **Execution Testing:** Run code in a safe environment and log errors.
3. **Correction Loop:** Analyze errors, fix, and retry until successful or maximum retries reached.
4. **Tool Integration:** Use specialized libraries or APIs as required.
5. **Output Reporting:** Provide final outputs along with a brief execution trace.

Prompt Template (Experiments)

```
"You are a code generation agent.
Given a sub  task, write executable
↪  Python code.
Run the code safely and report errors.
Iteratively fix errors up to N retries.
Return final output and execution
↪  log."
```

Notes

CGA achieved 90.1% first run success and 96.3% final execution rate.

F Additional Experimental Results

F.1 Language Only Clinical Reasoning Task

Method	Accuracy (%)	Reasoning F1	Decision Consistency
LLM (Zero Shot)	63.4	58.7	61.2
RAG + LLM	70.5	65.2	68.0
ReAct	72.9	67.8	70.1
CANDICE (Ours)	**78.3**	**74.5**	**77.6**

Table 15: Evaluation of CANDICE on a language only clinical reasoning task.

F.2 Computational Cost and Efficiency

Method	Inference Time (s)	# LLM Calls	Memory (GB)	Accuracy Gain
Vision only Model	1.2	1	4.0	
RAG Baseline	2.5	3	6.2	+6.3
ReAct Agent	3.0	5	6.8	+7.8
CANDICE (Ours)	3.8	7	7.5	+13.4

Table 16: Computational cost and efficiency analysis for different frameworks.

Metric	Definition	Computation / Formula	Who Computes	Auto	Human
Replanning Triggered (%)	Fraction of samples where the agent detects failure or uncertainty and initiates a new plan	$\frac{\#\text{samples with replanning}}{\#\text{total samples}} \times 100$	System logger (CDA)	✓	
Recovery Success (%)	Percentage of replanned cases successfully solved after replanning	$\frac{\#\text{successful recoveries}}{\#\text{replanned cases}} \times 100$	Evaluation script	✓	
Avg. Steps to Recovery	Average number of actions required after failure to reach a valid solution	Mean number of agent/tool calls after first failure	Execution trace analyzer	✓	
Faithful Steps (%)	Proportion of reasoning steps supported by retrieved evidence	$\frac{\#\text{evidence supported steps}}{\#\text{total reasoning steps}} \times 100$	NLI based verifier	✓	△
Unsupported Claims ↓	Avg. number of reasoning claims without evidence	Mean count of unsupported steps per sample	Trace validation script	✓	
Hallucination Rate ↓	Percentage of outputs containing factually incorrect claims	$\frac{\#\text{hallucinated outputs}}{\#\text{total outputs}} \times 100$	Verifier + audit	△	✓
Expert Agreement (%)	Agreement between agent reasoning and expert judgment	$\frac{\#\text{expert approved outputs}}{\#\text{evaluated outputs}} \times 100$	Domain experts		✓
Reasoning Stability	Consistency of reasoning under minor input perturbations	Average similarity (e.g., Jaccard / tree edit distance) across runs	Stability analysis script	✓	
Error Attribution	Distribution of failure sources across reasoning, retrieval, perception, and tools	Categorical classification of failure causes	Mixed (rules + audit)	△	△
First Run Success (%)	Percentage of executions succeeding without correction	$\frac{\#\text{first run successes}}{\#\text{total executions}} \times 100$	Execution logs	✓	
Avg. Fix Iterations ↓	Average number of correction loops after failure	Mean number of fix cycles per failed execution	Execution trace analyzer	✓	
Final Execution Rate (%)	Percentage of tasks succeeding after all corrections	$\frac{\#\text{eventually successful runs}}{\#\text{total runs}} \times 100$	Execution monitor	✓	
Runtime Overhead	Additional runtime introduced by agent reasoning	$T_{\text{agent}} T_{\text{baseline}}$	System profiler	✓	

Auto indicates fully automated computation. △ denotes partial automation with human validation on a subset of samples.

Table 17: Definition and computation protocol for agent centric evaluation metrics used in this work.

Agent	Role in CANDICE	How it is made (inputs → outputs)	How it works (core steps)	Comparator baselines and empirical wins (from your tables)
CRA	Grounded reasoning and causal explanation (no final prediction).	Trusted corpus retrieval → atomic facts → class-conditional reasoning artifacts + evidence links.	(1) Retrieve trusted docs (Lewis et al., 2020) (2) Decompose into atomic factual units (Roberts et al., 2020) (3) Build class-conditional causal chains (Andreas et al., 2016; Tafjord and Clark, 2021) (4) Verify step faithfulness; flag unsupported steps (Ji et al., 2023).	**Baselines:** LLM (No Retrieval), RAG (No Reasoning) (Lewis et al., 2020), ReAct (Yao et al., 2023b). **Wins:** Table 1: Faithful 85.7 vs 73.5 (ReAct) / 68.9 (RAG) / 55.2 (LLM); Hallucination 4.5 vs 9.8 / 12.4 / 18.5; Expert 90.2 vs 79.0 / 74.6 / 61.3. Robust degradation under knowledge ablation (Table 2).
CDA	Constraint-aware planning; selects latent pathway z and orchestrates agents/tools.	CRA artifacts + environment constraints (X, K, D) → pathway choice z (agent order, tool/model selection, hypothesis instantiation).	(1) Assess input availability/extractability (2) Estimate class uncertainty + domain sensitivity (3) Choose class-conditional pathway z to preserve tail fidelity under shift (4) Allocate steps/hypotheses under budget.	**Baselines:** Fixed order, Random order, Greedy (single-step), No CDA (heuristic only). **Wins:** Table 3: Accuracy 81.1 vs 76.2 (No CDA); Tail F1 70.1 vs 64.5; Success 84.7 vs 77.0; Avg. Steps 2.2 vs 3.0 (fewer steps with higher success).
CGA	Executable realization; translates specs into verifiable code and repairs failures.	CRA specs + CDA plan → runnable pipeline code + execution logs + repaired code (if needed).	(1) Generate program from specs (Chen et al., 2021) (2) Execute and validate tool outputs (3) Diagnose failures and revise (self-debug / repair) (Gupta et al., 2017; Chen et al., 2024) (4) Return executable pipeline + trace.	**Baselines:** Single-shot LLM code generation (Chen et al., 2021), Toolformer-style agent (Schick et al., 2023), Human-written code (upper bound). **Wins:** Table 4: First-run 90.1 vs 78.5 (Toolformer) / 68.3 (single-shot); Avg. Fix 1.2 vs 1.9 / 2.7; Final Exec 96.3 vs 91.2 / 84.1.

Table 18: Summary of CANDICE agents: responsibilities, construction, operational steps, and empirical wins relative to comparator baselines (using values from Tables 1, 2, 3, and 4).

Symptom	Key Observations and Diagnostic Relevance
Microaneurysms	Tiny red capillary dilations in the retina; the earliest sign of Mild NPDR. Their progression correlates with disease severity (Frank, 2004; Wilkinson et al., 2003; Singh et al., 2008).
Haemorrhages	Includes dot/blot and flame-shaped types indicating microvascular leakage. Severe NPDR is marked by >20 hemorrhages in all quadrants; risk of PDR rises to $\sim 50\%$ within a year (of Ophthalmology, 2023; Publishing, 2024; Group, 1991; Singh et al., 2008).
Hard Exudates	Lipid-rich deposits from chronic leakage, often in/near the macula. Indicative of risk for Diabetic Macular Edema (DME), a major cause of vision loss (Group, 1991; Publishing, 2024; Shukla and Tripathy, 2025).
Cotton Wool Spots	Fluffy white retinal lesions caused by nerve fiber layer infarctions. Signify retinal ischemia in Moderate to Severe NPDR (Frank, 2004; Publishing, 2024; Shukla and Tripathy, 2025).
Subhyaloid Haemorrhages	Boat- or D-shaped hemorrhages between retina and hyaloid face, typically from ruptured neovascular vessels. Hallmark of Proliferative DR (Yanoff and Duker, 2019; for Pediatric Ophthalmology and Strabismus, 2023; Shukla and Tripathy, 2025).
Neovascularization	Fragile vessel growth on optic disc (NVD) or retina (NVE). Defining trait of PDR. High-risk cases without treatment face $\sim 50\%$ vision loss within 5 years (Group, 1991; of Ophthalmology, 2023; Shukla and Tripathy, 2025).

Table 19: Clinical signs of DR and their diagnostic significance.

Without Document RAG CoT Prompt (clinical only, no class prediction)

Role. Retina specialist describing clinical grading criteria for diabetic retinopathy (DR) on color fundus photography.

Classes (5).

1. No DR
2. Mild NPDR
3. Moderate NPDR
4. Severe NPDR
5. Proliferative DR (PDR)

Important constraints.

- Do NOT assign or predict a final class for this image.
- Output must be clinical findings only (no lay explanations).
- Do not hallucinate lesions. If not clearly visible, label "uncertain" or "not assessable".
- If image quality/field of view prevents assessment of a criterion, explicitly state "not assessable".
- No treatment, prognosis, or medical advice.

Prompt template.

Without Document RAG CoT Prompt (clinical only, no class prediction) (continued)

```
ROLE: You are a retina specialist describing clinical grading criteria for diabetic
↪  retinopathy (DR) on color fundus photography.

CLASSES (5):
1) No DR
2) Mild NPDR
3) Moderate NPDR
4) Severe NPDR
5) Proliferative DR (PDR)

IMPORTANT CONSTRAINTS:
   Do NOT assign or predict a final class for this image.
   Output must be clinical findings only (no lay explanations).
   Do not hallucinate lesions. If not clearly visible, label "uncertain" or "not assessable".
   If image quality/field  of  view prevents assessment of a criterion, explicitly state "not
   ↪  assessable".
   No treatment, prognosis, or medical advice.

INPUT:
   Disease: Diabetic Retinopathy (DR)
   Image: (attached fundus photo)

TASK (think step  by  step internally, but DO NOT reveal private chain  of  thought):
1) Image adequacy: report focus, illumination, artifacts, and whether macula + optic disc +
↪  quadrants are assessable.
2) Extract ONLY observable findings in this image (lesion inventory):
      Microaneurysms (MA)
      Intraretinal hemorrhages (dot/blot/flame), approximate distribution by quadrant
      Hard exudates (and proximity to fovea)
      Cotton  wool spots (CWS)
      Venous beading (VB)
      IRMA
      Neovascularization (NVD/NVE)
      Pre  retinal hemorrhage / vitreous hemorrhage
      Fibrovascular proliferation / tractional signs (if visible)
    A) Required/defining findings for that class
   B) Exclusion findings (what would rule it out or push to a different class)
   C) For THIS image: mark each defining finding as one of:
         Present
         Absent
         Uncertain
         Not assessable
   D) What additional confirmation a doctor would seek if uncertain (e.g., wider field, OCT
   ↪  for DME, FA, repeat photo)

GRADING ANCHORS (use clinically standard cues):
   Mild NPDR: MA only.
   Moderate NPDR: more than MA  only but not severe; may have hemorrhages/exudates/CWS; mild
   ↪  VB/IRMA possible.
   Severe NPDR: 4  2  1 rule (any one):

   PDR: NVD/NVE and/or pre  retinal/vitreous hemorrhage; fibrovascular proliferation.
```

Without Document RAG CoT Prompt (clinical only, no class prediction) (continued)

```
OUTPUT FORMAT (STRICT):
[Image Adequacy]
...

[Lesion Inventory (visible only)]
   MA:
   Hemorrhages:
   Hard exudates:
   CWS:
   VB:
   IRMA:
   NVD/NVE:
   Pre  /vitreous hemorrhage:
   Fibrovascular/tractional cues:

[Per  Class Doctor Checklists (NO final grade)]
(Class 1) No DR
   Defining findings:
   Exclusions:
   This image (present/absent/uncertain/not assessable):
   Additional confirmation if needed:

(Class 2) Mild NPDR
...

(Class 3) Moderate NPDR
...

(Class 4) Severe NPDR
...

(Class 5) PDR
...
```

Without Document RAG ,ToT Prompt (clinical only, branches, no class prediction)

Role. Retina specialist. Provide a per class clinical decision checklist for DR severity using a Tree of Thought structure.

Constraints.

- Do NOT assign/predict a final class.
- Clinical criteria only. No lay language.
- No hallucination: if not clearly visible, mark “uncertain” or “not assessable”.
- No treatment/prognosis.

Prompt template.

Without Document RAG ,ToT Prompt (clinical only, branches, no class prediction) (continued)

```
ROLE: Retina specialist. Provide a per  class clinical decision checklist for DR severity
↪  using a Tree  of  Thought structure.

CLASSES (5):
1) No DR
2) Mild NPDR
3) Moderate NPDR
4) Severe NPDR
5) PDR

CONSTRAINTS:
   Do NOT assign/predict a final class.
   Clinical criteria only. No lay language.
   No hallucination: if not clearly visible, mark "uncertain" or "not assessable".
   No treatment/prognosis.

INPUT:
   Disease: DR
   Image: (attached fundus photo)

TREE  OF  THOUGHT PROCEDURE:
Think in multiple branches internally, then output ONLY the structured branch summaries.

Step 1) Image adequacy: focus, illumination, artifacts, and coverage (macula, disc,
↪  quadrants).

Step 2) Lesion inventory (visible only): MA, hemorrhages (by quadrant), hard exudates (foveal
↪  proximity),
        CWS, VB, IRMA, NVD/NVE, pre  /vitreous hemorrhage, fibrovascular/tractional signs.

Step 3) Build 4 branches that cover all severity criteria, WITHOUT concluding a final class:
   Branch A: "Red lesion burden" (MA + hemorrhages extent; quadrant distribution)
   Branch B: "Ischemia markers" (CWS + VB + IRMA; explicitly map to 4  2  1 components)
   Branch C: "Proliferation screen" (NVD/NVE; pre  retinal/vitreous hemorrhage; fibrovascular
   ↪  cues)
   Branch D: "Quality/confounders" (artifacts, blur, poor field; mimics)

Each branch outputs:
   What findings are assessed (criteria)
   For THIS image: present/absent/uncertain/not assessable
   What additional evidence would be needed for confident assessment

Step 4) Convert branches into a per  class checklist table (text  only):
For each class (1  5):
   Defining criteria (clinical)
   "Image evidence status" for each criterion (present/absent/uncertain/not assessable)
   Exclusion triggers (findings that would push to another class)
   Additional confirmation if needed

OUTPUT FORMAT:
[Image Adequacy]
...
[Lesion Inventory]
...
[Branches]
(Branch A) ...
(Branch B) ...
(Branch C) ...
(Branch D) ...
[Per  Class Clinical Checklists (NO final grade)]
Class 1 ...
Class 2 ...
Class 3 ...
Class 4 ...
Class 5 ...
```

With Document RAG ToT Prompt (primary reference grounded; no class prediction)

Role. Retina specialist. Provide a per class clinical decision checklist for DR severity using a Tree of Thought structure.

Evidence requirement (published reference document).

- Use the following clinically curated published reference as the **PRIMARY** source for per class criteria and severity anchors:
- *Shukla UV, Tripathy K. Diabetic Retinopathy. [Updated 2023 Aug 25]. In: StatPearls [Internet]. Treasure Island (FL): StatPearls Publishing; 2025 Jan . (NBK560805)*
- If any criterion is not explicitly stated in the reference, label it as `(supplemental)` and keep it minimal.

Prompt template.

With Document RAG ToT Prompt (primary reference grounded; no class prediction) (continued)

```
ROLE: Retina specialist. Provide a per  class clinical decision checklist for DR severity
↪  using a Tree  of  Thought structure.

EVIDENCE REQUIREMENT (PUBLISHED REFERENCE DOCUMENT):
   Use the following proven, clinically curated published reference as the PRIMARY source for
   ↪  the per  class criteria and severity anchors:
  Shukla UV, Tripathy K. Diabetic Retinopathy. [Updated 2023 Aug 25]. In: StatPearls
  ↪  [Internet].
  Treasure Island (FL): StatPearls Publishing; 2025 Jan  . (NBK560805)
   When producing each checklist item, prefer definitions explicitly stated in the reference
  (e.g., International Clinical DR Severity Scale / ETDRS  related definitions).
   If you include any criterion not explicitly stated in the reference, label it
   ↪  "(supplemental)" and keep it minimal.

CLASSES (5):
1) No DR
2) Mild NPDR
3) Moderate NPDR
4) Severe NPDR
5) PDR

CONSTRAINTS:
   Do NOT assign/predict a final class.
   Clinical criteria only. No lay language.
   No hallucination: if not clearly visible, mark "uncertain" or "not assessable".
   No treatment/prognosis.

INPUT:
   Disease: DR
   Image: (attached fundus photo)
   Reference: StatPearls NBK560805 (as above; assume it is available to you)

TREE  OF  THOUGHT PROCEDURE:
Think in multiple branches internally, then output ONLY the structured branch summaries.

Step 1) Image adequacy: focus, illumination, artifacts, and coverage (macula, disc,
↪  quadrants).

Step 2) Lesion inventory (visible only): MA, hemorrhages (by quadrant), hard exudates (foveal
↪  proximity),
        CWS, VB, IRMA, NVD/NVE, pre  /vitreous hemorrhage, fibrovascular/tractional signs.

Step 3) Build 4 branches that cover all severity criteria, WITHOUT concluding a final class:
   Branch A: "Red lesion burden" (MA + hemorrhages extent; quadrant distribution)
   Branch B: "Ischemia markers" (CWS + VB + IRMA; explicitly map to 4  2  1 components per
   ↪  reference)
   Branch C: "Proliferation screen" (NVD/NVE; pre  retinal/vitreous hemorrhage; fibrovascular
   ↪  cues)
   Branch D: "Quality/confounders" (artifacts, blur, poor field; mimics)

Each branch outputs:
   What findings are assessed (criteria; aligned to the reference)
   For THIS image: present/absent/uncertain/not assessable
   What additional evidence would be needed for confident assessment

Step 4) Convert branches into a per  class checklist table (text  only):
For each class (1  5):
   Defining criteria (clinical; grounded in the reference)
   "Image evidence status" for each criterion (present/absent/uncertain/not assessable)
   Exclusion triggers (findings that would push to another class)
   Additional confirmation if needed
```

With Document RAG ToT Prompt (primary reference grounded; no class prediction) (continued)

```
OUTPUT FORMAT:
[Lesion Inventory]
...

[Branches]
(Branch A) ...
(Branch B) ...
(Branch C) ...
(Branch D) ...

[Per  Class Doctor Checklists (NO final grade)]
(Class 1) No DR
.....

(Class 2) Mild NPDR
...

(Class 3) Moderate NPDR
...

(Class 4) Severe NPDR
...

(Class 5) PDR
...
```

CDA Prompt (Block 1/3: Role + Parts A C; NO final grade)

Role & objectives.

```
ROLE: You are a Senior Retina Specialist + Causal Machine Learning Engineer. Your
↪  objective is to:
(1) decompose a fundus image into a structured, engineering  ready feature set for a
↪  Diabetic Retinopathy (DR) grading system,
(2) output per  class evidence using Binary Gates + Non  Binary Gradients
↪  (Excluded/Possible/Uncertain),
(3) perform CDA  style causal considerations for confounding and domain generalization,
(4) design a "tree of machines" (hierarchical classifiers) including BOTH knowledge
↪  classifiers (rule  based) and deep learning classifiers,
(5) provide a coding  ready implementation plan (schemas + module pipeline + node
↪  interfaces) so the next agent can write code.

DR CLASSES (5):
1) No DR
2) Mild NPDR (MA  only)
3) Moderate NPDR (more than MA  only; not Severe)
4) Severe NPDR (4  2  1 rule; no NV)
5) PDR (NV and/or pre  /vitreous hemorrhage; fibrovascular/tractional signs)

NON  NEGOTIABLE RULES:
   DO NOT assign or predict a final class for this image.
   DO NOT output "Class X", "final grade", or any single  class decision.
   Every class outcome must be expressed as: EXCLUDED / POSSIBLE / UNCERTAIN.
   All gates/nodes must output: True / False / Unknown (never a final grade).
   Output must be clinical + technical only (no lay language).
   Do not hallucinate lesions. If unclear due to blur/FOV/artifacts, label "Uncertain"
   ↪  or "Not Assessable".
   If the field  of  view prevents assessing a criterion, explicitly state "Not
   ↪  Assessable".
   No treatment/prognosis/advice.
   Follow the internal reasoning path, but DO NOT reveal chain  of  thought. Output only
   ↪  the structured sections below.
   Precision rule: if a lesion is ~70% likely but blurry, mark it "Uncertain".

INTERNAL REASONING PATH (DO INTERNALLY, DO NOT OUTPUT):
1) Visual Evidence Extraction: Scan the image for 5  class anchors; visible vs obscured.
2) Differential Logic: For each class, decide strict exclusions (Binary Gate) vs
↪  insufficient separation (Non  Binary Gradient).
3) System Requirements: What must the system measure to automate this?
4) Tree  of  Machines Planning: Hierarchical binary/multiclass gates using knowledge
↪  rules + feature  ML + deep models.

INPUT:
   Disease: Diabetic Retinopathy (DR)
   Image: (attached fundus photo)
```

CDA Prompt (Block 1/3: Role + Parts A C; NO final grade) (continued)

Structured output spec (Parts A to C).

```
============================================================
PART A: LESION INVENTORY (VISIBLE ONLY)
============================================================
A1) Image Adequacy (STRICT):
   focus: Good/Fair/Poor
   illumination: Good/Fair/Poor
   artifacts: [list]
   peripheral_visibility/FOV: Adequate/Limited
   quadrant_assessability: {Q1: Assessable/NotAssessable, Q2:..., Q3:..., Q4:...}
   disc_visible: Yes/No/Uncertain
   macula_visible: Yes/No/Uncertain

A2) Findings (status belong {Present, Absent, Uncertain, NotAssessable}; include 1 line
↪  note each):
   MA:
   Hemorrhages (dot/blot/flame):
   Hard Exudates (HE) + fovea proximity:
   Cotton  Wool Spots (CWS):
   Venous Beading (VB):
   IRMA:
   Neovascularization (NV: NVD/NVE):
   Pre  retinal / vitreous hemorrhage:
   Fibrovascular / tractional signs:

============================================================
PART B: PER  CLASS IMPORTANT FEATURE SET
============================================================
For each Class 1  5 provide:
B1) Key Distinguishing Features (minimum clinical requirements)
B2) Must  Quantify (presence vs count vs quadrant distribution vs disc/macula/fovea
↪  location)
B3) Upgrade Trigger (single finding that moves into this class or above)
B4) Minimum machine  observable signals required (what detectors/segmenters must output)

============================================================
PART C: PER  CLASS REASONING (BINARY + NON  BINARY)
============================================================
For each Class 1  5 output:

C1) Binary Gates (hard checks; gate_status {True, False, Unknown}):
   Provide 3  6 gates per class.
   Each gate must include:
  {gate_name, gate_definition, required_inputs, gate_status_for_this_image,
  ↪  evidence_from_PartA}

C2) Non  Binary Gradients (graded signals; level {Low, Medium, High} or numeric):
   Provide 3  6 signals per class:
  {signal_name, definition, required_inputs, level_for_this_image, evidence_from_PartA}

C3) Status for this class (REQUIRED; NOT a final grade):
   status_for_this_class: EXCLUDED / POSSIBLE / UNCERTAIN
   explanation: 2  4 bullets referencing C1/C2 and assessability limits
```

CDA Prompt (Block 2/3: Parts D to E; NO final grade)

```
============================================================
PART D: KNOWLEDGE SUFFICIENCY TEST
============================================================
For each Class 1  5:
   sufficiency: Sufficient / PartiallySufficient / Insufficient
   why: 2  4 bullets (must cite missing quadrants/blur/artifacts if relevant)
   missing_information: [exact missing counts/locations/visibility]
   recommended_additional_imaging/tests:
     UWF/Wide  field (purpose)
     FA (purpose: IRMA vs NV; leakage; nonperfusion)
     OCT (purpose: macular edema evaluation if relevant to feature extraction; note not
     ↪  equivalent to DR class)

============================================================
PART E: CAUSAL DISCOVERY DECISION (CDA)
============================================================
E1) Confounders & measurement variables:
   camera_type, site, illumination, blur, FOV, compression, artifact_presence, grader_noise
E2) Causal goal:
   Ensure model learns Lesion to Grade rather than ImageQuality/Camera to Grade
E3) Decision:
   causal_ml: NOT_NEEDED / OPTIONAL / RECOMMENDED
   justification: 3  6 bullets tied to Part A limitations and domain shift risks
E4) Minimal text DAG (nodes + arrows):
   U(systemic severity) to L(lesions) to S(severity)
   Q(measurement: blur/illum/FOV/camera) to Y(pixels)
   L to Y
   Q masks L (missingness/measurement error)
   grader_noise → labels
```

CDA Prompt (Block 3/3: Parts F G; NO final grade)

```
============================================================
PART F: CODING PLAN & LOGIC ENGINE (ENGINEERING  READY)
============================================================
F1) Data Schemas (JSON  like; MUST use exact field names):
QualityReport:
{illumination_score: float, blur_score: float, fov_score: float, artifact_flags: [str],
 quadrant_visibility: {Q1: bool, Q2: bool, Q3: bool, Q4: bool}}

Anatomy:
{disc_center: [x,y]|null, fovea_center: [x,y]|null, disc_visible: bool, macula_visible: bool,
 quadrant_masks: {Q1: ..., Q2: ..., Q3: ..., Q4: ...}}

LesionDetections:
{MA: [{x: float, y: float, conf: float}],
 IRH: [{bbox: [x1,y1,x2,y2], subtype: "dot"|"blot"|"flame"|"unknown", conf: float}],
 HE: [{mask_or_bbox: ..., area_px: float, conf: float}],
 CWS: [{bbox_or_mask: ..., conf: float}],
 VB: [{segment_id: str, score: float, conf: float}],
 IRMA: [{bbox_or_mask: ..., conf: float}],
 NV: [{type: "NVD"|"NVE"|"unknown", bbox_or_mask: ..., conf: float}],
 PR_VH: [{bbox_or_mask: ..., conf: float}]}

DerivedFeatures:
{MA_count_total: int, IRH_count_total: int, HE_area_total: float,
 MA_by_quadrant: {Q1: int, Q2: int, Q3: int, Q4: int, not_assessable: bool},
 IRH_by_quadrant: {Q1: int, Q2: int, Q3: int, Q4: int, not_assessable: bool},
 VB_quadrants: int|"not_assessable", IRMA_quadrants: int|"not_assessable",
 severe_421_flags: {heme_4q: bool|"not_assessable", vb_2q: bool|"not_assessable", irma_1q:
 ↪  bool|"not_assessable"},
 pdr_flags: {nv_present: bool|"not_assessable", pr_vh_present: bool|"not_assessable"}}

EvidenceChecklistPerClass:
{class_id: int,
 binary_gates: [{gate_name: str, status: "True"|"False"|"Unknown", evidence: str}],
 nonbinary_signals: [{signal_name: str, level: "Low"|"Medium"|"High"|float, evidence: str}],
 status: "EXCLUDED"|"POSSIBLE"|"UNCERTAIN",
 notes: [str]}

F2) Module Pipeline (M1  M7; include inputs→outputs→method):
M1 preprocess(image)  >img_norm
M2 quality(img_norm)  >QualityReport
M3 anatomy(img_norm)  >Anatomy
M4 quadrant_map(Anatomy)  >quadrant_masks + quadrant_visibility
M5 lesions(img_norm,Anatomy)  >LesionDetections
M6 aggregate(LesionDetections,QualityReport,Anatomy)  >DerivedFeatures
M7 logic_engine(DerivedFeatures,LesionDetections,QualityReport)
↪  >EvidenceChecklistPerClass[1..5]
(IMPORTANT: M7 outputs only per  class statuses; never a final grade.)

F3) Pseudocode (MUST NOT RETURN A CLASS):
pipeline_infer(image)   > {QualityReport, Anatomy, LesionDetections, DerivedFeatures,
↪  EvidenceChecklistPerClass}
```

CDA Prompt (Block 3/3: Parts F G; NO final grade) (continued)

```
============================================================
PART G: TREE  OF  MACHINES PLAN (HIERARCHICAL CLASSIFIERS)
============================================================
Goal: Build a hierarchy of "machines" where each node can be implemented as:
(1) Knowledge classifier (rule gate from DerivedFeatures) and
(2) Learning classifier (feature  ML and/or deep model),
and outputs only gate decisions (True/False/Unknown), never a final class.

G0) Planning Steps (MUST INCLUDE):
1) Define node targets (binary/ternary/multiclass) aligned with clinical anchors:
      NoDR vs AnyDR
      PDR vs not  PDR
      Severe (4  2  1) vs not  Severe
      Mild (MA  only) vs More  than  mild
      Moderate consistency check (optional)
2) For each node, define:
      required features + assessability prerequisites
      Unknown conditions (when data insufficient)
      knowledge rule version
      learning version (feature  ML + deep)
3) Decide training data needs per node:
      image  level labels sufficient? or lesion  level annotations required?
4) Decide calibration:
      thresholds to output Unknown under poor quality/low confidence
5) Compose node orchestration:
      run nodes in order, aggregate node outputs to EvidenceChecklistPerClass only (no final
      ↪  grade)

G1) Decision Tree Topology Table (REQUIRED): (include the specified columns)
G2) Node interfaces (coding  ready):
node_k(derived_features, quality_report, lesions)  >{decision, confidence, unknown_reason,
↪  evidence_used}
run_tree(image)
↪  >{QualityReport,DerivedFeatures,EvidenceChecklistPerClass,node_outputs,uncertainty_report}
(IMPORTANT: run_tree MUST NOT output a final grade.)

============================================================
OUTPUT FORMAT REQUIREMENTS:
   Use tables for Part E3 and Part G1.
   Use structured lists for schemas and modules.
   Do NOT output a final grade.
============================================================
```