Coherent Hierarchical Multi-Label Learning to Defer for Medical Imaging

Joshua Strong[1] **Emma Sun**[1] **Pramit Saha**[1] **Helen Higham**[2] **J. Alison Noble**[1]

[1]Department of Engineering Science, University of Oxford, Oxford, UK
[2]Nuffield Department of Anaesthesia, Oxford University Hospitals NHS Foundation Trust, Oxford, UK

Abstract

Learning to Defer (L2D) enables a model to predict autonomously or defer to an expert, but prior work largely assumes flat label spaces. In medical imaging, findings are often hierarchically related: independent prediction and deferral decisions can therefore produce an incoherent human–AI handoff even when the model's predictions are compatible with the taxonomy. We formulate hierarchical multi-label L2D as a structured delegation problem and formalise *deferral coherence* under a Selective-Exclusion handoff contract, combining taxonomic satisfiability, workflow admissibility, and deductive closure. For tree taxonomies and additive system risk, we characterise the Bayes-optimal coherent policy and show that even oracle nodewise Bayes decisions can violate deferral coherence. We develop two complementary approaches: exact coherent projection, which guarantees coherent actions through dynamic programming without retraining, and Taxonomic Belief Propagation (TBP), which defines a joint action model with coherent support. Recursive Policy Optimisation (RPO) trains through TBP's hierarchical dependencies. Across four medical-imaging benchmarks using real-reader annotations or controlled experts, binary-relevance L2D exhibits systematic incoherence. Projection eliminates it exactly, while TBP+RPO with fast marginal decoding achieves near-zero incoherence while retaining strong system utility. These findings highlight the need to model the coherence of expert delegation, not only consistency among predicted labels.

Webpage: https://josh-strong.github.io/coherent-l2d-site/.

1 Introduction

Deep learning performs strongly on many medical-imaging tasks, but is typically utilised as an assistive tool rather than an autonomous decision-maker [1–4]. Learning to Defer (L2D) captures this human–AI setting by allowing a model either to predict or to defer to an expert [5]. Existing L2D methods, however, largely assume flat classification label spaces, where prediction and deferral decisions are made per label without hierarchical constraints [6]. In contrast, medical-imaging decisions are often hierarchical and multi-label: broad findings relate taxonomically to more specific subtypes, and multiple findings may co-occur [7–9]. These taxonomic relationships create dependencies not only between predicted labels, but also between decisions about which findings the model predicts and which it delegates to an expert. We therefore study, to our knowledge, the first L2D formulation that requires the joint configuration of predictions and deferrals to be coherent with both the label hierarchy and an explicit expert-handoff contract. This exposes a failure mode not captured by flat L2D or standard hierarchical multi-label classification (HMLC) alone: *a system's predicted labels may be compatible with the taxonomy, yet its pattern of deferrals may constitute an incoherent handoff to the human*.

A natural starting point is to apply Binary Relevance (BR) [10] to hierarchical L2D, treating each finding as an independent decision to predict or defer. However, because findings are related by

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

the taxonomy, these decisions can conflict. For example, a model may predict a specific finding as present while ruling out its broader parent finding; defer a broad finding to the clinical reader while independently predicting one of its subtypes as present; or defer a finding whose absence is already implied by a prediction at its parent. We call such failures *deferral incoherence*. While the first case is a familiar contradiction between hierarchical labels, the latter two reveal a distinct problem arising from expert delegation: the model's predictions may be logically compatible, yet its division of responsibility with the clinical reader can be inconsistent or redundant.

To address these failures, we formalise *deferral coherence* by combining taxonomic constraints with an explicit *Selective-Exclusion* handoff contract. Under this contract, deferring a broad finding delegates that decision to a clinical reader: the model may still rule out specific subtypes, but cannot independently predict them as present. We develop two complementary approaches to addressing incoherence: *exact coherent projection*, which uses dynamic programming to enforce constraints at inference, and *Taxonomic Belief Propagation* (TBP) with *Recursive Policy Optimisation* (RPO), which builds the constraints into a joint action model and trains through its hierarchical recursion. We evaluate both approaches on VinDr-CXR [11] and CheXpert [9] using real-reader annotations, and on PadChest [12] and ADPv2 [13] using controlled experts.

Relation to nearby settings. Flat L2D studies when to defer but not how deferral interacts with a taxonomy [6]. HMLC studies hierarchical label consistency but not expert handoff actions. Hierarchical selective classification allows abstention or backoff in a hierarchy, but typically in single-label settings and without modelling expert-provided labels on deferred nodes [14]. Our formulation brings together three elements: a multi-label hierarchy, ternary prediction/deferral actions, and system utility under expert delegation. An extended related literature review is given in Appendix A.

Contributions. First, we introduce hierarchical multi-label L2D as a structured human–AI handoff problem, in which prediction and deferral decisions must be coherent across a taxonomy. Second, we formalise *deferral incoherence* through taxonomic satisfiability, workflow admissibility, and deductive closure, and characterise the Bayes-optimal coherent deferral rule. This yields a key negative result: even oracle nodewise Bayes L2D can produce delegation violations and deductive defects. Third, we propose two complementary approaches: exact coherent projection, which guarantees zero incoherence at inference, and TBP+RPO, in which TBP defines a joint action model with coherent support and RPO trains through its hierarchical recursion. Across four medical-imaging benchmarks, naïve binary-relevance L2D (BR-L2D) exhibits systematic incoherence, projection eliminates it, and TBP+RPO with fast marginal decoding achieves near-zero incoherence while retaining strong utility.

2 Background

Hierarchical multi-label classification (HMLC) represents relationships between findings as a taxonomy $\mathcal{T} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is the set of nodes representing findings and $\mathcal{E}$ is the set of parent–child relations. Let $y_j \in \{0, 1\}$ indicate the absence or presence of finding j. A positive child requires its parent to be positive, while an absent parent implies absent descendants. We call a binary label assignment *label-consistent* if it respects these implications throughout the taxonomy. HMLC methods enforce or exploit such constraints over binary label values [15, 16].

Whereas HMLC concerns label values, L2D additionally concerns *actions*: for an input x (e.g., a medical image), the system decides whether to predict each finding autonomously or defer to an expert, with performance evaluated at the level of the combined human–AI system [5, 17]. Let $m_j \in \{0, 1\}$ denote the expert's prediction for finding j. As a baseline, we extend L2D to hierarchical multi-label classification using Binary Relevance (BR), treating each node $j \in \mathcal{V}$ as an independent decision with action space $\mathcal{A} = \{0, 1, \perp\}$, where 0 and 1 denote autonomous predictions and $\perp$ denotes deferral. Given a local action distribution $\pi_j(\cdot \mid x)$ and a single-concept L2D surrogate $\mathcal{L}_\phi$, BR minimises

$$\mathcal{J}_{\mathrm{BR}}(x, y, m) = \sum\nolimits_{j \in \mathcal{V}} \mathcal{L}_\phi\big(\pi_j(\cdot \mid x), y_j, m_j\big). \tag{1}$$

This decomposition is computationally simple in that it requires one local loss per node and permits independent action decoding, without joint optimisation over the taxonomy. However, by treating actions independently, it ignores the ancestor–descendant dependencies that can make the resulting predictions and deferrals incoherent.

3 Deferral Coherence

The label-consistency constraints introduced above govern binary predictions, but do not by themselves determine which findings a model may defer to an expert. We therefore define *deferral coherence* through three requirements: (i) the model's autonomous predictions must be compatible with the taxonomy, (ii) prediction and deferral decisions must respect an explicit handoff contract, and (iii) the model must not defer findings whose values are already determined by its own predictions. We formalise these requirements below, with an informal explanation in Appendix B.

3.1 Setup and Label-Space Assumptions

We consider a tree taxonomy $\mathcal{T} = (\mathcal{V}, \mathcal{E})$, with parent map $pa(\cdot)$ and children $C(p)$ for each node p. Each finding $v \in \mathcal{V}$ has a binary label $y_v \in \{0, 1\}$ indicating its absence or presence. Figure 1 illustrates an example schematic subtree used throughout this section.

As introduced in Section 2, label consistency requires **upward implication**: for every edge $(p \to c) \in \mathcal{E}$,

$$y_c = 1 \implies y_p = 1. \tag{2}$$

Equivalently, its contrapositive gives **downward implication**: $y_p = 0 \implies y_c = 0$.

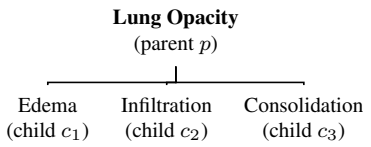


Figure 1: Illustrative subtree used throughout Section 3.

We additionally adopt an **open-world assumption**, under which the listed children need not exhaust all possible subtypes of a parent finding:

$$y_p = 1 \not\Rightarrow \exists c \in C(p) \text{ such that } y_c = 1. \tag{3}$$

Thus, a parent may be present even when all its modelled children are absent.

3.2 Taxonomic Satisfiability

We now extend the notion of label consistency to L2D actions. For each finding $v \in \mathcal{V}$, the model selects an action $a_v \in \mathcal{A} := \{0, 1, \perp\}$, where 0 and 1 are autonomous predictions and $\perp$ delegates the decision to the expert. Importantly, deferral is not a third binary label value: it leaves the finding's value unspecified by the model.

We therefore ask whether the model's autonomous predictions could all hold simultaneously under the taxonomy. To do so, we consider the binary label assignments that remain possible after respecting every non-deferred prediction.

Definition 3.1 (Compatibility set and taxonomic satisfiability)**.** Let $y = (y_v)_{v \in \mathcal{V}}$ be a binary label vector, and let

$$\mathcal{Y}_{\mathcal{T}} := \left\{ y \in \{0,1\}^{|\mathcal{V}|} : \forall (p \to c) \in \mathcal{E},\ y_c = 1 \Rightarrow y_p = 1 \right\}$$

denote the set of label-consistent assignments under upward implication (Eq. (2)). For an action vector $a \in \mathcal{A}^{|\mathcal{V}|}$, define

$$\mathsf{Comp}(a) := \{ y \in \mathcal{Y}_{\mathcal{T}} : \forall v \in \mathcal{V},\ a_v \neq \perp \Rightarrow y_v = a_v \}.$$

This is the *compatibility set*: the set of complete binary label assignments compatible with all autonomous model predictions. Deferred nodes impose no additional label restrictions. An action vector a is *taxonomically satisfiable* if $\mathsf{Comp}(a) \neq \emptyset$, meaning that at least one such assignment exists.

Definition 3.2. (Taxonomic contradiction) An action vector a is a *taxonomic contradiction* if $\mathsf{Comp}(a) = \emptyset$.

For example, predicting *Lung Opacity* absent while predicting *Edema* present is a taxonomic contradiction: no label-consistent assignment can satisfy both predictions. In contrast, deferring *Lung Opacity* while predicting *Edema* present is taxonomically satisfiable, since the parent could be present. Whether this constitutes an admissible handoff is a separate question addressed next.

For a parent and its immediate children considered in isolation, satisfiability reduces to a simple local test:

Proposition 1 (Local satisfiability criterion)**.** *Let p be a parent node with children $C(p)$. The local configuration $(a_p, \{a_c\}_{c \in C(p)})$ is taxonomically satisfiable if and only if it does not predict any child present while predicting the parent absent:*

$$\neg\Big(a_p = 0 \ \wedge \ \exists c \in C(p) : a_c = 1\Big).$$

3.3 Workflow Admissibility: Deferring a Parent

Taxonomic satisfiability ensures that the model's autonomous predictions are logically compatible, but does not determine whether its decisions constitute a coherent handoff to the expert. In particular, we must specify what the model is permitted to predict when it defers a broader finding. We adopt a handoff contract motivated by assistive imaging workflows and refined with input from a clinician coauthor.

Definition 3.3 (Selective-Exclusion deferral contract)**.** Under *Selective-Exclusion* (SE), if the model defers a parent finding, it may independently rule out specific subtypes but cannot predict any of them as present. Formally, for every edge $(p \to c)$,

$$a_p = \perp \Rightarrow a_c \in \{0, \perp\}.$$

Together with taxonomic satisfiability, this requires any child predicted present to have a parent also predicted present.

This contract reflects the distinction between ruling out a subtype and committing to the presence of a broader finding. For example, deferring *Lung Opacity* while predicting *Consolidation* present is taxonomically satisfiable, but violates SE: the positive child prediction already implies that the deferred parent is present. In contrast, predicting *Consolidation* absent does not determine whether *Lung Opacity* is present, so such exclusions remain permitted under the open-world assumption (Eq. (3)).

Definition 3.4. (Delegation / contract violation) A local configuration $(a_p, \{a_c\}_{c \in C(p)})$ is a *delegation violation* under SE if

$$a_p = \perp \ \wedge \ \exists c \in C(p) : a_c = 1.$$

Remark 3.5 (Alternative semantics: strong subtree handoff). SE is not the only possible handoff contract. A stricter alternative enforces *strong subtree handoff*: if a parent is deferred, all its descendants must also be deferred. This may suit workflows requiring full transfer of responsibility, but disallows selective exclusion. We formalise this alternative in Appendix C and evaluate it in Appendix C.1.

3.4 Deductive Closure and Deductive Defects

Even when predictions are taxonomically satisfiable and respect the handoff contract, deferral may still be unnecessary if the model has already determined a finding's value through its own predictions. For example, predicting *Lung Opacity* absent necessarily implies that *Infiltration* is absent. Deferring *Infiltration* would therefore delegate a decision whose outcome is already fixed by the model's parent prediction. We formalise this through *taxonomic entailment*.

Definition 3.6 (Taxonomic entailment from actions)**.** For a taxonomically satisfiable action vector a and node $v \in \mathcal{V}$, write

$$a \models_{\mathcal{T}} (v = b) \quad \Longleftrightarrow \quad \forall y \in \mathsf{Comp}(a), \ y_v = b.$$

Thus, the label value $(v = b)$ is *entailed* if every taxonomy-consistent completion compatible with the model's autonomous predictions assigns value b to node v.

Definition 3.7 (Deductive closure)**.** A taxonomically satisfiable action vector a is *deductively closed* if every entailed label value is explicitly predicted by the model:

$$a \models_{\mathcal{T}} (v = b) \Rightarrow a_v = b, \qquad \forall v \in \mathcal{V}, \ b \in \{0, 1\}.$$

Definition 3.8. (Deductive defect) An action vector a has a *deductive defect* if it is taxonomically satisfiable and contract-admissible, but is not deductively closed.

Under the hierarchy's downward implication, predicting a parent absent already determines the absence of all its descendants. This gives a simple case of deductive defect:

Table 1: Complete local classification of parent–children handoff decisions under Selective-Exclusion. Each status describes the parent and its immediate children, with violations classified in the order: taxonomic satisfiability, workflow admissibility, and deductive closure.

Child pattern $\mathbf{a}_C$	Status	Interpretation
Parent absent ($a_p = 0$)		
$\exists c : a_c = 1$	**Taxonomic contradiction**	Positive child under asserted negative parent
$\forall c : a_c \neq 1,\ \exists c : a_c = \bot$	**Deductive defect**	Deferred child already entailed absent
$\forall c : a_c = 0$	**Coherent**	All immediate children correctly excluded
Parent present ($a_p = 1$)		
$\exists c : a_c = 1$	**Coherent**	Consistent subtype assertion under a positive parent
$\forall c : a_c = 0$	**Coherent**	Open-world case: unmodelled subtypes may explain the parent
$\forall c : a_c \neq 1,\ \exists c : a_c = \bot$	**Coherent**	Selective delegation at the subtype level
Parent deferred ($a_p = \bot$)		
$\exists c : a_c = 1$	**Delegation violation**	Positive child commits deferred parent present
$\forall c : a_c \in \{0, \bot\}$	**Coherent**	Parent deferred while children are only excluded or deferred

Corollary 3.9 (Downward implication yields deductive defects)**.** *Suppose a is taxonomically satisfiable and contract-admissible, and $a_p = 0$ for some parent p. Then, for every child $c \in C(p)$,*

$$a \models_{\mathcal{T}} (c = 0).$$

Consequently, if any such child has $a_c = \bot$, the action vector has a deductive defect.

3.5 Local Characterisation of Deferral Coherence

We now combine the three requirements introduced above into a single notion of *deferral coherence*. Although these requirements concern the full configuration of predictions and deferrals across the taxonomy, we show that, under SE, coherence can be characterised entirely through local parent–child relationships.

Definition 3.10 (Deferral coherence)**.** An action vector $a \in \mathcal{A}^{|\mathcal{V}|}$ is *deferral coherent* under SE if it is (i) taxonomically satisfiable, (ii) contract-admissible, and (iii) deductively closed.

Definition 3.11 (Deferral incoherence)**.** An action vector is *deferral-incoherent* under SE if it fails to satisfy at least one of these three requirements.

The following result shows that global deferral coherence can be verified by checking only three forbidden parent–child action patterns.

Proposition 2 (Local characterisation of deferral coherence)**.** *Assume a tree taxonomy and the open-world assumption in Eq. (3). An action vector $a \in \mathcal{A}^{|\mathcal{V}|}$ is deferral coherent under SE if and only if every edge $(p \to c) \in \mathcal{E}$ avoids:*

$$a_p = 0 \ \wedge \ a_c = 1 \quad \textit{(taxonomic contradiction),}$$
$$a_p = \bot \ \wedge \ a_c = 1 \quad \textit{(delegation violation),}$$
$$a_p = 0 \ \wedge \ a_c = \bot \quad \textit{(deductive defect).}$$

Intuitively, if a parent is predicted absent, all its children must also be predicted absent. If a parent is deferred, its children may only be predicted absent or deferred. A parent predicted present permits all three child actions under the open-world assumption. Enforcing these rules on every edge is therefore sufficient to guarantee coherence across the entire taxonomy. Table 1 summarises the resulting classification of local parent–children configurations.

3.6 Bayes-Optimal Coherent Deferral

Having characterised which joint action configurations are coherent, we now ask how to select the one that minimises system risk. In standard flat L2D, the Bayes-optimal decision can be made independently for each label: defer when the expert's conditional correctness exceeds the classifier's Bayes confidence [5, Equation 6]. In hierarchical L2D, the same risk-based principle applies, but the actions must also satisfy the coherence constraints established above.

Let Y_v and M_v denote the ground-truth and expert labels at node v, respectively. With optional node weights $w_v \geq 0$ and deferral costs $\lambda_v \geq 0$, define the conditional risk of each action as

$$\rho_v(a \mid x) = \begin{cases} w_v P(Y_v \neq a \mid x), & a \in \{0, 1\}, \\ w_v \big[P(M_v \neq Y_v \mid x) + \lambda_v\big], & a = \perp. \end{cases}$$

Thus, an autonomous prediction incurs its expected classification error, while deferral incurs the expert's expected error plus a deferral cost. We consider additive system risk across nodes; this does not require independence between their labels or expert errors.

Without coherence constraints, the Bayes-optimal action at each node minimises its own conditional risk:

$$a_v^{\mathrm{BR},B}(x) \in \arg\min_{a \in \mathcal{A}} \rho_v(a \mid x).$$

For $w_v > 0$, this rule defers when

$$P(M_v = Y_v \mid x) \geq \max\{P(Y_v = 0 \mid x), P(Y_v = 1 \mid x)\} + \lambda_v.$$

When $\lambda_v = 0$, this recovers the standard flat-L2D condition: defer if the expert is more likely to be correct than the best autonomous prediction.

Under SE, however, the action selected at a parent restricts the actions available to its children. Let $\mathcal{C}_{\mathrm{SE}}$ denote the set of coherent joint action vectors from Definition 3.10. The admissible child actions given parent action i are $\Gamma_{\mathrm{SE}}(0) = \{0\}$, $\Gamma_{\mathrm{SE}}(1) = \mathcal{A}$, $\Gamma_{\mathrm{SE}}(\perp) = \{0, \perp\}$. These constraints turn the independent nodewise decisions into a structured optimisation problem.

Key result: Bayes-optimal coherent deferral. The Bayes-optimal coherent action vector minimises total conditional risk over the SE-coherent action set:

$$a_{\mathrm{SE}}^{B}(x) \in \arg\min_{a \in \mathcal{C}_{\mathrm{SE}}} \sum_{v \in \mathcal{V}} \rho_v(a_v \mid x). \tag{4}$$

For a tree taxonomy, this can be computed by the subtree recursion

$$B_v(i \mid x) = \min_{j \in \Gamma_{\mathrm{SE}}(i)} \left[\rho_v(j \mid x) + \sum_{u \in C(v)} B_u(j \mid x) \right],$$

where $B_v(i \mid x)$ is the minimum achievable risk over the subtree rooted at v, given parent action i. At the root, the minimisation is over all $j \in \mathcal{A}$.

The recursion captures the central difference from nodewise L2D: choosing an action at an internal node determines which actions remain feasible throughout its subtree. The optimal choice therefore balances the node's own conditional risk against the minimum achievable risk of its descendants. This same tree structure will later support coherent decoding under learned model scores.

Importantly, incoherence is not merely a consequence of imperfect training or poorly calibrated models. Even if every node uses its true conditional label probabilities and expert correctness, independent Bayes-optimal decisions can still violate the handoff requirements.

Proposition 3 (Nodewise Bayes L2D can be deferral-incoherent)**.** *Consider an edge $(p \to c)$ with $Y_c = 1 \Rightarrow Y_p = 1$, zero deferral costs $\lambda_p = \lambda_c = 0$, positive weights $w_p, w_c > 0$, and no ties in the local action scores. Let*

$$p_v(x) = P(Y_v = 1 \mid x), \qquad q_v(x) = P(M_v = Y_v \mid x),$$

and define the nodewise Bayes scores by

$$s_v(0 \mid x) = 1 - p_v(x), \quad s_v(1 \mid x) = p_v(x), \quad s_v(\perp \mid x) = q_v(x).$$

The independent nodewise Bayes rule

$$a_v^{\mathrm{BR},B}(x) \in \arg\max_{a \in \mathcal{A}} s_v(a \mid x)$$

cannot produce $(a_p, a_c) = (0, 1)$, but there exist distributions for which it produces $(\perp, 1)$ or $(0, \perp)$. Hence, even oracle-calibrated BR-L2D can be deferral-incoherent under SE. The proof is given in Appendix D.

4 Contract-Aware Decoding and Training

We develop two complementary approaches to coherent hierarchical L2D: an inference-only decoder that projects predictions onto the SE-coherent action set, and a joint action model that incorporates the handoff constraints by construction.

Setup and local primitives. Each node $v \in \mathcal{V}$ takes an action $a_v \in \mathcal{A} = \{0, 1, \perp\}$. With node-specific logits $f_\theta^{(v)}(x) \in \mathbb{R}^{|\mathcal{A}|}$, define

$$\boldsymbol{\eta}_v(a \mid x) := \mathrm{Softmax}(f_\theta^{(v)}(x))_a, \qquad a \in \mathcal{A}. \tag{5}$$

These primitives represent local preferences over prediction and deferral actions, before feasibility constraints are imposed. By themselves, they do not define a coherent joint policy.

4.1 Exact Coherent Projection

BR+Projection is an inference-only structured baseline. Training is unchanged: we first learn local BR-L2D primitives $\boldsymbol{\eta}_v(\cdot \mid x)$ independently at each node. At test time, we replace independent nodewise decoding with exact MAP decoding over the SE-coherent action set $\mathcal{C}_{\mathrm{SE}}$ from Definition 3.10:

$$\hat{\mathbf{a}}^{\mathrm{Proj}}(x) \in \arg\max_{\mathbf{a} \in \mathcal{C}_{\mathrm{SE}}} \sum\nolimits_{v \in \mathcal{V}} \log \boldsymbol{\eta}_v(a_v \mid x). \tag{6}$$

This uses the same tree-structured dynamic programming approach as the Bayes recursion in Section 3.6, with learned log primitive scores replacing conditional risks. Projection is therefore exact MAP under the trained local action model, but is not necessarily Bayes-risk optimal. A single coherent MAP decode costs $O(|\mathcal{V}||\mathcal{A}|^2)$; computing the budgeted deferral ranking requires additional constrained decodes.

For budgeted evaluation, we rank deferrals under the same coherent model. For node t and action $a^\star$, define the constrained MAP value

$$V_t(a^\star \mid x) = \max\nolimits_{\mathbf{a} \in \mathcal{C}_{\mathrm{SE}}:\, a_t = a^\star} \sum\nolimits_{v \in \mathcal{V}} \log \boldsymbol{\eta}_v(a_v \mid x).$$

We rank node-level deferrals by

$$s_{\mathrm{Proj}}(x, t) = V_t(\perp \mid x) - \max\{V_t(0 \mid x), V_t(1 \mid x)\}.$$

At each budget, we select deferrals using this ranking and apply a feasibility closure where necessary. The tree DP then finds the exact coherent MAP action vector conditional on the resulting defer mask. Thus, exactness concerns decoding for a fixed mask, rather than jointly optimising over all possible defer allocations. Full details are given in Appendix H.4.

4.2 Taxonomic Belief Propagation

Our second approach builds the handoff contract into the model. Taxonomic Belief Propagation (TBP) maps local primitives $\{\boldsymbol{\eta}_v\}$ to unconditional action marginals $\{\boldsymbol{\mu}_v\}$ through a tree-structured joint action model. Unlike ordinary label propagation, its transitions operate on prediction/deferral actions: the parent action determines which child actions are admissible.

For each non-root node v, define

$$\mathbf{T}_v(x) \in [0,1]^{3\times 3}, \qquad (\mathbf{T}_v)_{ij} = P(a_v = j \mid a_{pa(v)} = i, x),$$

with rows and columns ordered as $[0, 1, \perp]$. Under SE,

$$a_{pa(v)} = 0 \Rightarrow a_v = 0, \qquad a_{pa(v)} = 1 \Rightarrow a_v \sim \boldsymbol{\eta}_v(\cdot \mid x), \qquad a_{pa(v)} = \perp \Rightarrow a_v \in \{0, \perp\}.$$

Thus,

$$\mathbf{T}_v(x) = \begin{bmatrix} 1 & 0 & 0 \\ \boldsymbol{\eta}_v(0 \mid x) & \boldsymbol{\eta}_v(1 \mid x) & \boldsymbol{\eta}_v(\perp \mid x) \\ \alpha_v(x) & 0 & 1 - \alpha_v(x) \end{bmatrix}, \qquad \alpha_v(x) = \frac{\boldsymbol{\eta}_v(0 \mid x)}{\boldsymbol{\eta}_v(0 \mid x) + \boldsymbol{\eta}_v(\perp \mid x)}. \tag{7}$$

The deferred-parent row renormalises the local primitive over the admissible actions $\{0, \perp\}$. Appendix E.7 shows that this is the unique reverse-KL projection onto the admissible probability simplex face.

Let $\mathcal{R}$ denote the set of taxonomy roots. For each root $r \in \mathcal{R}$, $\boldsymbol{\mu}_r(x) = \boldsymbol{\eta}_r(\cdot \mid x)$; for non-root nodes,

$$\boldsymbol{\mu}_v(x) = \boldsymbol{\mu}_{pa(v)}(x)^\top \mathbf{T}_v(x). \tag{8}$$

Equivalently, TBP induces the joint action model

$$P_{\mathrm{TBP}}(\mathbf{a} \mid x) = \prod_{r \in \mathcal{R}} \boldsymbol{\eta}_r(a_r \mid x) \prod_{v \notin \mathcal{R}} \mathbf{T}_v(x)_{a_{pa(v)}, a_v}. \tag{9}$$

The structural zeros in (7) assign forbidden parent–child action pairs zero probability. Consequently, the joint model has coherent support by construction, without requiring post-hoc repair. Appendix E gives the normalisation, constraint-enforcement, monotonicity, and reverse-KL proofs. Exact TBP MAP decoding is described in Appendix I.

4.3 Contract-Induced Non-Separability Motivates RPO

Projection shows that coherence can be imposed after training, but does not remove the mismatch between independent BR training and structured inference. BR learns local primitives as if each node could be decoded independently, whereas coherent decisions couple ancestor and descendant actions. Under SE, predicting a parent absent forces its descendants absent, while deferring a parent prevents positive assertions below it. Thus, an ancestor decision should account not only for its local risk but also for the downstream options it makes available.

Proposition 4 (Contract-induced non-separability)**.** *Under SE, there exist parent–child defer problems for which the locally Bayes-optimal parent action differs from the parent action induced by the Bayes-optimal coherent policy.*

Proof sketch. Consider an input x for which deferring the parent has slightly lower local risk than predicting it present. However, predicting the parent present enables a much lower-risk child-present action that is forbidden under parent deferral. The coherent policy therefore predicts the parent present, despite deferral being locally optimal. Appendix F gives a constructive example.

4.4 Recursive Policy Optimisation

The preceding result motivates training through the hierarchical dependencies used by TBP. Rather than optimising only local primitives, Recursive Policy Optimisation (RPO) optimises losses on the TBP-composed marginals $\{\boldsymbol{\mu}_v(x)\}$, allowing descendant losses to update ancestor primitives.

We train in two stages. Stage I learns local primitives with the BR objective; Stage II activates TBP and fine-tunes through the composed marginals:

$$\mathcal{J}_{\mathrm{StageI}} = \sum\nolimits_{v \in \mathcal{V}} \mathcal{L}_\phi\big(\boldsymbol{\eta}_v(x), y_v, m_v\big), \qquad \mathcal{J}_{\mathrm{RPO}} = \sum\nolimits_{v \in \mathcal{V}} \mathcal{L}_\phi\big(\boldsymbol{\mu}_v(x), y_v, m_v\big). \tag{10}$$

Stage II allows the model to account for the downstream consequences of deferring, asserting, or excluding a broader finding. Importantly, the objective remains a sum of per-node surrogate losses on TBP-composed marginals, rather than the full structured risk of a particular budgeted decoder.

Joint model versus decoding. The TBP joint model has coherent support by construction. However, independently decoding its node marginals can select an action combination that has zero probability under the joint model. Consequently, the fast marginal decoder used in the main experiments is efficient but not guaranteed coherent. Exact TBP MAP decoding guarantees coherence at additional computational cost (Appendix I).

5 Experiments

We evaluate three questions: whether independent nodewise L2D produces deferral incoherence, whether exact projection eliminates it, and whether TBP+RPO improves the utility–coherence trade-off compared with continued BR training.

Datasets and experts. VinDr-CXR [11] and CheXpert [9] provide the primary real-reader evidence through retrospective radiologist annotations. For each expert task, one reader supplies the expert labels, while the remaining readers form the reference consensus; reference scores are thresholded

Table 2: **Primary results across four medical-imaging benchmarks.** Entries are areas under budget-swept curves. Higher utility and lower incoherence are better. Incoherence is measured over parent–children neighbourhoods. Best means are bolded; second-best are underlined.

Method	BalAcc ↑	F1-I ↑	F1-L ↑	Tax. ↓	Ded. ↓	Del. ↓	Any ↓
– **VinDr-CXR** [11] (*real-reader*, large tax., pooled 15 runs) –							
BR-L2D	.852	.512	.749	≈ 0	.088	.019	.107
BR (cont.)	.847	<u>.514</u>	<u>.763</u>	≈ 0	.115	.024	.138
BR+Proj.	**.853**	.513	.749	**= 0**	**= 0**	**= 0**	**= 0**
TBP+RPO	**.853**	**.515**	**.768**	**= 0**	≈ 0	≈ 0	<u>.004</u>
– **CheXpert** [9] (*real-reader*, small tax., pooled 20 runs) –							
BR-L2D	.811	.641	.734	**= 0**	.011	.038	.050
BR (cont.)	<u>.823</u>	<u>.648</u>	<u>.749</u>	**= 0**	.014	.036	.050
BR+Proj.	.812	.642	.735	**= 0**	**= 0**	**= 0**	**= 0**
TBP+RPO	**.831**	**.652**	**.764**	**= 0**	<u>≈ 0</u>	<u>≈ 0</u>	<u>≈ 0</u>
– **PadChest** [12] (*controlled-expert*, xlarge tax.) –							
BR-L2D	.679	.230	.421	<u>≈ 0</u>	.122	.002	.124
BR (cont.)	.671	.218	.410	.000	.129	.002	.131
BR+Proj.	<u>.692</u>	<u>.240</u>	<u>.430</u>	**= 0**	**= 0**	**= 0**	**= 0**
TBP+RPO	**.721**	**.272**	**.437**	**= 0**	<u>≈ 0</u>	<u>≈ 0</u>	<u>≈ 0</u>
– **ADPv2** [13] (*controlled-expert*, medium tax.) –							
BR-L2D	.896	.863	.878	≈ 0	.037	.032	.068
BR (cont.)	**.900**	**.868**	**.882**	<u>≈ 0</u>	.030	.032	.062
BR+Proj.	.895	.861	.876	**= 0**	**= 0**	**= 0**	**= 0**
TBP+RPO	**.900**	**.868**	**.882**	**= 0**	<u>≈ 0</u>	<u>.008</u>	<u>.009</u>

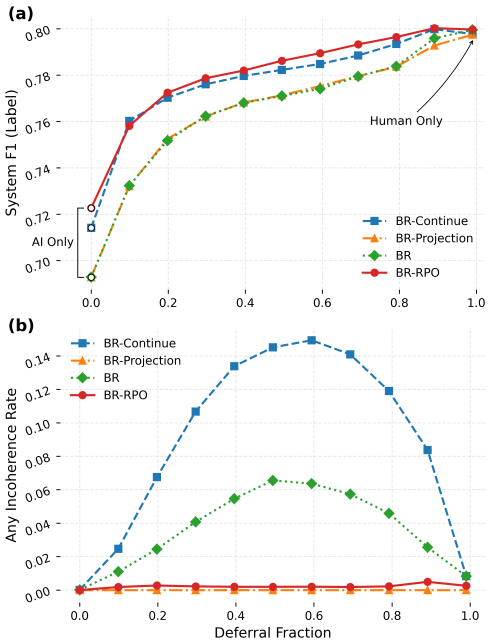


Figure 2: **VinDr-CXR deferral curves.** Mean over 5 seeds deferring to radiologist R9. **(a)** System F1 across node-level deferral budgets. **(b)** Any incoherence over the same sweep. Projection is identically coherent; RPO stays near-coherent while preserving utility.

at 0.5 for correctness and evaluation. We pool five seeds across three VinDr-CXR and four CheXpert expert tasks, giving 15 and 20 runs, respectively. PadChest [12] provides a controlled-expert taxonomy-scale stress test with 61 retained nodes, while ADPv2 [13] tests a medium-size, non-CXR taxonomy. Dataset construction, hierarchy preprocessing, and soft-target handling are detailed in Appendix H.

Methods and evaluation. We compare BR-L2D, matched continued training (BR cont.), BR+Projection, and TBP+RPO with fast marginal decoding. BR continuation matches RPO's Stage-II epoch count and learning-rate schedule, controlling for additional optimisation. For each method, we sweep a global node-level deferral budget across the evaluation set and evaluate system predictions after replacing deferred decisions with expert labels. All entries in Table 2 are areas under budget-swept curves. Balanced accuracy pools study–label decisions; F1-I averages one F1 score per study, while F1-L is pooled micro-F1 across study–label decisions. Incoherence is measured before expert completion over each parent and its immediate children. Each neighbourhood receives the first violated status in Table 1; Any combines the three defect classes. We integrate these frequencies over the same budget sweep. Pairwise statistical tests and further evaluation details are in Appendix H.

Main results. BR-L2D exhibits incoherence across all four datasets, with local failures dominated by deductive defects and delegation violations rather than direct parent-absent/child-present contradictions. Continued training does not consistently resolve these failures: on VinDr-CXR, Any incoherence increases from .107 to .138. Projection eliminates incoherence without retraining, while TBP+RPO reduces it to near zero with strong system utility. On PadChest, TBP+RPO reduces Any incoherence from .124 to approximately zero while improving BalAcc AUC from .679 to .721. On ADPv2, it matches continued BR's utility at the reported precision while reducing Any incoherence from .062 to .009. Thus, coherence gains need not require a loss of utility, although utility improvements are dataset-dependent.

Deferral curves and deployment trade-offs. Figure 2 shows that BR incoherence peaks at intermediate budgets, where autonomous predictions and expert delegation coexist. Projection remains exactly coherent throughout the sweep, while TBP+RPO maintains near-zero incoherence with strong utility. Projection provides coherent decoding for an existing BR model without retraining; fast

Table 3: **Ablation summary.** Each ablation probes an alternative explanation or deployment concern.

Ablation	Question	Takeaway
BR continuation	Is RPO just longer training?	***No.*** Continued BR training does not consistently reduce incoherence and can increase it; RPO trains through deployed TBP recursion. App. L.2, H.6.
Stage-I initialisation	Does RPO depend on BR pretraining?	RPO remains beneficial across tested initialisations, including BR and hierarchy-aware losses. App. L.2.
Hierarchy-aware losses	Would standard HMLC objectives solve this?	Label-consistency losses address binary labels, not delegation violations or deductive defects in ternary action spaces. App. L.2.
Strong-subtree handoff	Is Selective-Exclusion the only valid contract?	***No.*** Stricter subtree handoff is feasible and exactly decodable, but shifts the utility–coherence trade-off. App. C, C.1.
Multi-expert deferral	Does coherence matter with several experts?	The same action-space coherence issue persists with expert-indexed deferral actions. App. L.1.
Exact TBP MAP	Is residual RPO incoherence from TBP itself?	***No.*** Residual incoherence comes from the fast marginal decoder. Exact MAP guarantees zero incoherence, with a dataset-dependent utility trade-off. App. I.
Runtime	Are coherent methods practical?	A single tree decode scales linearly in taxonomy size; exact budgeted ranking is costlier, motivating fast RPO decoding. App. J.

TBP+RPO offers lower inference cost. Exact TBP MAP decoding also guarantees coherence, with additional computational cost and a dataset-dependent utility trade-off (Appendices I and J).

Ablations. Table 3 summarises comparisons with continued training and alternative Stage-I objectives, together with tests of stronger handoff semantics, multi-expert routing, exact decoding, and runtime.

6 Conclusion, Limitations, and Future Work

We introduced hierarchical multi-label L2D as a structured human–AI handoff problem and formalised *deferral coherence* through taxonomic satisfiability, workflow admissibility, and deductive closure. Under SE, we characterised the Bayes-optimal coherent policy for additive system risk and showed that even oracle nodewise Bayes L2D can produce deferral-specific incoherence. Across four medical-imaging benchmarks, these failures arise systematically under independent L2D. Exact projection guarantees coherent actions without retraining, while TBP+RPO enables fast, near-coherent inference with strong utility. These findings highlight the importance of modelling delegation explicitly, rather than enforcing consistency only over predicted labels.

Limitations and future work. Our main formulation assumes tree-structured taxonomies and the SE contract. Although we evaluate a stricter handoff contract, the appropriate delegation semantics remain workflow-dependent. Our evaluations use retrospective annotations or controlled experts, rather than prospective clinical interactions. In particular, retrospective expert annotations are not a controlled reader study: they do not establish how readers would interpret a delegated case, respond to the model's prediction, or revise their decisions under the proposed handoff semantics. Expert performance may therefore differ under deployment conditions, where the information presented, case selection, and time constraints can affect decisions. Our results do not establish benefits for reader comprehension, reading time, or clinical workflow. Reference and expert labels undergo hierarchy preprocessing. Handling frequent reader–taxonomy disagreement remains future work. Further priorities include controlled reader studies and prospective validation of the handoff semantics, alongside inference for more general taxonomies (Appendix K).

Acknowledgements

This work was supported by the Engineering and Physical Sciences Research Council. JS is funded by the EPSRC Center for Doctoral Training in Health Data Science [EP/S02428X/1]. ES is funded by an EPSRC Doctoral Training Partnership [EP/W524311/1]. AN acknowledges EPSRC Turing AI Fellowship: Ultra Sound Multi-Modal Video-based Human-Machine Collaboration [EP/X040186/1]. JS gratefully acknowledges financial support from Kellogg College, University of Oxford, towards attendance at NeurIPS 2026.

References

[1] Bibb Allen, Sheela Agarwal, Laura Coombs, Christoph Wald, and Keith Dreyer. 2020 acr data science institute artificial intelligence survey. *Journal of the American College of Radiology*, 18

(8):1153–1159, 2021.

[2] Christoph D. Becker, Elmar Kotter, Laure Fournier, and Luis Martí-Bonmatí. Current practical experience with artificial intelligence in clinical radiology: a survey of the european society of radiology. *Insights into Imaging*, 13(1), June 2022. ISSN 1869-4101. doi: 10.1186/s13244-022-01247-y. URL http://dx.doi.org/10.1186/s13244-022-01247-y.

[3] Fatma A Eltawil, Michael Atalla, Emily Boulos, Afsaneh Amirabadi, and Pascal N Tyrrell. Analyzing barriers and enablers for the acceptance of artificial intelligence innovations into radiology practice: a scoping review. *Tomography*, 9(4):1443–1455, 2023.

[4] Joshua Strong, Harry Rogers, Emma Sun, Anna Louise Todsen, Jody Ede, Cherry Lumley, Nick Yeung, Helen Higham, and J Alison Noble. Human-AI collaboration in healthcare: A scoping review. *NPJ Digit. Med.*, June 2026.

[5] Hussein Mozannar and David Sontag. Consistent estimators for learning to defer to an expert. In *International Conference on Machine Learning*, pages 7076–7087. PMLR, 2020.

[6] Joshua Strong, Emma Sun, Harry Rogers, Helen Higham, and Alison Noble. Learning to defer: A survey. December 2025.

[7] Curtis P Langlotz. RadLex: a new method for indexing online educational materials. *Radiographics*, 26(6):1595–1597, November 2006.

[8] L Bos and K Donnelly. Snomed-ct: The advanced terminology and coding system for ehealth. *Stud Health Technol Inform*, 121:279–290, 2006.

[9] Jeremy Irvin, Pranav Rajpurkar, Michael Ko, Yifan Yu, Silviana Ciurea-Ilcus, Chris Chute, Henrik Marklund, Behzad Haghgoo, Robyn Ball, Katie Shpanskaya, et al. Chexpert: A large chest radiograph dataset with uncertainty labels and expert comparison. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 590–597, 2019.

[10] Min-Ling Zhang, Yu-Kun Li, Xu-Ying Liu, and Xin Geng. Binary relevance for multi-label learning: an overview. *Frontiers of Computer Science*, 12(2):191–202, 2018.

[11] Ha Quy Nguyen, Hieu Huy Pham, le tuan linh, Minh Dao, and lam khanh. VinDr-CXR: An open dataset of chest X-rays with radiologist annotations. *PhysioNet*, June 2021. doi: 10.13026/3akn-b287. URL https://doi.org/10.13026/3akn-b287. Version 1.0.0.

[12] Aurelia Bustos, Antonio Pertusa, Jose-Maria Salinas, and Maria de la Iglesia-Vayá. Padchest: A large chest x-ray image dataset with multi-label annotated reports. *Medical Image Analysis*, 66:101797, 2020. ISSN 1361-8415. doi: https://doi.org/10.1016/j.media.2020.101797. URL https://www.sciencedirect.com/science/article/pii/S1361841520301614.

[13] Zhiyuan Yang, Kai Li, Sophia Ghamoshi Ramandi, Patricia Brassard, Hakim Khellaf, Vincent Quoc-Huy Trinh, Jennifer Zhang, Lina Chen, Corwyn Rowsell, Sonal Varma, et al. Adpv2: A hierarchical histological tissue type-annotated dataset for potential biomarker discovery of colorectal disease. *Journal of Pathology Informatics*, page 100537, 2025.

[14] Shani Goren, Ido Galil, and Ran El-Yaniv. Hierarchical selective classification. *Advances in Neural Information Processing Systems*, 37:111047–111073, 2024.

[15] Haomin Chen, Shun Miao, Daguang Xu, Gregory D Hager, and Adam P Harrison. Deep hierarchical multi-label classification of chest x-ray images. In *International conference on medical imaging with deep learning*, pages 109–120. PMLR, 2019.

[16] Eleonora Giunchiglia and Thomas Lukasiewicz. Coherent hierarchical multi-label classification networks. *Advances in neural information processing systems*, 33:9662–9673, 2020.

[17] David Madras, Toni Pitassi, and Richard Zemel. Predict responsibly: improving fairness and accuracy by learning to defer. *Advances in neural information processing systems*, 31, 2018.

[18] C. K. Chow. An optimum character recognition system using decision functions. *IRE Transactions on Electronic Computers*, EC-6(4):247–254, 1957. doi: 10.1109/TEC.1957.5222035.

[19] C. Chow. On optimum recognition error and reject tradeoff. *IEEE Transactions on Information Theory*, 16(1):41–46, 1970. doi: 10.1109/TIT.1970.1054406.

[20] Ran El-Yaniv et al. On the foundations of noise-free selective classification. *Journal of Machine Learning Research*, 11(5), 2010.

[21] Yonatan Geifman and Ran El-Yaniv. Selective classification for deep neural networks. *Advances in neural information processing systems*, 30, 2017.

[22] Yonatan Geifman and Ran El-Yaniv. Selectivenet: A deep neural network with an integrated reject option. In *International conference on machine learning*, pages 2151–2159. PMLR, 2019.

[23] Harikrishna Narasimhan, Wittawat Jitkrittum, Aditya K Menon, Ankit Rawat, and Sanjiv Kumar. Post-hoc estimators for learning to defer to an expert. *Advances in Neural Information Processing Systems*, 35:29292–29304, 2022.

[24] Hussein Mozannar, Hunter Lang, Dennis Wei, Prasanna Sattigeri, Subhro Das, and David Sontag. Who should predict? exact algorithms for learning to defer to humans. In *International conference on artificial intelligence and statistics*, pages 10520–10545. PMLR, 2023.

[25] Rajeev Verma, Daniel Barrejón, and Eric Nalisnick. Learning to defer to multiple experts: Consistent surrogate losses, confidence calibration, and conformal ensembles. In *International Conference on Artificial Intelligence and Statistics*, pages 11415–11434. PMLR, 2023.

[26] Andrew Ponomarev. A simple heuristic for controlling human workload in learning to defer. In *International Conference on Pattern Recognition*, pages 120–130. Springer, 2024.

[27] Jean Vieira Alves, Diogo Leitão, Sérgio Jesus, Marco O. P. Sampaio, Javier Liébana, Pedro Saleiro, Mario A. T. Figueiredo, and Pedro Bizarro. Cost-sensitive learning to defer to multiple experts with workload constraints. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=TAvGZm2Rqb.

[28] Anqi Mao, Christopher Mohri, Mehryar Mohri, and Yutao Zhong. Two-stage learning to defer with multiple experts. *Advances in neural information processing systems*, 36:3578–3606, 2023.

[29] Yannis Montreuil, Shu Heng Yeo, Axel Carlier, Lai Xing Ng, and Wei Tsang Ooi. Two-stage learning-to-defer for multi-task learning. *arXiv preprint arXiv:2410.15729*, 2024.

[30] Cuong C Nguyen, Thanh-Toan Do, and Gustavo Carneiro. Probabilistic learning to defer: Handling missing expert annotations and controlling workload distribution. In *The Thirteenth International Conference on Learning Representations*, 2025.

[31] Yannis Montreuil, Axel Carlier, Lai Xing Ng, and Wei Tsang Ooi. Why ask one when you can ask k? learning-to-defer to the top-k experts, 2025. URL https://arxiv.org/abs/2504.12988.

[32] Haoran Lai, Qingsong Yao, Zihang Jiang, Rongsheng Wang, Zhiyang He, Xiaodong Tao, and S Kevin Zhou. Carzero: Cross-attention alignment for radiology zero-shot classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11137–11146, 2024.

[33] Xiaoman Zhang, Chaoyi Wu, Ya Zhang, Weidi Xie, and Yanfeng Wang. Knowledge-enhanced visual-language pre-training on chest radiology images. *Nature Communications*, 14(1):4542, 2023.

[34] Ekin Tiu, Ellie Talius, Pujan Patel, Curtis P Langlotz, Andrew Y Ng, and Pranav Rajpurkar. Expert-level detection of pathologies from unannotated chest x-ray images via self-supervised learning. *Nature biomedical engineering*, 6(12):1399–1406, 2022.

Contents of Appendix

A Extended Literature Review **15**

B An Informal Walk-Through of Deferral Coherence **15**

C Alternative Deferral Semantics: Strong Subtree Handoff **17**

C.1 Contract Sensitivity Analysis 18

D Proof of Proposition 3 **19**

E Validity and Structural Properties of Taxonomic Belief Propagation **19**

E.1 Setup and notation 19

E.2 TBP transition kernel and recursion (Selective-Exclusion) 20

E.3 Property (i): Normalisation 20

E.4 Property (ii): Nonnegativity and bounds 20

E.5 Property (iii): Contract and coherence constraints are enforced 20

E.6 Monotonicity-style implications 21

E.7 Reverse-KL characterisation of the deferred-parent row 21

E.8 Summary 22

F Why end-to-end TBP fine-tuning can improve system utility **22**

F.1 A contract-constrained system risk 23

F.2 Stage I vs. Stage II objectives: a loss-agnostic view 23

F.3 Myopic (separable) vs. contract-aware optimality 23

F.4 A fixed-x counterexample with strict utility gain 24

F.5 Interpretation and link to RPO 24

G Dataset details **25**

G.1 Dataset Licenses and Access Terms 25

G.2 VinDr-CXR 25

G.3 CheXpert 25

G.4 PadChest 25

G.5 ADPv2 25

H Experimental Details **25**

H.1 Dataset Construction and Taxonomy Summaries 25

H.2 Hierarchy Consistency of Reference and Expert Labels 28

H.3 Model Architecture and Training Hyperparameters 29

H.4 Exact Coherent Projection and Dynamic Programs 31

H.5 Budget-Swept Utility and Coherence Metrics 33

H.6 Statistical Testing 35

H.7 Additional CheXpert Statistics 36

H.8 Reproducibility Details 36

I Exact Decoding under the TBP Joint Model **37**

J Inference Runtime **37**

K Extension Beyond Trees: DAG Taxonomies **38**

K.1 DAG label space and action coherence 38

K.2 Multiple-parent feasibility 39

K.3 Coherent projection on a DAG 39

K.4 TBP-style models on a DAG 40

K.5 RPO for DAG taxonomies 40

K.6 Practical interpretation 41

L Additional Experiments and Ablations **41**

L.1 Extension to Multiple Experts 41

L.2 Sensitivity to the Stage I objective 43

A Extended Literature Review

Learning to Defer and selective prediction. Learning to Defer (L2D) formalises human–AI collaboration as an *allocation* decision: the model either predicts or defers to an expert to optimise system-level utility rather than standalone accuracy [5]. Strong et al. [6] provide a recent survey of the area. L2D is closely related to selective classification / learning with a reject option [18–22], but differs in that deferral routes to an external decision-maker with its own error profile rather than simply abstaining. Beyond the original consistent-surrogate formulation [5], later work studies alternative optimisation and deployment regimes, including post-hoc estimators that learn deferral policies on top of a trained predictor [23] and exact / constrained formulations that optimise deferral subject to explicit capacity constraints [24].

Extensions of L2D. Several extensions generalise L2D, addressing routing among multiple experts [25], coverage/workload constraints [26, 27], structured two-training pipelines (assuming a frozen predictor) [28, 29], and missing expert annotations [30]. Most closely adjacent to our training curriculum, [29] develop two-stage L2D formulations for *multi-task* settings (learning a deferral mechanism on top of a multi-head predictor) and further study querying *top-k* experts and related guarantees [31]. These works primarily treat tasks as multiple outputs that share representation and routing constraints. Our focus is different: the outputs themselves are coupled by a *taxonomy*, and coherence is defined by logical satisfiability, an explicit parent-deferral contract, and deductive closure.

Hierarchical multi-label prediction. Hierarchical multi-label classification (HMLC) exploits taxonomies via conditional factorisation and/or constraint-based objectives to improve calibration and enforce hierarchy-consistent predictions [15, 16]. This literature typically constrains *label values* (present/absent) but does not model the *delegation action* introduced by L2D. As a result, directly extending L2D with Binary Relevance can produce workflow-incoherent configurations (e.g., deferring a parent while asserting a child) even when the underlying label predictions are hierarchy-consistent.

Hierarchy-aware abstention and coherence. Recent work has begun to study hierarchy-aware abstention in structured label spaces. Goren et al. [14] formulate hierarchical selective classification, where a classifier may back off to an ancestor node instead of fully rejecting an uncertain example. Their setting is related in motivation, but differs from ours because it is single-label, does not involve expert deferral, and does not address coherence of joint ternary actions in a multi-label taxonomy. Our contribution sits at this intersection. We define *deferral coherence* as the conjunction of (i) taxonomic satisfiability of assertions, (ii) workflow admissibility under an explicit contract (Selective-Exclusion), and (iii) deductive closure. We then characterise the Bayes-optimal coherent deferral rule over the constrained action set and show that independent nodewise Bayes L2D can still be action-incoherent. Algorithmically, we enforce these semantics using contract-specialised tree MAP decoding and a contract-indexed top-down Markov recursion optimised end-to-end with Recursive Policy Optimisation. Our method complements prior hierarchy-aware prediction losses [15, 16] by extending the structured semantics from binary labels to the ternary prediction/deferral action space used in L2D.

Societal impacts. This work could support clearer human–AI collaboration in medical imaging by making the allocation of prediction and deferral decisions explicit and reducing inconsistent or logically redundant handoffs. However, coherence does not guarantee diagnostic correctness or clinical safety: an erroneous parent-level exclusion can force descendant findings to be predicted absent rather than deferred. Unsuitable taxonomies or handoff contracts could also restrict expert involvement in ways that conflict with clinical practice. Our retrospective evaluations do not establish patient or workflow benefits. Before clinical deployment, prospective validation should assess diagnostic errors, clinician workload, and differences in performance and deferral rates across patient groups, while preserving clinicians' ability to review and override model decisions.

B An Informal Walk-Through of Deferral Coherence

This appendix is an informal companion to Section 3. The main text gives the formal definitions and propositions. Here we describe the same ideas in plain language. It can be read either before or after

Section 3. A useful picture to keep in mind is the simple subtree in Figure 1: a parent concept such as *Lung Opacity*, with children such as *Edema*, *Infiltration*, and *Consolidation*.

At each node, the model chooses an *action*: predict absence (0), predict presence (1), or defer to the expert ($\perp$). The key point is that $\perp$ is *not* a third medical label. It is a workflow choice meaning that the model is not taking responsibility for that node and is instead handing the decision to the expert.

Coherence lives in the action space, not just the label space. In ordinary hierarchical multi-label classification, the main concern is logical consistency of labels: a child should not be present if its parent is absent. Once deferral is introduced, that is no longer sufficient. We must also ask whether the *handoff* itself is well-formed. A configuration can be logically possible as a set of labels and still correspond to an inconsistent deferral decision.

A simple three-question checklist. For any parent and its children, coherence can be understood by asking:

1. **Could the model's assertions all be true together?** If not, we have a taxonomic contradiction.
2. **Is the model deferring a parent while still making a child-present claim that already commits the parent?** If yes, we have a delegation violation.
3. **Is the model deferring a node whose value is already implied by its own earlier assertions?** If yes, we have a deductive defect.

Taxonomic contradiction: saying something impossible. The familiar bad case is *parent absent, child present*. For example, *Lung Opacity* = *absent* and *Edema* = *present* cannot both be true. Another way to say this is that, after leaving deferred nodes undecided, there should still remain at least one hierarchy-consistent way the world could look. If no such completion exists, the action vector is taxonomically impossible.

Delegation violation: deferring a question while also answering it indirectly. Under our Selective-Exclusion contract, deferring a parent still allows the model to *exclude* specific children, but it does *not* allow the model to assert any child as present. The reason is operational rather than purely logical: saying *Edema* = *present* already commits *Lung Opacity* = *present*. So the model cannot honestly say "the expert should decide the parent" while simultaneously making a positive subtype assertion underneath it. Equivalently, Selective-Exclusion permits statements like "I am not deciding Lung Opacity overall, but I can already exclude Edema," while forbidding statements like "I defer Lung Opacity, but I assert Edema is present."

Deductive defect: deferring something the model has already settled itself. Suppose the model says *Lung Opacity* = *absent*. Then every child under that node must also be absent. In that case, a choice such as *Edema* = *defer* is not contradictory, but it is still defective: the model is handing off a question whose answer is already fixed by its own parent assertion. Equivalently, deductive closure says that when all hierarchy-consistent completions compatible with the model's current assertions agree on a node, the model should state that value explicitly rather than leave it deferred.

Why "parent present, all children absent" can still be coherent. Section 3 adopts an *open-world* view of the taxonomy: the listed children need not exhaust all possible ways that a parent can be present. So a configuration such as *Lung Opacity* = *present* with all listed children marked absent is still allowed. Intuitively, the parent may be present for a reason that is real but not captured by the modelled child set. This is why the theory only forbids specific bad patterns. It does not require every positive parent to be "explained" by one positive child.

Why this makes the problem structured rather than nodewise independent. The action chosen at a parent changes what is allowed below it. If the parent is marked absent, the subtree is forced absent. If the parent is deferred, children can no longer be asserted present under Selective-Exclusion. So the best action at a node cannot always be chosen independently of its descendants. This is the intuition behind the non-separability result in Section 4.3: a locally sensible defer decision at an internal node can remove valuable options lower in the tree.

Why even nodewise Bayes decisions can fail. In flat L2D, the Bayes rule at one label compares the model's confidence with the expert's probability of being correct. In a hierarchy, doing this independently at every node can still produce an incoherent handoff. For example, the parent may

Table 4: Informal reading of the local parent–children patterns under Selective-Exclusion.

Parent action	**Children**	**Interpretation**
Absent	At least one child present	Impossible : the model asserts a positive child under a negative parent.
Absent	All children absent	Coherent : the model fully closes that part of the tree.
Absent	No child present, but at least one child deferred	Incomplete reasoning : those deferred children are already forced absent by the parent decision.
Present	Any mix of absent / present / defer	Coherent : once the parent is present, the model may exclude, assert, or defer children.
Deferred	At least one child present	Bad handoff : the model says the expert should decide the parent, but a positive child assertion already decides it.
Deferred	Every child absent or deferred	Coherent : the model defers the parent while still being allowed to selectively exclude or defer children.

locally prefer deferral because the expert is strong at the parent, while the child may locally prefer an autonomous positive assertion because the model is strong at the child. This gives a configuration such as *Lung Opacity* = *defer* and *Edema* = *present*, which is a delegation violation. Thus, incoherence is not only a training error. It can arise from solving the wrong, nodewise Bayes problem.

Take-home message. Deferral coherence is best thought of as three common-sense rules acting together:

1. do not assert an impossible parent–child combination;
2. do not defer a question while answering it indirectly through a descendant;
3. do not defer a label that your own assertions have already determined.

Section 3 makes these rules precise, and Section 4 then builds estimators whose predictions respect this coherent decision space.

C Alternative Deferral Semantics: Strong Subtree Handoff

Our main method adopts the *Selective-Exclusion* contract (Definition 3.3), which allows selectively excluding subtypes under parent deferral. For completeness, we also consider a stricter workflow semantics in which deferral at a parent triggers a *subtree handoff*: if a parent is deferred then all descendants must be deferred.

Modified transition kernel. Let $\boldsymbol{\eta}_v(a \mid x) = \mathbb{P}_\theta(a_v = a \mid a_{pa(v)} = 1, x)$ denote the local primitive as in the main text. Under strong subtree handoff, the deferral row becomes deterministic:

$$\mathbf{T}_v^{(A)}(x) = \begin{bmatrix} 1 & 0 & 0 \\ \boldsymbol{\eta}_v(0) & \boldsymbol{\eta}_v(1) & \boldsymbol{\eta}_v(\perp) \\ 0 & 0 & 1 \end{bmatrix}, \tag{11}$$

which enforces $a_{pa(v)} = \perp \Rightarrow a_v = \perp$ for every edge.

Resulting recursion. With $\boldsymbol{\mu}_v = \boldsymbol{\mu}_{pa(v)}^\top \mathbf{T}_v^{(A)}$, the marginals become:

$$\boldsymbol{\mu}_v(1) = \boldsymbol{\mu}_{pa(v)}(1)\, \boldsymbol{\eta}_v(1), \tag{12}$$
$$\boldsymbol{\mu}_v(0) = \boldsymbol{\mu}_{pa(v)}(0) + \boldsymbol{\mu}_{pa(v)}(1)\, \boldsymbol{\eta}_v(0), \tag{13}$$
$$\boldsymbol{\mu}_v(\perp) = \boldsymbol{\mu}_{pa(v)}(\perp) + \boldsymbol{\mu}_{pa(v)}(1)\, \boldsymbol{\eta}_v(\perp). \tag{14}$$

This semantics guarantees monotone deferral down the hierarchy, at the cost of disallowing selective exclusion.

The Bayes-optimal coherent decoder under SSH has the same form as Eq. (4), but with the SSH feasible set. Equivalently, the tree dynamic program uses

$$\Gamma_{\mathrm{SSH}}(0) = \{0\}, \qquad \Gamma_{\mathrm{SSH}}(1) = \mathcal{A}, \qquad \Gamma_{\mathrm{SSH}}(\perp) = \{\perp\}.$$

Thus changing the handoff contract changes the admissible subtree actions, but not the conditional-risk principle.

C.1 Contract Sensitivity Analysis

We evaluate the strong subtree handoff (SSH) contract as a stricter alternative to Selective-Exclusion, where deferring an internal node forces all descendants to be deferred. Tables 5 and 6 show that the same coherent-decoding machinery applies under SSH: projection again gives zero incoherence, while RPO remains near-coherent and typically preserves strong utility.

Table 5: **Balanced utility on VinDr-CXR and CheXpert under the strong subtree handoff (SSH) contract.** Values are mean ± standard deviation for the four utility metrics over the SSH BR-family runs: 15 runs for VinDr-CXR pooled across R8/R9/R10 with BalAcc stopping, and 20 runs for CheXpert pooled across bc1/bc2/bc3/bc5 with F1-I stopping. The final AU-Neigh-Any column reports pooled means only. Higher is better for the first four columns and lower is better for AU-Neigh-Any. Best means are bolded and second-best means are underlined, using the unrounded pooled means to break near-ties. In each dataset block, the last two rows are the two proposed coherent methods: BR+Projection and TBP+RPO, with TBP+RPO evaluated under SSH.

Method	BalAcc ↑	AU-F1-Lmacro ↑	F1-I ↑	F1-L ↑	AU-Neigh-Any ↓
VinDr-CXR [11]					
BR-L2D	0.8525 ± 0.0127	0.4695 ± 0.0180	0.5121 ± 0.0260	0.7494 ± 0.0156	0.107348
BR (cont.)	0.8473 ± 0.0118	$\underline{0.4780 \pm 0.0191}$	$\underline{0.5135 \pm 0.0268}$	$\underline{0.7627 \pm 0.0154}$	0.138449
BR+Projection	$\underline{0.8530 \pm 0.0125}$	0.4706 ± 0.0180	0.5129 ± 0.0258	0.7493 ± 0.0156	**0.000000**
TBP+RPO	$\mathbf{0.8553 \pm 0.0157}$	$\mathbf{0.5190 \pm 0.0297}$	$\mathbf{0.5214 \pm 0.0269}$	$\mathbf{0.7791 \pm 0.0168}$	$\underline{0.000477}$
CheXpert [9]					
BR-L2D	0.8108 ± 0.0084	0.4936 ± 0.0325	0.6469 ± 0.0320	0.7361 ± 0.0140	0.078582
BR (cont.)	$\underline{0.8224 \pm 0.0175}$	$\underline{0.5109 \pm 0.0370}$	$\underline{0.6552 \pm 0.0347}$	$\underline{0.7518 \pm 0.0269}$	0.072095
BR+Projection	0.8104 ± 0.0076	0.4943 ± 0.0332	0.6483 ± 0.0322	0.7363 ± 0.0136	**0.000000**
TBP+RPO	$\mathbf{0.8373 \pm 0.0095}$	$\mathbf{0.5475 \pm 0.0315}$	$\mathbf{0.6657 \pm 0.0328}$	$\mathbf{0.7740 \pm 0.0138}$	$\underline{0.000647}$

Table 6: **Budget-swept decomposition of deferral incoherence on VinDr-CXR and CheXpert under the strong subtree handoff (SSH) contract.** Values are pooled AUC means over the same SSH BR-family runs as Table 5: 15 VinDr-CXR runs (BalAcc-stopped) and 20 CheXpert runs (F1-I-stopped). The left block is edge-weighted (EW), i.e. a micro-average over $(\text{example}, \text{edge})$ pairs; the right block is the coarser neighbourhood-partition (NP) view over $(\text{example}, \text{parent-neighbourhood})$ pairs. Lower is better. In each dataset block, the last two rows are the two proposed coherent methods: BR+Projection and TBP+RPO, with TBP+RPO evaluated under SSH.

Method	Edge-Weighted				Neighbourhood-Partition			
	Tax. ↓	Ded. ↓	Del. ↓	Any ↓	Tax. ↓	Ded. ↓	Del. ↓	Any ↓
VinDr-CXR [11]								
BR-L2D	0.000010	0.043205	0.010970	0.054185	0.000030	0.088128	0.019190	0.107348
BR (cont.)	0.000007	0.061516	0.014627	0.076150	0.000020	0.114878	0.023551	0.138449
BR+Projection	**0.000000**	**0.000000**	**0.000000**	**0.000000**	**0.000000**	**0.000000**	**0.000000**	**0.000000**
TBP+RPO	**0.000000**	0.000183	**0.000000**	0.000183	**0.000000**	0.000477	**0.000000**	0.000477
CheXpert [9]								
BR-L2D	**0.000000**	0.011519	0.034789	0.046308	**0.000000**	0.021758	0.056823	0.078582
BR (cont.)	0.000009	0.014994	0.031695	0.046698	0.000017	0.025054	0.047024	0.072095
BR+Projection	**0.000000**	**0.000000**	**0.000000**	**0.000000**	**0.000000**	**0.000000**	**0.000000**	**0.000000**
TBP+RPO	**0.000000**	0.000343	**0.000000**	0.000343	**0.000000**	0.000647	**0.000000**	0.000647

D Proof of Proposition 3

Proof. For any edge $(p \to c)$, upward implication gives

$$Y_c = 1 \Rightarrow Y_p = 1, \qquad \text{hence} \qquad \pi_c(x) := P(Y_c = 1 \mid x) \le P(Y_p = 1 \mid x) =: \pi_p(x).$$

Assume no ties. If the nodewise Bayes rule predicts $a_c = 1$, then

$$\pi_c(x) > 1 - \pi_c(x),$$

so $\pi_c(x) > 1/2$. By $\pi_c(x) \le \pi_p(x)$, we have $\pi_p(x) > 1/2$, so the parent absent action cannot be locally Bayes-optimal. Hence the nodewise Bayes rule cannot produce $a_p = 0,\ a_c = 1$.

The two deferral-specific failures can occur by construction. For a delegation violation, take

$$\pi_p = .70, \qquad \pi_c = .60, \qquad q_p = .80, \qquad q_c = .40,$$

which is compatible with upward implication, e.g.

$$P(Y_p = 1, Y_c = 1) = .60, \quad P(Y_p = 1, Y_c = 0) = .10, \quad P(Y_p = 0, Y_c = 0) = .30.$$

Then the parent Bayes scores are $(1 - \pi_p, \pi_p, q_p) = (.30, .70, .80)$, so $a_p = \perp$, while the child scores are $(.40, .60, .40)$, so $a_c = 1$. Thus $(a_p, a_c) = (\perp, 1)$ is a delegation violation.

For a deductive defect, take

$$\pi_p = .20, \qquad \pi_c = .10, \qquad q_p = .10, \qquad q_c = .95,$$

compatible for example with

$$P(Y_p = 1, Y_c = 1) = .10, \quad P(Y_p = 1, Y_c = 0) = .10, \quad P(Y_p = 0, Y_c = 0) = .80.$$

The parent scores are $(.80, .20, .10)$, so $a_p = 0$, while the child scores are $(.90, .10, .95)$, so $a_c = \perp$. Thus $(a_p, a_c) = (0, \perp)$ is a deductive defect. Therefore, even oracle nodewise Bayes L2D can be deferral-incoherent under SE: it may either violate the SE contract through a delegation violation or fail deductive closure through a deductive defect. □

E Validity and Structural Properties of Taxonomic Belief Propagation

In this appendix, we prove basic validity and structural properties of the *Taxonomic Belief Propagation* (TBP) mechanism in the main paper (§4.2) under the **Selective-Exclusion** contract (§3.3). Throughout, we consider a single edge $(pa(v) \to v)$ in a tree taxonomy.

E.1 Setup and notation

Let $\boldsymbol{\mu}_{pa(v)}(x) \in \Delta^2$ denote the *unconditional* marginal action distribution of the parent node, and let $\boldsymbol{\eta}_v(\cdot \mid x) \in \Delta^2$ denote the *local primitive* distribution at node v conditioned on the parent being asserted present, as in Eq. (5). That is, for $\mathcal{A} = \{0, 1, \perp\}$,

$$\sum_{a \in \mathcal{A}} \boldsymbol{\mu}_{pa(v)}(a \mid x) = 1, \qquad \boldsymbol{\mu}_{pa(v)}(a \mid x) \ge 0, \tag{15}$$

$$\sum_{a \in \mathcal{A}} \boldsymbol{\eta}_v(a \mid x) = 1, \qquad \boldsymbol{\eta}_v(a \mid x) \ge 0. \tag{16}$$

Under Selective-Exclusion, when the parent is deferred, the child may not be asserted present; the remaining mass is split between 0 and $\perp$ by renormalising $\boldsymbol{\eta}_v$ onto $\{0, \perp\}$. Define

$$\alpha_v(x) := \mathbb{P}(a_v = 0 \mid a_{pa(v)} = \perp, x) = \frac{\boldsymbol{\eta}_v(0 \mid x)}{\boldsymbol{\eta}_v(0 \mid x) + \boldsymbol{\eta}_v(\perp \mid x)}, \tag{17}$$

with the convention $\alpha_v(x) = 1$ when $\boldsymbol{\eta}_v(0 \mid x) + \boldsymbol{\eta}_v(\perp \mid x) = 0$. By construction, $\alpha_v(x) \in [0, 1]$.

E.2 TBP transition kernel and recursion (Selective-Exclusion)

Define the Selective-Exclusion transition kernel $\mathbf{T}_v(x) \in [0,1]^{3\times 3}$ (rows/cols ordered as $[0, 1, \perp]$) by Eq. (7):

$$\mathbf{T}_v(x) = \begin{bmatrix} 1 & 0 & 0 \\ \boldsymbol{\eta}_v(0 \mid x) & \boldsymbol{\eta}_v(1 \mid x) & \boldsymbol{\eta}_v(\perp \mid x) \\ \alpha_v(x) & 0 & 1-\alpha_v(x) \end{bmatrix}. \tag{18}$$

TBP defines the child marginal by passing the parent marginal through the kernel, as in Eq. (8):

$$\boldsymbol{\mu}_v(x) \;=\; \boldsymbol{\mu}_{pa(v)}(x)^\top \, \mathbf{T}_v(x). \tag{19}$$

Expanding (19) yields exactly the main-paper equations:

$$\boldsymbol{\mu}_v(1 \mid x) = \boldsymbol{\mu}_{pa(v)}(1 \mid x)\, \boldsymbol{\eta}_v(1 \mid x), \tag{20}$$

$$\boldsymbol{\mu}_v(0 \mid x) = \boldsymbol{\mu}_{pa(v)}(0 \mid x) + \boldsymbol{\mu}_{pa(v)}(1 \mid x)\, \boldsymbol{\eta}_v(0 \mid x) + \boldsymbol{\mu}_{pa(v)}(\perp \mid x)\, \alpha_v(x), \tag{21}$$

$$\boldsymbol{\mu}_v(\perp \mid x) = \boldsymbol{\mu}_{pa(v)}(1 \mid x)\, \boldsymbol{\eta}_v(\perp \mid x) + \boldsymbol{\mu}_{pa(v)}(\perp \mid x)\, [1-\alpha_v(x)]. \tag{22}$$

E.3 Property (i): Normalisation

Theorem E.1 (Normalisation)**.** *For every x, the TBP output $\boldsymbol{\mu}_v(x)$ is a valid probability distribution: $\sum_{a\in\mathcal{A}} \boldsymbol{\mu}_v(a \mid x) = 1$.*

Proof. Sum Eqs. (20)–(22):

$$\begin{aligned} &\boldsymbol{\mu}_v(0 \mid x) + \boldsymbol{\mu}_v(1 \mid x) + \boldsymbol{\mu}_v(\perp \mid x) \\ &= \boldsymbol{\mu}_{pa(v)}(0 \mid x) + \boldsymbol{\mu}_{pa(v)}(1 \mid x)\big(\boldsymbol{\eta}_v(0 \mid x) + \boldsymbol{\eta}_v(1 \mid x) + \boldsymbol{\eta}_v(\perp \mid x)\big) \\ &\quad + \boldsymbol{\mu}_{pa(v)}(\perp \mid x)\big(\alpha_v(x) + (1-\alpha_v(x))\big) \\ &= \boldsymbol{\mu}_{pa(v)}(0 \mid x) + \boldsymbol{\mu}_{pa(v)}(1 \mid x)\cdot 1 + \boldsymbol{\mu}_{pa(v)}(\perp \mid x)\cdot 1 \\ &= 1, \end{aligned} \tag{23}$$

using that $\boldsymbol{\eta}_v(\cdot \mid x)$ and $\boldsymbol{\mu}_{pa(v)}(\cdot \mid x)$ are normalised. □

E.4 Property (ii): Nonnegativity and bounds

Theorem E.2 (Bounds)**.** *For all $a \in \mathcal{A}$ and all x, $0 \le \boldsymbol{\mu}_v(a \mid x) \le 1$.*

Proof. All terms in Eqs. (20)–(22) are sums and products of nonnegative quantities, and $\alpha_v(x) \in [0,1]$, so $\boldsymbol{\mu}_v(a \mid x) \ge 0$ for each a. By §E.1, the components sum to 1, hence no single component can exceed 1. □

E.5 Property (iii): Contract and coherence constraints are enforced

Theorem E.3 (Coherence constraints as zero-probability transitions)**.** *The TBP kernel enforces the local coherence constraints induced by taxonomic satisfiability, deductive closure, and the Selective-Exclusion contract:*

1. *If the parent is asserted absent, then the child is forced absent: $\mathbb{P}(a_v = 1 \mid a_{pa(v)} = 0, x) = \mathbb{P}(a_v = \perp \mid a_{pa(v)} = 0, x) = 0$.*
2. *If the parent is deferred, the child cannot be asserted present: $\mathbb{P}(a_v = 1 \mid a_{pa(v)} = \perp, x) = 0$.*

Proof. Both claims follow immediately from the corresponding rows of the transition matrix (18). The first row is $[1, 0, 0]$, and the third row is $[\alpha_v(x), 0, 1-\alpha_v(x)]$. □

Proposition 5 (Coherent handoff preserves label consistency)**.** *Let the completed system label be*

$$\hat{y}_v^{\mathrm{sys}}(a, m) = \begin{cases} a_v, & a_v \in \{0,1\}, \\ m_v, & a_v = \perp. \end{cases}$$

If $a \in \mathcal{C}_{\mathrm{SE}}$ and the expert label vector $m \in \mathcal{Y}_{\mathcal{T}}$, then $\hat{y}^{\mathrm{sys}}(a, m) \in \mathcal{Y}_{\mathcal{T}}$.

Proof. For any edge $(p \to c)$, if $a_p = 0$, coherence forces $a_c = 0$. If $a_p = 1$, the completed parent is positive. If $a_p = \perp$, coherence gives $a_c \in \{0, \perp\}$; in the deferred-child case both labels come from the expert, whose labels are hierarchy-consistent. Thus no completed positive child can occur under a completed negative parent. □

E.6 Monotonicity-style implications

Theorem E.4 (Upward-implication-style marginal bound)**.** *For every x, $\boldsymbol{\mu}_v(1 \mid x) \le \boldsymbol{\mu}_{pa(v)}(1 \mid x)$.*

Proof. By (20), $\boldsymbol{\mu}_v(1 \mid x) = \boldsymbol{\mu}_{pa(v)}(1 \mid x)\, \boldsymbol{\eta}_v(1 \mid x)$. Since $\boldsymbol{\eta}_v(1 \mid x) \in [0, 1]$, the inequality follows. □

Theorem E.5 (Effect of parent deferral mass under Selective-Exclusion)**.** *Under Selective-Exclusion, the contribution of parent deferral mass to the child marginal is split between the child's 0 and $\perp$ outcomes. Algebraically, in Eqs. (21)–(22), the coefficient of $\boldsymbol{\mu}_{pa(v)}(\perp \mid x)$ is $\alpha_v(x)$ in $\boldsymbol{\mu}_v(0 \mid x)$ and $1 - \alpha_v(x)$ in $\boldsymbol{\mu}_v(\perp \mid x)$. Thus parent deferral mass is redistributed between child absence and child deferral with weights $\alpha_v(x)$ and $1 - \alpha_v(x)$, respectively.*

Proof. This follows directly from the expanded TBP recursion:

$$\boldsymbol{\mu}_v(0 \mid x) = \boldsymbol{\mu}_{pa(v)}(0 \mid x) + \boldsymbol{\mu}_{pa(v)}(1 \mid x)\boldsymbol{\eta}_v(0 \mid x) + \boldsymbol{\mu}_{pa(v)}(\perp \mid x)\alpha_v(x),$$

and

$$\boldsymbol{\mu}_v(\perp \mid x) = \boldsymbol{\mu}_{pa(v)}(1 \mid x)\boldsymbol{\eta}_v(\perp \mid x) + \boldsymbol{\mu}_{pa(v)}(\perp \mid x)(1 - \alpha_v(x)).$$

The coefficients multiplying $\boldsymbol{\mu}_{pa(v)}(\perp \mid x)$ are therefore $\alpha_v(x)$ and $1 - \alpha_v(x)$. Since $\alpha_v(x) \in [0, 1]$, parent deferral mass is split between child 0 and child $\perp$. This is an algebraic decomposition of the recursion, not an independent perturbation of one coordinate of the parent marginal off the probability simplex. □

E.7 Reverse-KL characterisation of the deferred-parent row

Proposition 6 (Selective-Exclusion renormalisation as an exact KL projection)**.** *Fix a non-root node v and input x. Let*

$$\boldsymbol{\eta}_v(\cdot \mid x) \in \Delta^2$$

be the local primitive distribution over $\mathcal{A} = \{0, 1, \perp\}$, with coordinates ordered as $[0, 1, \perp]$. Assume

$$\boldsymbol{\eta}_v(0 \mid x) + \boldsymbol{\eta}_v(\perp \mid x) > 0,$$

which holds in our parameterisation since $\boldsymbol{\eta}_v$ is produced by a softmax. Under Selective-Exclusion, when the parent is deferred, the admissible child-action distributions are

$$\mathcal{Q}_v^{\perp}(x) := \{q \in \Delta^2 : \; q(1) = 0\}.$$

Then the unique solution of

$$q_v^\star(\cdot \mid x) \in \arg\min_{q \in \mathcal{Q}_v^{\perp}(x)} D_{\mathrm{KL}}\big(q \,\|\, \boldsymbol{\eta}_v(\cdot \mid x)\big) \tag{24}$$

is

$$q_v^\star(\cdot \mid x) = \left[\frac{\boldsymbol{\eta}_v(0 \mid x)}{\boldsymbol{\eta}_v(0 \mid x) + \boldsymbol{\eta}_v(\perp \mid x)},\; 0,\; \frac{\boldsymbol{\eta}_v(\perp \mid x)}{\boldsymbol{\eta}_v(0 \mid x) + \boldsymbol{\eta}_v(\perp \mid x)}\right]. \tag{25}$$

Equivalently,

$$q_v^\star(0 \mid x) = \alpha_v(x), \qquad q_v^\star(\perp \mid x) = 1 - \alpha_v(x),$$

where $\alpha_v(x)$ is defined in Eq. (17).

Proof. Write

$$\eta_0 := \boldsymbol{\eta}_v(0 \mid x), \qquad \eta_1 := \boldsymbol{\eta}_v(1 \mid x), \qquad \eta_\perp := \boldsymbol{\eta}_v(\perp \mid x).$$

Any feasible $q \in \mathcal{Q}_v^{\perp}(x)$ has the form

$$q = (q_0, 0, q_\perp), \qquad q_0 \ge 0, \quad q_\perp \ge 0, \quad q_0 + q_\perp = 1.$$

Substituting this into the reverse KL objective gives

$$D_{\mathrm{KL}}(q\|\boldsymbol{\eta}_v) = q_0 \log \frac{q_0}{\eta_0} + q_\perp \log \frac{q_\perp}{\eta_\perp},$$

since the middle term is $q_1 \log\big(q_1/\eta_1\big) = 0$ by feasibility.

We therefore solve

$$\min_{q_0, q_\perp \geq 0,\ q_0+q_\perp=1} \left\{ q_0 \log \frac{q_0}{\eta_0} + q_\perp \log \frac{q_\perp}{\eta_\perp} \right\}.$$

Introducing a Lagrange multiplier λ for the constraint $q_0 + q_\perp = 1$, the Lagrangian is

$$\mathcal{L}(q_0, q_\perp, \lambda) = q_0 \log \frac{q_0}{\eta_0} + q_\perp \log \frac{q_\perp}{\eta_\perp} + \lambda(q_0 + q_\perp - 1).$$

The first-order optimality conditions are

$$\frac{\partial \mathcal{L}}{\partial q_0} = \log \frac{q_0}{\eta_0} + 1 + \lambda = 0, \qquad \frac{\partial \mathcal{L}}{\partial q_\perp} = \log \frac{q_\perp}{\eta_\perp} + 1 + \lambda = 0.$$

Subtracting the two equations yields

$$\log \frac{q_0}{\eta_0} = \log \frac{q_\perp}{\eta_\perp},$$

and hence

$$\frac{q_0}{\eta_0} = \frac{q_\perp}{\eta_\perp} = c$$

for some constant $c > 0$. Using the normalisation constraint,

$$q_0 + q_\perp = c(\eta_0 + \eta_\perp) = 1,$$

so

$$c = \frac{1}{\eta_0 + \eta_\perp}.$$

Therefore

$$q_0^\star = \frac{\eta_0}{\eta_0 + \eta_\perp}, \qquad q_\perp^\star = \frac{\eta_\perp}{\eta_0 + \eta_\perp},$$

which proves Eq. (25).

Finally, since $\eta_0, \eta_\perp > 0$ under the softmax parameterisation, the map

$$(q_0, q_\perp) \mapsto q_0 \log \frac{q_0}{\eta_0} + q_\perp \log \frac{q_\perp}{\eta_\perp}$$

is strictly convex on the feasible set. Hence the stationary point above is the unique global minimiser of (24). By Eq. (17), its 0 coordinate is exactly $\alpha_v(x)$. □

E.8 Summary

The results above show that TBP under Selective-Exclusion (i) produces valid normalised marginals, (ii) enforces the local coherence constraints by assigning zero probability to forbidden transitions, and (iii) preserves an upward-implication-style monotonicity bound on the *present* action while explicitly characterising how parent deferral probability is redistributed between child 0 and child $\perp$ via $\alpha_v(x)$. Moreover, this redistribution is not heuristic: $\alpha_v(x)$ is exactly the unique reverse-KL projection of the local primitive onto the admissible face that excludes the 1 action under parent deferral.

F Why end-to-end TBP fine-tuning can improve system utility

This appendix expands the non-separability argument from Section 4.3. The Bayes recursion in Section 3.6 shows that an internal-node action should account for downstream subtree value, not only local risk. Stage II fine-tuning through TBP (RPO) can therefore improve *system utility* even when coherence defects are already rare after Stage I. At a high level, Stage II can help for two independent reasons:

1. **Contract-induced non-separability (gating).** Under a deferral contract (Selective-Exclusion), internal-node actions change the feasible action space of descendants. Therefore, the Bayes-optimal *coherent* policy is generally *not* separable across nodes, whereas many Stage I objectives are locally trained (e.g., BR-style L2D, per-node surrogates, or masked classification losses).
2. **Composition/semantics mismatch.** Stage I may optimise predictions under a hierarchy semantics that differs from TBP's *contract-constrained ternary* composition. Even when Stage I is hierarchy-aware (e.g., constraint/closure-based HMLC surrogates such as MCLoss), the resulting parameters need not be stationary for the TBP-induced marginals that are used at inference in our deferral system.

We formalise these points below and give an explicit two-node counterexample with strict improvement.

F.1 A contract-constrained system risk

Consider a taxonomy with a single edge $(p \to c)$ and labels $(y_p, y_c) \in \{0,1\}^2$ satisfying upward implication $y_c = 1 \Rightarrow y_p = 1$. A policy selects actions $a_p, a_c \in \mathcal{A} := \{0, 1, \bot\}$. Under Selective-Exclusion (Def. 3.3), if $a_p = \bot$ then $a_c \in \{0, \bot\}$ (in particular, $a_c \neq 1$).

We evaluate system performance with a simple (yet standard) "predict vs. defer" risk: when the system asserts 0 or 1, it incurs 0–1 error against the ground-truth; when it defers, it incurs the expert's 0–1 error plus a deferral cost $\lambda > 0$. Formally, for node $v \in \{p, c\}$, ground truth y_v, and expert prediction m_v, define

$$\ell(a_v, y_v, m_v) = \begin{cases} \mathbb{I}[a_v \neq y_v], & a_v \in \{0,1\}, \\ \mathbb{I}[m_v \neq y_v] + \lambda, & a_v = \bot. \end{cases} \tag{26}$$

The (expected) system risk is the sum over nodes:

$$R(\pi) := \mathbb{E}_{(x,y,m)}\big[\, \ell(a_p, y_p, m_p) + \ell(a_c, y_c, m_c) \,\big], \tag{27}$$

where (a_p, a_c) are drawn (or decoded) from the policy $\pi(\cdot \mid x)$ subject to Selective-Exclusion.

F.2 Stage I vs. Stage II objectives: a loss-agnostic view

Let $\{\boldsymbol{\eta}_v(\cdot \mid x)\}_{v \in \mathcal{V}}$ denote local primitives produced by a network (Eq. 5 in the main text). TBP defines a deterministic, differentiable composition map

$$\boldsymbol{\mu}(\cdot \mid x) \;=\; F_{\text{TBP}}\big(\boldsymbol{\eta}(\cdot \mid x)\big), \tag{28}$$

via the transition kernels and recursion in Eq. (7)–(8). Stage I training with an arbitrary surrogate can be abstracted as optimising some objective of the form

$$\mathcal{J}_{\text{base}}(\theta) \;=\; \mathbb{E}\Big[\, \sum_{v \in \mathcal{V}} \mathcal{L}_v^{(I)}\big(\boldsymbol{\eta}_v(x;\theta), y_v, m_v\big) \,\Big], \tag{29}$$

where $\mathcal{L}_v^{(I)}$ may be an L2D surrogate, a hierarchy-aware classification loss, a constraint-based objective, or any mixture thereof. Stage II (RPO) instead optimises the *inference-time* TBP-induced marginals:

$$\mathcal{J}_{\text{RPO}}(\theta) \;=\; \mathbb{E}\Big[\, \sum_{v \in \mathcal{V}} \mathcal{L}_v^{(II)}\big(\boldsymbol{\mu}_v(x;\theta), y_v, m_v\big) \,\Big], \qquad \boldsymbol{\mu} = F_{\text{TBP}}(\boldsymbol{\eta}). \tag{30}$$

Even when $\mathcal{L}_v^{(I)} = \mathcal{L}_v^{(II)}$ (same surrogate), the objectives differ because F_{TBP} couples nodes and modifies the effective decision distribution. Hence, Stage I optima need not be stationary for Stage II.

F.3 Myopic (separable) vs. contract-aware optimality

A separable training objective (e.g., BR-style L2D) effectively optimises a nodewise criterion

$$\min_{\pi_p, \pi_c} \; \mathbb{E}\big[\ell(a_p, y_p, m_p)\big] + \mathbb{E}\big[\ell(a_c, y_c, m_c)\big], \tag{31}$$

which treats the choice of a_p as independent of the consequences for the feasible set of a_c. In contrast, under Selective-Exclusion, the optimal decision at p must account for the downstream best response:

$$\underbrace{\mathbb{E}[\ell(a_p, y_p, m_p) \mid x]}_{\text{parent risk}} + \underbrace{\min_{a_c \in \mathcal{A}_c(a_p)} \mathbb{E}[\ell(a_c, y_c, m_c) \mid x]}_{\text{best achievable child risk given } a_p}, \tag{32}$$

where $\mathcal{A}_c(a_p) = \mathcal{A}$ if $a_p = 1$ and $\mathcal{A}_c(a_p) = \{0, \perp\}$ if $a_p = \perp$ (and $\mathcal{A}_c(a_p) = \{0\}$ if $a_p = 0$ under TBP's satisfiability row). Thus, deferring an internal node can have a non-local cost by forcing suboptimal actions downstream.

F.4 A fixed-x counterexample with strict utility gain

We now construct an explicit conditional distribution at a single input $x^\star$ where the locally optimal choice at p is to defer, but the globally optimal coherent policy at that same $x^\star$ asserts $p = 1$ to unlock a much better decision at c. All inequalities are strict.

Proposition 7 (Option value of commitment under Selective-Exclusion)**.** *There exist an input $x^\star$, a conditional label distribution satisfying $y_c = 1 \Rightarrow y_p = 1$, an expert m, a deferral cost $\lambda > 0$, and a model class such that: (i) the nodewise Bayes-optimal action at p conditional on $x^\star$ (minimising $\mathbb{E}[\ell(a_p, y_p, m_p) \mid x^\star]$) is to defer ($a_p = \perp$), but (ii) the globally optimal Selective-Exclusion-admissible policy at that same $x^\star$ attains strictly lower system risk by asserting $a_p = 1$.*

Proof. Fix an input $x^\star$. Conditional on $x^\star$, let

$$\mathbb{P}\big((y_p, y_c) = (1, 1) \mid x^\star\big) = 0.9, \qquad \mathbb{P}\big((y_p, y_c) = (0, 0) \mid x^\star\big) = 0.1.$$

This satisfies $y_c = 1 \Rightarrow y_p = 1$.

Let the parent expert be perfect, so deferring at p incurs conditional risk $\lambda = 0.05$. Let the child expert have error rate 0.4 conditional on $x^\star$, so child deferral incurs risk $0.4 + \lambda = 0.45$. Assume the model class can represent: (a) a child action $a_c = 1$ with conditional risk 0.1 at $x^\star$; (b) a parent action $a_p = 1$ with conditional risk 0.1 at $x^\star$.

(i) Local conditional optimum at the parent. At $x^\star$, asserting $a_p = 1$ has conditional risk 0.1, while deferring has conditional risk 0.05. Hence the nodewise Bayes-optimal parent action conditional on $x^\star$ is

$$\arg\min_{a_p} \mathbb{E}[\ell(a_p, y_p, m_p) \mid x^\star] = \perp.$$

(ii) Global coherent optimum differs at the same $x^\star$. Under Selective-Exclusion, if $a_p = \perp$ then the child cannot assert 1. At $x^\star$, the child therefore chooses between asserting 0 (risk 0.9) and deferring (risk 0.45), so the best admissible child action is $a_c = \perp$. The resulting coherent conditional system risk is

$$R_{\text{defer-parent}}(x^\star) = 0.05 + 0.45 = 0.50.$$

If instead the parent asserts $a_p = 1$, then the child may assert 1 as well. The coherent policy

$$a_p = 1, \qquad a_c = 1$$

therefore has conditional system risk

$$R_{\text{assert-parent}}(x^\star) = 0.10 + 0.10 = 0.20.$$

Thus

$$R_{\text{assert-parent}}(x^\star) = 0.20 < 0.50 = R_{\text{defer-parent}}(x^\star),$$

even though deferring at the parent is locally optimal conditional on the same $x^\star$. □

F.5 Interpretation and link to RPO

Proposition 7 isolates the *contract-induced non-separability* behind Stage II gains: deferring an internal node can be locally optimal (because the expert is strong), yet globally suboptimal because it removes downstream autonomy under the contract, forcing costly deferrals or errors at descendants.

TBP makes descendant marginals depend on ancestor marginals (e.g., for $(p \to c)$, $\boldsymbol{\mu}_c(1 \mid x) = \boldsymbol{\mu}_p(1 \mid x)\,\boldsymbol{\eta}_c(1 \mid x)$. Hence, when RPO optimises a loss on the *unconditional* marginals, gradients from descendant losses backpropagate into ancestor decisions. In the above construction, child loss incurred at $x^\star$ induces a gradient signal that increases $\boldsymbol{\mu}_p(1 \mid x^\star)$, because raising the parent's "present" marginal enlarges the feasible child action set and eliminates forced child deferral.

G Dataset details

G.1 Dataset Licenses and Access Terms

All datasets used in this work were accessed under their respective research-use terms. VinDr-CXR v1.0.0 is a restricted PhysioNet resource released under the PhysioNet Credentialed Health Data License 1.5.0 and the PhysioNet Credentialed Health Data Use Agreement 1.5.0; access requires credentialed PhysioNet status, completion of the required CITI training, and signing the project DUA. CheXpert is distributed through Stanford AIMI/Stanford ML Group; its dataset documentation describes the terms as personal, non-commercial, non-clinical research use. PadChest is distributed under the BIMCV-PadChest Dataset Research Use Agreement, which permits research use without charge, prohibits sale and redistribution, requires approval for non-academic use, and states that the dataset is for research use only. ADPv2 is distributed through Zenodo in three parts under the Creative Commons Attribution-NonCommercial 4.0 International license.

We do not redistribute any medical images, labels, or dataset files. Reproducibility therefore requires users to obtain each dataset directly from the official provider and comply with the corresponding license, data-use agreement, credentialing, and non-clinical-use restrictions.

G.2 VinDr-CXR

Figure 3 shows the annotation-compatible tree used for VinDr-CXR after introducing synthetic grouping nodes and excluding `No finding` from training and evaluation.

G.3 CheXpert

Figure 4 shows the CheXpert taxonomy used to define hierarchy constraints and parent–child handoff coherence.

G.4 PadChest

Figure 5 shows the larger PadChest taxonomy used as a controlled-expert stress test for taxonomy scale.

G.5 ADPv2

Figure 6 shows the ADPv2 tissue taxonomy used to test the framework outside chest radiography.

H Experimental Details

This appendix provides implementation details for dataset construction, taxonomy structure, preprocessing, model architecture, optimisation, projection decoding, budget-swept evaluation, and statistical testing. All details below were checked against the released code and saved experiment configurations.

H.1 Dataset Construction and Taxonomy Summaries

VinDr-CXR. We use the VinDr-CXR v1.0.0 training annotations from PhysioNet, specifically the study-level label file `image_labels_train.csv` together with the radiographs under `images/train/`. The label column `No finding` is excluded. The dataset contains three reader annotations per retained study. For each expert task, one radiologist is treated as the expert m, while the remaining two radiologists are averaged label-wise to form the reference label y. Thus, the

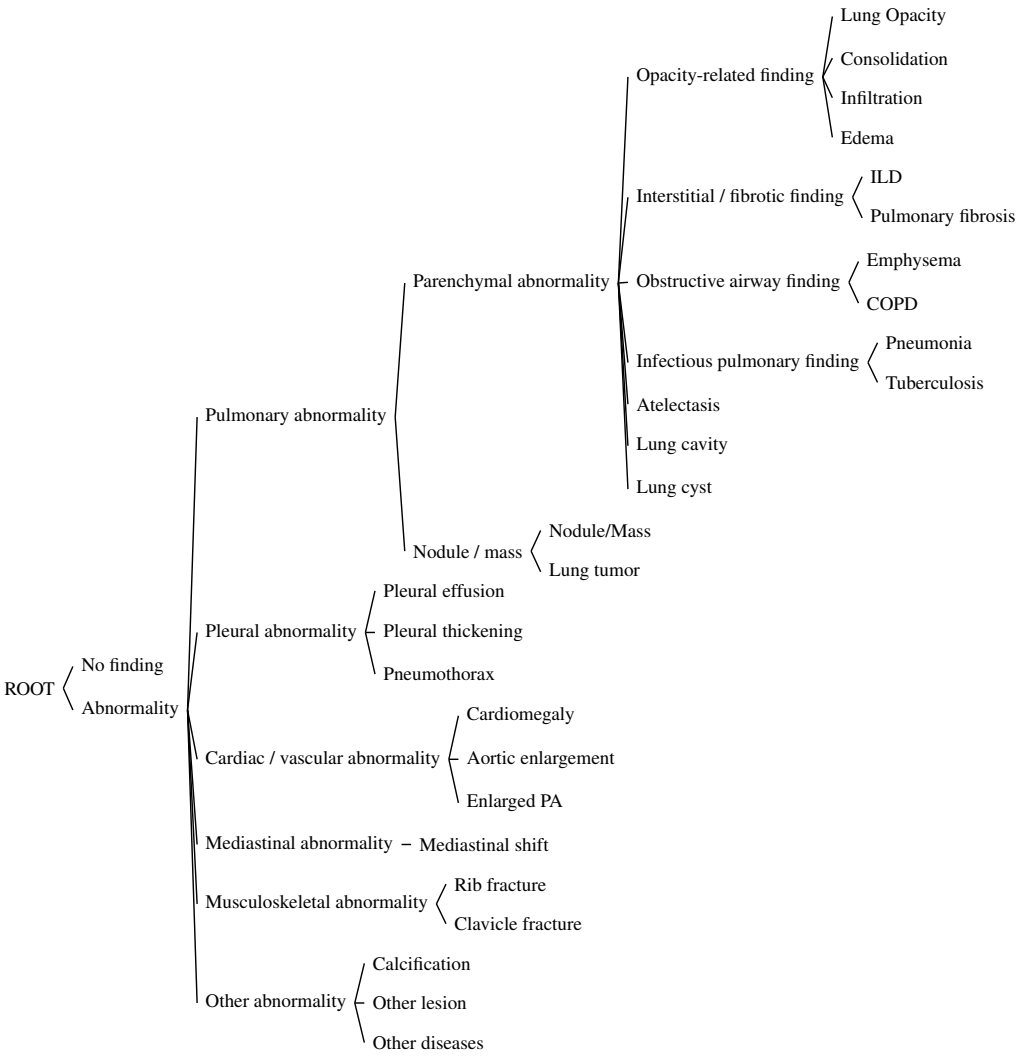


Figure 3: Annotation-compatible chest X-ray taxonomy for the VinDr-CXR dataset. Problematic raw-to-raw implication edges from the original taxonomy are replaced by synthetic internal grouping nodes. The raw source taxonomy contains a `No finding` node under `ROOT`, but all reported experiments exclude `No finding` from training and evaluation.

reference consensus scores are soft values in $\{0, 0.5, 1\}$ at the original leaf labels before hierarchy expansion. The three expert tasks are the three radiologists present in the current pipeline, namely `R8`, `R9`, and `R10`. The resulting sample counts are 6,582 for `R8`, 6,125 for `R9`, and 6,467 for `R10`. The stratified train/validation/test splits are:

$$\texttt{R8} : 3948/1317/1317, \qquad \texttt{R9} : 3675/1225/1225, \qquad \texttt{R10} : 3879/1294/1294.$$

In the main text, VinDr-CXR results are averaged across these three expert tasks.

CheXpert. We use the radiologist-comparison subset distributed through the repository layout under `data/chexpert/`, with reader-specific annotation CSVs `bc1_gt.csv`, `bc2_gt.csv`, `bc3_gt.csv`, `bc5_gt.csv`, together with the associated study directories under `data/chexpert/chexlocalize/`. We exclude `No Finding` and use the 13 CheXpert pathology/device labels listed in the code as base labels. For each expert task, one reader is treated as the expert m, while the remaining four readers are averaged label-wise to form the reference label y. Thus, the reference labels are soft scores in $\{0, 0.25, 0.5, 0.75, 1\}$ at the base labels before hierarchy expansion. The four expert tasks are `bc1`, `bc2`, `bc3`, and `bc5`. Each task contains 500 studies, split by multilabel-stratified sampling into 300 train, 100 validation, and 100 test examples. The raw taxonomy contains 19 non-root nodes, of which 9 are internal and 10 are leaves; in the reported runs, all 19 labels are retained after preprocessing.

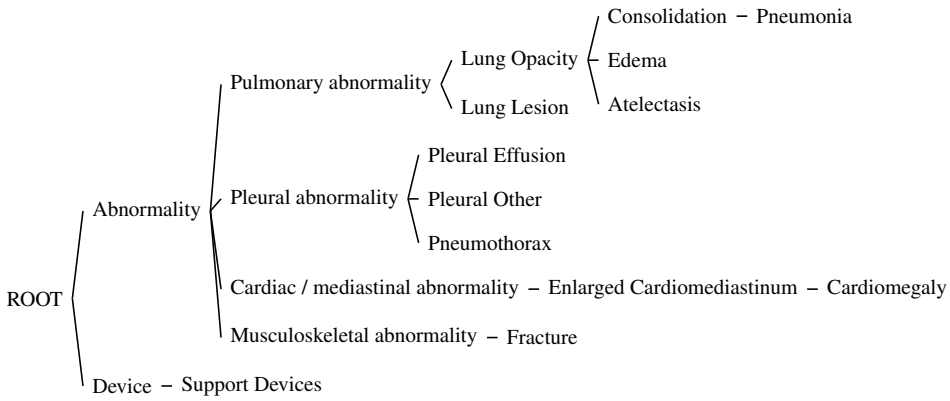


Figure 4: Taxonomy utilised for the CheXpert dataset.

Table 7: **Summary of the real-reader datasets and raw taxonomies used in the main experiments.** For VinDr-CXR, the number of usable studies depends on which radiologist is treated as the expert, because we require exactly three reader annotations and retain only studies containing the target reader. Raw non-root taxonomy counts are reported here; after low-prevalence filtering, the VinDr-CXR experiments retained 34–35 labels depending on the expert, while CheXpert retained all 19 labels. The controlled-expert PadChest and ADPv2 protocols are detailed in the text immediately below.

Dataset	Studies	Readers	Expert Tasks	Nodes	Internal	Leaves
VinDr-CXR	6,125–6,582 per task	3	3	36	13	23
CheXpert	500	5	5	19	9	10

PadChest. We construct the PadChest controlled-expert cohort from cached expert-prediction CSVs together with the PadChest master label CSV. Only `PA` studies with `MethodLabel = Physician` are retained, and any overlap with the RNN-label pool is removed by image ID before splitting. Ground-truth labels are parsed from the PadChest label strings and then upward-closed through the evaluation taxonomy. The resulting retained cohort contains 6,320 studies/images. Splits are patient-disjoint and multilabel-stratified: first 20% of patients are assigned to test, and then 1/8 of the remaining patients are assigned to validation, yielding an overall 70/10/20 patient split. Patient counts are stable across seeds (4,298 train / 615 val / 1,229 test), while study counts vary slightly because the split is patient-level; for example, seed 42 yields 4,413/640/1,267 studies. After applying the PadChest low-prevalence filter (`PADCHEST_MIN_POSITIVE_SAMPLES=10`), the reported runs retain 61 labels (38 leaves and 23 internal nodes). The expert is a precomputed synthetic physician-style predictor loaded from cached per-image predictions rather than trained online during L2D; the codebase supports `pc_carzero` and `pc_carzero_tuned` expert IDs.

ADPv2. We construct the ADPv2 controlled-expert cohort from the `colon_annotations_batch_*` and `colon_metadata_batch_*` CSVs, merged on patch ID, with image paths resolved as `images/<PatchID>_<PatchName>.png`. Ground-truth labels are upward-closed through the ADPv2 taxonomy. Splits are WSI-disjoint by `wsi_name`: 20% of WSIs are assigned to test and then 25% of the remaining WSIs to validation, giving an approximate 60/20/20 split by WSI. Exact train/validation patch counts were not preserved in the current saved run summaries, but archived diagnostics confirm test split sizes between 3,985 and 4,000 patches across the reported seeds and 25 retained labels in all reported runs after applying `ADPV2_MIN_POSITIVE_SAMPLES=10`. The ADPv2 experts are synthetic and built online from handcrafted morphology, colour, and texture features using truth-conditioned profile-specific calibration; available expert IDs are `generalist`, `epi`, `stroma`, and `immune`, with five-fold cross-fitting in the expert-construction pipeline.

Taxonomy definitions. The taxonomies are implemented as explicit child-to-parent maps, i.e. dictionaries of the form

$$\texttt{taxonomy[label]} = \texttt{parent_label},$$

with `"ROOT"` used as the sentinel top node. Auxiliary structures, including immediate-children maps, index-level ancestry maps, descendant matrices, topological orders, and reverse topological orders,



Figure 5: Taxonomy utilised for the PadChest dataset.

are derived from this parent map at runtime. These are dataset-specific engineering taxonomies for coherent deferral experiments: they preserve clinically meaningful implication structure where possible, but may also introduce synthetic grouping nodes to obtain annotation-compatible trees. VinDr-CXR has a single root child (`Abnormality`); CheXpert has two root children (`Abnormality` and `Device`), so the kept structure is a forest. The main-text *Lung Opacity* example is illustrative rather than a literal copy of every dataset taxonomy; in particular, the VinDr-CXR evaluation taxonomy uses an *Opacity-related finding* grouping node and places *Lung Opacity* as a sibling of *Consolidation*, *Infiltration*, and *Edema*.

H.2 Hierarchy Consistency of Reference and Expert Labels

Both the reference labels y and the expert labels m are converted into hierarchy-consistent binary label vectors before hierarchical training and coherence evaluation. The preprocessing pipeline is the following.

- For each expert task, we first construct the expert label matrix m from the target reader and the soft reference matrix y by averaging the remaining readers.

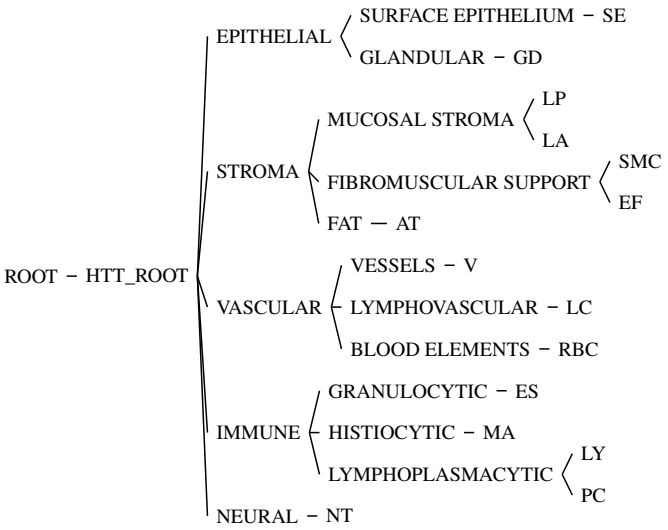


Figure 6: Taxonomy utilised for the ADPv2 dataset.

- We threshold the soft reference matrix at 0.5 to obtain a binary copy used for hierarchy expansion.
- We add missing internal taxonomy nodes and enforce upward closure by propagating positive labels from children to all ancestors.
- For expert labels, we likewise add missing internal nodes, threshold any non-binary values at 0.5, and apply the same upward-closure procedure.
- We then build the final reference matrix by keeping the original soft values on the original annotated labels and inserting hierarchy-expanded binary values for the newly introduced internal nodes.
- Finally, we enforce monotonicity of the reference scores by clipping each parent score from below by the maximum of its kept children.

Thus, upward closure is applied to both the expert labels and a binarised copy of the reference labels. Raw hierarchy violations are not discarded; instead, they are corrected by closure. The current pipeline does not apply any additional remapping for uncertain labels beyond the numeric labels already present in the input CSVs. The `No finding`/`No Finding` columns are excluded before model construction.

The exact order is:

raw reader annotations $\rightarrow$ single-expert split $\rightarrow$
reader averaging for $y \rightarrow$ thresholded copy for hierarchy expansion $\rightarrow$
internal-node insertion and upward closure $\rightarrow$ parent-monotonicity correction for y.

After splitting, we apply low-prevalence filtering on the *training* split only. A label is retained if it has at least 5 positive training examples on VinDr-CXR and at least 3 on CheXpert, where positivity is defined by thresholding at 0.5. All ancestors of retained labels are also retained. This yields 34 kept labels for `R9` and 35 kept labels for `R8`/`R10` on VinDr-CXR, and 19 kept labels for every CheXpert task.

Soft versus hard targets. After reader averaging, each node has a soft consensus score $s_v \in [0, 1]$. For discrete correctness and evaluation, we define the hard majority label

$$Y_v := \mathbb{I}[s_v \geq 0.5].$$

In the reported experiments, the classification term of the defer-aware training loss uses s_v, while expert-correctness terms, correctness-based masking, and all reported F1-, balanced-accuracy-, and accuracy-based metrics use Y_v. The binary label notation used in the theory therefore corresponds to the thresholded consensus label Y_v in experiments; the soft score s_v is used only in the class-fit term during training.

H.3 Model Architecture and Training Hyperparameters

All methods on a given dataset use the same image encoder and the same per-node ternary action heads.

Architecture. For VinDr-CXR, the encoder is a `torchvision` ResNet-18 backbone, initialised with `ImageNet1K_V1` weights. For CheXpert, the code is written to use a `torchxrayvision` DenseNet-121 backbone with weights `densenet121-res224-chex`. For PadChest, the encoder is an `XRVBackbone` built on the chest-X-ray-pretrained TorchXRayVision DenseNet-121. For ADPv2, the encoder is a `ResNet18Backbone`. The X-ray-pretrained backbones are frozen by default in the current codepath.

On top of the encoder, we use a shared projection block followed by one three-way linear head per node. Concretely, if the encoder output is $h(x)$, we compute

$$z(x) = \mathrm{LayerNorm}\big(\mathrm{Dropout}(\mathrm{ReLU}(Wh(x) + b))\big),$$

and then attach one linear map $\mathbb{R}^{256} \to \mathbb{R}^3$ per taxonomy node.

Additional architectural details are:

- shared hidden dimension: 256
- dropout: 0.1
- ternary action heads: one linear head per kept node

Image preprocessing. VinDr-CXR images are loaded as RGB JPEGs, resized to a cached size of 256×256, and then transformed as follows:

- training: random resized crop to 224×224 with scale range $(0.8, 1.0)$, random horizontal flip with probability 0.5, ImageNet normalisation
- validation/test: centre crop to 224×224, ImageNet normalisation

The ImageNet normalisation constants are

$$\mu = (0.485, 0.456, 0.406), \qquad \sigma = (0.229, 0.224, 0.225).$$

CheXpert images are loaded as greyscale images, resized directly to 224×224, randomly flipped horizontally during training with probability 0.5, converted to a single-channel tensor, and normalised to the range $[-1, 1]$ via

$$x \mapsto \frac{x/255 - 0.5}{0.5}.$$

When the `torchvision` DenseNet fallback is used, the single channel is repeated to three channels inside the backbone wrapper before feature extraction.

PadChest uses the same greyscale 224×224 X-ray preprocessing and $[-1, 1]$ normalisation as the XRV-based CheXpert pipeline.

ADPv2 patches are loaded as RGB images. Training uses `RandomResizedCrop(224)`, random horizontal flip, and colour jitter, while validation/test use deterministic resize preprocessing.

Optimisation. Unless otherwise stated, all models are optimised with:

- optimiser: AdamW
- base learning rate: 10^{-3}
- weight decay: 10^{-4}
- batch size: 128
- number of seeds: 5

Checkpoint selection. The balanced-accuracy-stopped BR-family runs reported in the main text use validation **AU-SysBalAcc** as the stopping metric. Earlier runs in the repository also include F1-I-stopped checkpoints, but we do not mix stopping rules within the main utility comparison. Early stopping uses patience 25 and minimum delta 10^{-4}. For each method family, the reported model is the checkpoint with the best validation value of the designated stopping metric under the inference semantics corresponding to that family.

Stage I / Stage II schedule. For BR-L2D and BR (cont.), Stage I optimises the local BR objective

$$\mathcal{J}_{\mathrm{BR}} = \sum_{v \in \mathcal{V}} \mathcal{L}_\phi(\boldsymbol{\eta}_v(x), y_v, m_v).$$

For TBP+RPO, Stage I uses the same objective, and Stage II fine-tunes through TBP using

$$\mathcal{J}_{\mathrm{RPO}} = \sum_{v \in \mathcal{V}} \mathcal{L}_\phi(\boldsymbol{\mu}_v(x), y_v, m_v).$$

The exact schedule in the released runs is:

- Stage I epochs: 100
- Stage II epochs: 100
- learning-rate drop at Stage II: factor 0.1
- continuation schedule: identical to Stage II in epoch count and LR change

Loss function. The defer-aware surrogate $\mathcal{L}_\phi$ used in the reported experiments is the default `mozannar` loss in the code, i.e. the Mozannar–Sontag style defer-aware cross-entropy implemented in `lce_eq7_per_sample`. The codebase also contains Mao-style and RS-style alternatives, but these were not used in the main reported runs.

H.4 Exact Coherent Projection and Dynamic Programs

BR+Projection is an inference-only structured method: training remains standard BR-L2D, but both defer ranking and final action decoding are performed under the coherent projection model induced by the Selective-Exclusion contract.

Coherent feasible set. Let

$$\mathcal{C}_{\mathrm{SE}} = \Big\{ a \in \mathcal{A}^{|\mathcal{V}|} : \forall (p \to c) \in \mathcal{E},\ a_p = 0 \Rightarrow a_c = 0,\ a_p = \perp \Rightarrow a_c \in \{0, \perp\} \Big\}.$$

This is exactly the set of joint action vectors satisfying taxonomic satisfiability, deductive closure, and the Selective-Exclusion parent-deferral contract from Proposition 2. For notational simplicity we present the tree case below; for a forest, the dynamic programs factor independently across roots and the root scores are summed.

Full coherent projection. Ignoring the budget constraint, coherent projection is the MAP decode

$$\hat{\mathbf{a}}^{\mathrm{Proj}}(x) \in \arg\max_{a \in \mathcal{C}_{\mathrm{SE}}} \sum_{v \in \mathcal{V}} \log \boldsymbol{\eta}_v(a_v \mid x). \tag{33}$$

This dynamic program has the same form as the coherent Bayes decoder in Eq. (4), but the scores are different. Eq. (33) uses learned log primitive scores, whereas the Bayes decoder uses conditional risks or utilities. Thus projection is an exact coherent MAP decoder under the learned local action model; it is Bayes-risk optimal only when the local scores are calibrated as Bayes utilities for the target system loss.

Structured action scores for defer ranking. To rank node-level deferrals under a budget, we do not use the local BR marginals directly. Instead, for each node $t \in \mathcal{V}$ and action $a^\star \in \{0, 1, \perp\}$, we define the constrained MAP action score

$$V_t(a^\star \mid x) = \max_{\mathbf{a} \in \mathcal{C}_{\mathrm{SE}}:\, a_t = a^\star} \sum_{v \in \mathcal{V}} \log \boldsymbol{\eta}_v(a_v \mid x). \tag{34}$$

This score measures, under the coherent projection model itself, how compatible each action at node t is with the best globally coherent configuration.

We then define the projection defer-priority score

$$s_{\mathrm{Proj}}(x, t) = V_t(\perp \mid x) - \max\Big(V_t(0 \mid x),\, V_t(1 \mid x)\Big). \tag{35}$$

At budget b, the top $b \cdot N_{\text{total}}$ node-level decisions according to (35) are selected for deferral.

Dynamic program for constrained action scores. Because the taxonomy is a tree, (34) is solved exactly by dynamic programming. Let $\mathrm{Ch}(v)$ denote the immediate children of node v, and let the default admissible child-action sets under Selective-Exclusion be

$$A(i) = \begin{cases} \{0\}, & i = 0, \\ \{0, 1, \perp\}, & i = 1, \\ \{0, \perp\}, & i = \perp. \end{cases}$$

For a fixed target node t and target action $a^\star$, define the clamped admissible set

$$A_v(i; t, a^\star) = \begin{cases} A(i) \cap \{a^\star\}, & v = t, \\ A(i), & v \neq t. \end{cases}$$

If $A_v(i; t, a^\star) = \varnothing$, the corresponding DP value is set to $-\infty$.

For non-root nodes, define

$$G_v(j; t, a^\star) = \log \boldsymbol{\eta}_v(j \mid x) + \sum_{u \in \mathrm{Ch}(v)} F_u(j; t, a^\star),$$

and the Bellman recursion

$$F_v(i; t, a^\star) = \max_{j \in A_v(i;t,a^\star)} G_v(j; t, a^\star).$$

For the root node r, the admissible set is

$$A_r(t, a^\star) = \begin{cases} \{a^\star\}, & r = t, \\ \{0, 1, \perp\}, & r \neq t. \end{cases}$$

The constrained action value is then

$$V_t(a^\star \mid x) = \max_{j \in A_r(t,a^\star)} G_r(j; t, a^\star).$$

In practice, this dynamic program is run once for each pair $(t, a^\star) \in \mathcal{V} \times \{0, 1, \perp\}$.

Projection under a deferral budget. At budget b, let $D_b(x) \subseteq \mathcal{V}$ denote the set of nodes selected for deferral by (35). The final budgeted coherent decode is

$$\hat{\mathbf{a}}_b^{\mathrm{Proj}}(x) \in \arg \max_{\mathbf{a} \in \mathcal{C}_{\mathrm{SE}} \cap \mathcal{F}_b(x)} \sum_{v \in \mathcal{V}} \log \boldsymbol{\eta}_v(a_v \mid x), \tag{36}$$

where

$$\mathcal{F}_b(x) = \left\{ a \in \mathcal{A}^{|\mathcal{V}|} : a_v = \perp \, \forall v \in D_b(x), \quad a_v \in \{0, 1\} \, \forall v \notin D_b(x) \right\}.$$

Thus, selected nodes are clamped to $\perp$, while all remaining nodes are restricted to autonomous actions.

Dynamic program for budgeted coherent decoding. For non-root nodes, define the budget-constrained admissible action set

$$B_v(i; D_b(x)) = \begin{cases} \{\perp\} \cap A(i), & v \in D_b(x), \\ \{0, 1\} \cap A(i), & v \notin D_b(x). \end{cases}$$

For the root node r,

$$B_r(D_b(x)) = \begin{cases} \{\perp\}, & r \in D_b(x), \\ \{0, 1\}, & r \notin D_b(x). \end{cases}$$

Define

$$G_v^b(j) = \log \boldsymbol{\eta}_v(j \mid x) + \sum_{u \in \mathrm{Ch}(v)} F_u^b(j),$$

and

$$F_v^b(i) = \max_{j \in B_v(i; D_b(x))} G_v^b(j).$$

For the root,

$$F_r^b = \max_{j \in B_r(D_b(x))} G_r^b(j).$$

The maximising actions are then recovered by backtracking. This produces the exact MAP action vector under both the coherence constraints and the externally imposed defer budget.

Implementation note. In the released implementation, the raw top-k defer set selected by the global ranking is first passed through a deterministic feasibility closure before the exact dynamic program is run. The raw mask can be infeasible when it forces both an ancestor and a descendant to defer but leaves an intermediate node on that path unforced. Under Selective-Exclusion, such an intermediate node cannot be asserted present below a deferred parent; if it is not forced to defer, the budget mask restricts it to an autonomous action, so it is forced absent, making the lower deferred descendant infeasible. The closure fixes exactly this case by replacing $D_b(x)$ with the minimal superset $D_b^{\mathrm{cl}}(x) \supseteq D_b(x)$ that connects forced-deferred ancestor–descendant pairs by deferred paths.

Concretely, if $p \in D_b^{\mathrm{cl}}(x)$ and the subtree rooted at child c contains any node in $D_b^{\mathrm{cl}}(x)$, then c is added to $D_b^{\mathrm{cl}}(x)$, with this rule applied top-down until no more nodes are added.

Complexity. Since $|\mathcal{A}| = 3$ is fixed and the taxonomy is a tree, both dynamic programs are linear in the number of nodes up to a small constant factor. The budgeted decode is

$$O(|\mathcal{V}|\,|\mathcal{A}|^2),$$

which is linear in practice. The constrained action-score computation requires one such DP for each $(t, a^\star)$ pair, yielding

$$O(|\mathcal{V}|^2\,|\mathcal{A}|^3)$$

in the straightforward exact implementation, which is acceptable for this offline baseline.

H.5 Budget-Swept Utility and Coherence Metrics

Budget sweep. For each image x_i and node v, we compute a method-specific defer-priority score

$$s(x_i, v) = r_{iv}(\perp) - \max\big(r_{iv}(0),\, r_{iv}(1)\big),$$

where r is the method-specific ranking value:

- for BR-L2D and BR (cont.), $r_{iv} = \boldsymbol{\eta}_v(\cdot \mid x_i)$;
- for BR+Projection, $r_{iv} = V_v(\cdot \mid x_i)$, the constrained MAP action scores from (34);
- for TBP+RPO, $r_{iv} = \boldsymbol{\mu}_v(\cdot \mid x_i)$.

Flattening all (i, v) pairs across the evaluation set gives a global ranking over node-level decisions. At budget b, the top $b \cdot N_{\text{total}}$ ranked decisions are clamped to $\perp$, where $N_{\text{total}} = N \cdot L$.

Threshold discretisation and integration. The released runs use `NUM_BUDGET_THRESHOLDS=101`. Concretely, the implementation samples an integer threshold grid by taking the unique values in

$$\mathrm{linspace}(0, N_{\text{total}}, 101 + 1),$$

and always including both endpoints 0 and N_{total}. Thus the number of evaluated thresholds is at most 102, and can be smaller when duplicate integer thresholds arise. All AUC-style quantities are then computed by trapezoidal integration using `sklearn.metrics.auc` over budget fraction.

Why we emphasise balanced metrics. Chest X-ray labels are sparse and strongly negative-dominant. Metrics based on balanced accuracy and macro-F1 therefore provide a more faithful picture of comparative system utility than raw correctness averaged over all study-label decisions. For that reason, the main comparison emphasises AU-SysBalAcc and macro-F1 metrics. These summaries are descriptive, however, and should not be read as a replacement for a deployment-specific cost-sensitive risk model. In particular, the Bayes characterisation in Section 3.6 assumes additive nodewise system risk, whereas F1 and balanced-accuracy AUCs are evaluation summaries over a budget sweep.

AU-SysBalAcc. Let $\hat{Y}^{\mathrm{sys}}(b) \in \{0,1\}^{N \times L}$ denote the system predictions at budget b, and flatten the $(N \times L)$ study-label decisions into one binary vector. Let

$$\mathrm{TPR}(b) = \frac{\mathrm{TP}(b)}{\mathrm{TP}(b) + \mathrm{FN}(b)}, \qquad \mathrm{TNR}(b) = \frac{\mathrm{TN}(b)}{\mathrm{TN}(b) + \mathrm{FP}(b)}.$$

The system balanced accuracy at budget b is

$$\mathrm{BalAcc}^{\mathrm{sys}}(b) = \frac{1}{2}\big(\mathrm{TPR}(b) + \mathrm{TNR}(b)\big).$$

We then define

$$\text{AU-SysBalAcc} = \int_0^1 \mathrm{BalAcc}^{\mathrm{sys}}(b)\, db.$$

F1-L (pooled label-wise F1). Let $\hat{Y}^{\mathrm{sys}}(b) \in \{0,1\}^{N \times L}$ denote the system predictions at budget b. In the current implementation, F1-L is computed by flattening all study-label decisions into one long binary vector and evaluating a single F1 score:

$$\mathrm{SysF1}_L(b) = \mathrm{F1}\Big(\mathrm{vec}\big(\hat{Y}^{\mathrm{sys}}(b)\big),\, \mathrm{vec}(Y)\Big),$$

where Y is the reference label matrix and $\mathrm{vec}(\cdot)$ denotes flattening. This is therefore a pooled or micro-style F1 over study-label decisions rather than a macro average over labels. We then define

$$\text{AU-SysF1-L} = \int_0^1 \mathrm{SysF1}_L(b)\, db,$$

approximated numerically by the sampled threshold grid and trapezoidal integration described above.

AU-F1-Lmacro. The macro label-wise F1 treats each label equally. At budget b,

$$\mathrm{F1}_{L,\mathrm{macro}}(b) = \frac{1}{L}\sum_{v=1}^{L} \mathrm{F1}\Big(\hat{Y}^{\mathrm{sys}}_{:v}(b),\, Y_{:v}\Big),$$

and

$$\text{AU-F1-Lmacro} = \int_0^1 \mathrm{F1}_{L,\mathrm{macro}}(b)\, db.$$

F1-I / AU-F1-Imacro. The instance-wise system F1 computes one F1 score per study and averages across studies:

$$\mathrm{SysF1}_I(b) = \frac{1}{N}\sum_{i=1}^{N} \mathrm{F1}\Big(\hat{Y}^{\mathrm{sys}}_{i:}(b),\, Y_{i:}\Big),$$

and

$$\text{AU-SysF1-I} = \int_0^1 \mathrm{SysF1}_I(b)\, db.$$

In the current codepath, AU-F1-Imacro is populated from this same per-study F1 computation, so AU-F1-Imacro and F1-I are numerically identical. We use the F1-I name in the main text because it is the more direct description of the reported quantity.

Neighbourhood-partition incoherence. Let $\mathcal{P} \subseteq \mathcal{V}$ be the set of kept parent nodes with at least one kept immediate child. For each image i, parent node $p \in \mathcal{P}$, and budget b, define the local neighbourhood action tuple

$$a^{(i)}_{N(p)}(b) = \big(a_{ip}(b), \{a_{ic}(b)\}_{c\in C(p)}\big).$$

Using the partition induced by Proposition 2, define indicator functions

$$\delta_{\mathrm{tax}}(a^{(i)}_{N(p)}(b)), \quad \delta_{\mathrm{ded}}(a^{(i)}_{N(p)}(b)), \quad \delta_{\mathrm{del}}(a^{(i)}_{N(p)}(b)),$$

which take value 1 if the local neighbourhood falls into the corresponding incoherent class and 0 otherwise. The "any incoherence" indicator is

$$\delta_{\mathrm{any}} = \max\{\delta_{\mathrm{tax}},\, \delta_{\mathrm{ded}},\, \delta_{\mathrm{del}}\}.$$

The neighbourhood-partition rates at budget b are then

$$R_{\mathrm{type}}(b) = \frac{1}{N|\mathcal{P}|}\sum_{i=1}^{N}\sum_{p\in\mathcal{P}} \delta_{\mathrm{type}}(a^{(i)}_{N(p)}(b)), \qquad \mathrm{type} \in \{\mathrm{tax}, \mathrm{ded}, \mathrm{del}, \mathrm{any}\}.$$

In the implementation, the partition uses immediate children only and "first violated predicate wins":

1. taxonomic contradiction: parent absent and any child present
2. delegation violation: parent defer and any child present
3. deductive defect: parent absent, no child present, and any child defer
4. coherent: all remaining neighbourhoods

Finally,

$$\text{AU-Neigh-Any} = \int_0^1 R_{\mathrm{any}}(b)\, db,$$

and similarly for the defect-specific neighbourhood-partition rates used throughout the paper.

Edge-weighted incoherence. In addition to the neighbourhood partition, we also compute an edge-weighted view in which the unit of analysis is the immediate parent–child edge. Let $\mathcal{E}_{\text{kept}}$ be the set of kept immediate edges. Then the denominator is

$$N\,|\mathcal{E}_{\text{kept}}|,$$

Table 8: **Pairwise one-sided Wilcoxon tests for the VinDr-CXR main comparison.** Each cell reports the direction favoured by the test together with the unadjusted p-value. "ns" denotes that the reported one-sided comparison was not significant at $p < 0.05$.

Pair	BalAcc	AU-F1-Lmacro	F1-I
BR vs Projection	Projection (0.0473)	Projection (0.0473)	Projection (0.0177)
BR vs cont.	BR (3.1×10^{-5})	cont. (0.0151)	ns
BR vs RPO	ns	RPO (3.1×10^{-5})	ns
Projection vs cont.	Projection (3.1×10^{-5})	cont. (0.0365)	ns
Projection vs RPO	ns	RPO (3.1×10^{-5})	ns
cont. vs RPO	RPO (9.2×10^{-5})	RPO (6.1×10^{-5})	ns

Pair	F1-L	EW Any	NP Any
BR vs Projection	ns	Projection (3.1×10^{-5})	Projection (3.1×10^{-5})
BR vs cont.	cont. (0.0010)	BR (0.0090)	BR (0.0108)
BR vs RPO	RPO (9.2×10^{-5})	RPO (3.1×10^{-5})	RPO (3.1×10^{-5})
Projection vs cont.	cont. (5.8×10^{-4})	Projection (3.1×10^{-5})	Projection (3.1×10^{-5})
Projection vs RPO	RPO (9.2×10^{-5})	Projection (3.1×10^{-5})	Projection (3.1×10^{-5})
cont. vs RPO	RPO (0.0277)	RPO (3.1×10^{-5})	RPO (3.1×10^{-5})

and the edge-weighted defect rates at budget b are

$$R^{\text{edge}}_{\text{tax}}(b) = \frac{1}{N|\mathcal{E}_{\text{kept}}|} \sum_{i=1}^{N} \sum_{(p \to c) \in \mathcal{E}_{\text{kept}}} \mathbb{1}[a_{ip}(b) = 0,\ a_{ic}(b) = 1],$$

$$R^{\text{edge}}_{\text{ded}}(b) = \frac{1}{N|\mathcal{E}_{\text{kept}}|} \sum_{i=1}^{N} \sum_{(p \to c) \in \mathcal{E}_{\text{kept}}} \mathbb{1}[a_{ip}(b) = 0,\ a_{ic}(b) = \perp],$$

$$R^{\text{edge}}_{\text{del}}(b) = \frac{1}{N|\mathcal{E}_{\text{kept}}|} \sum_{i=1}^{N} \sum_{(p \to c) \in \mathcal{E}_{\text{kept}}} \mathbb{1}[a_{ip}(b) = \perp,\ a_{ic}(b) = 1],$$

and

$$R^{\text{edge}}_{\text{any}}(b) = \frac{1}{N|\mathcal{E}_{\text{kept}}|} \sum_{i=1}^{N} \sum_{(p \to c) \in \mathcal{E}_{\text{kept}}} \mathbb{1}\left[\begin{array}{l} (a_{ip}(b) = 0,\ a_{ic}(b) = 1) \vee \\ (a_{ip}(b) = 0,\ a_{ic}(b) = \perp) \vee \\ (a_{ip}(b) = \perp,\ a_{ic}(b) = 1) \end{array}\right].$$

Finally,

$$\text{AU-EW-Any} = \int_0^1 R^{\text{edge}}_{\text{any}}(b)\, db,$$

and similarly for the defect-specific edge-weighted AUCs used in the appendix decomposition tables.

H.6 Statistical Testing

All hypothesis tests are paired one-sided Wilcoxon signed-rank tests. The main-text comparisons use the 15 matched balanced-accuracy-stopped VinDr-CXR runs pooled across R8/R9/R10 and the 20 matched balanced-accuracy-stopped CheXpert runs pooled across bc1/bc2/bc3/bc5. For utility metrics, the alternative hypothesis is that the first method has higher performance; for incoherence metrics, the alternative is that the first method has lower incoherence. All reported p-values are unadjusted.

Table 9: **Per-expert mean utility winners for the VinDr-CXR main comparison.**

Expert	Best BalAcc	Best AU-F1-Lmacro	Best F1-I	Best F1-L
R8	Projection 0.8421	RPO 0.4850	Projection 0.4914	RPO 0.7519
R9	RPO 0.8689	RPO 0.5175	RPO 0.5510	RPO 0.7867
R10	Projection 0.8483	RPO 0.4924	RPO 0.5061	RPO 0.7644

Table 10: **Pairwise one-sided Wilcoxon tests for the pooled CheXpert real-reader sweep.** Each cell reports the direction favoured by the test together with the unadjusted p-value. The comparison pools the 20 matched runs from bc1/bc2/bc3/bc5. "ns" denotes that the reported one-sided comparison was not significant at $p < 0.05$.

Pair	BalAcc	AU-F1-Lmacro	F1-I
BR vs Projection	ns	ns	ns
BR vs cont.	cont. (0.0086)	cont. (2.4×10^{-4})	cont. (0.0133)
BR vs RPO	RPO (3.1×10^{-5})	RPO (3.5×10^{-4})	RPO (0.0012)
Projection vs cont.	cont. (4.3×10^{-4})	cont. (6.0×10^{-4})	cont. (0.0181)
Projection vs RPO	RPO (3.1×10^{-5})	RPO (8.4×10^{-4})	RPO (0.0036)
cont. vs RPO	RPO (0.0200)	ns	ns

Pair	F1-L	EW Any	NP Any
BR vs Projection	ns	Projection (9.5×10^{-7})	Projection (4.4×10^{-5})
BR vs cont.	cont. (0.0266)	ns	ns
BR vs RPO	RPO (1.8×10^{-5})	RPO (9.5×10^{-7})	RPO (9.5×10^{-7})
Projection vs cont.	cont. (0.0047)	Projection (4.4×10^{-5})	Projection (9.5×10^{-7})
Projection vs RPO	RPO (1.8×10^{-5})	Projection (0.0139)	Projection (0.0139)
cont. vs RPO	RPO (0.0148)	RPO (4.4×10^{-5})	RPO (9.5×10^{-7})

H.7 Additional CheXpert Statistics

The main-text CheXpert rows in Table 2 pool the four completed real-reader experts bc1, bc2, bc3, and bc5, with five matched seeds per expert (20 matched runs total); the same pooling is used for the incoherence decompositions. We collect the corresponding pooled significance tests and per-expert utility winners here.

Table 11: **Per-expert mean utility winners for the pooled CheXpert real-reader sweep.** Each entry reports the best method for that expert, averaged over the five matched seeds.

Expert	Best BalAcc	Best AU-F1-Lmacro	Best F1-I	Best F1-L
bc1	RPO 0.8299	RPO 0.5019	RPO 0.6599	RPO 0.7668
bc2	RPO 0.8388	RPO 0.5428	RPO 0.6775	RPO 0.7810
bc3	RPO 0.8353	RPO 0.5111	RPO 0.6278	RPO 0.7633
bc5	RPO 0.8198	cont. 0.5291	RPO 0.6426	RPO 0.7462

H.8 Reproducibility Details

Unless a seed list is explicitly supplied, the code generates the five seeds used in the released experiments via

$$42,\ 153,\ 264,\ 375,\ 486,$$

i.e. `42 + 111*i` for `i=0,...,4`. The reported experiments were run on a workstation with an AMD Ryzen 9 7950X CPU (16 cores / 32 threads), 62 GiB RAM, and a single NVIDIA GeForce RTX 4090 GPU with 24 GiB VRAM. The code runs on CUDA when `torch.cuda.is_available()` is true and otherwise falls back to CPU.

The software stack in the current release environment is:

- Python 3.12.2
- PyTorch 2.6.0
- torchvision 0.21.0
- scikit-learn 1.8.0
- NumPy 2.2.6
- pandas 2.2.3
- iterative-stratification 0.1.9

Code-level implementation details not central to the main text include:

- VinDr-CXR training uses random resized crop and horizontal flip; validation/test use centre crop.
- CheXpert training uses random horizontal flip only; validation/test are deterministic.
- The budget-sweep AUCs use 101 sampled intervals (at most 102 evaluated thresholds including endpoints).

- Projection decoding clamps local probabilities below by 10^{-12} before taking logs.

I Exact Decoding under the TBP Joint Model

The main-text RPO results use a fast budgeted decoder based on TBP marginals. This decoder is efficient and yields only negligible residual incoherence, but it is not itself guaranteed coherent. To separate coherence of the TBP joint model from coherence of the external decoder, we additionally evaluate an exact MAP decoder under the TBP model.

Exact TBP decode. Let

$$P_{\mathrm{TBP}}(\mathbf{a} \mid x) = \boldsymbol{\eta}_r(a_r \mid x) \prod_{v \neq r} \mathbf{T}_v(x)_{a_{pa(v)}, a_v}$$

be the TBP joint model from (9). At budget b, let $D_b^{\mathrm{raw}}(x)$ be the set of nodes selected for deferral by the same marginal ranking used in the main-text RPO experiments. As in Appendix H.4, we apply the deterministic feasibility closure that adds exactly the intermediate nodes needed to connect forced-deferred ancestor–descendant pairs by deferred paths, yielding $D_b^{\mathrm{cl}}(x)$. We then decode

$$\hat{\mathbf{a}}_b^{\mathrm{Exact}}(x) \in \arg \max_{\mathbf{a} \in \mathcal{C}_{\mathrm{SE}} \cap \mathcal{F}_b^{\mathrm{cl}}(x)} \log P_{\mathrm{TBP}}(\mathbf{a} \mid x), \tag{37}$$

where

$$\mathcal{F}_b^{\mathrm{cl}}(x) = \left\{ a \in \mathcal{A}^{|\mathcal{V}|} : a_v = \perp \, \forall v \in D_b^{\mathrm{cl}}(x), \quad a_v \in \{0,1\} \, \forall v \notin D_b^{\mathrm{cl}}(x) \right\}.$$

Because (37) maximises over the coherent feasible set itself, the resulting deterministic action vector is exactly coherent. As with projection, this is exact MAP decoding under the learned TBP joint model, not necessarily Bayes-risk decoding for the additive system loss unless the joint scores are calibrated to that risk.

Table 12: **Exact decoding under the TBP joint model.** Values are mean ± standard deviation over matched runs.

Dataset	F1-L	F1-I	AU-Neigh-Any
VinDr-CXR	0.7806 ± 0.0141	0.5175 ± 0.0247	0.0000 ± 0.0000
CheXpert	0.6983 ± 0.0185	0.6004 ± 0.0303	0.0000 ± 0.0000

As expected, exact decoding drives both edge-weighted and neighbourhood-partition incoherence to zero on both datasets. Its utility trade-off is dataset-dependent: on VinDr-CXR, exact decoding is comparable to the fast marginal RPO decoder and slightly improves the reported F1-L and F1-I values in this comparison, whereas on CheXpert it reduces utility. Thus, exact deterministic coherence under the TBP model is feasible, but the fast marginal decoder remains the preferred high-throughput operating point when its small residual incoherence is acceptable.

Closure diagnostics. Table 13 quantifies the effect of the feasibility closure. In both datasets it is rare and negligible: closure is activated in fewer than 0.4% of budgeted example-threshold pairs on VinDr-CXR and fewer than 0.05% on CheXpert, with mean added deferred nodes below 0.005 and realised-to-raw defer ratios essentially equal to one.

Table 13: **Feasibility-closure diagnostics for exact TBP decoding.**

Dataset	Closure activation rate	Mean added deferred nodes	Max added deferred nodes	Realised/raw defer ratio
VinDr-CXR	0.00380	0.00412	3	1.00024
CheXpert	0.000424	0.000424	1	1.000045

J Inference Runtime

To quantify the computational cost of coherent inference, we benchmarked per-study inference time and full budget-sweep evaluation time for three structured decoders: the fast marginal RPO decoder, the exact TBP MAP decoder from Appendix I, and exact coherent projection. All timings

were measured on CPU, exclude data-loading overhead, and are intended as representative runtime profiles rather than deployment-time service estimates. For exact projection, runtime includes both computation of the constrained MAP action scores $V_v(a \mid x)$ for all nodes and actions and the final budgeted coherent decode. For exact TBP decoding, runtime includes the TBP forward pass and a single exact MAP decode under the budgeted feasible set.

For per-study decode timing, we use a fixed within-study defer fraction of 10%. This is only for runtime benchmarking; the main paper's evaluation still uses the global dataset-level budget sweep. We additionally report the full runtime of the offline budget-swept evaluation with 101 budget thresholds.

Table 14: **Runtime of coherent decoding variants.** Per-study timings are measured on CPU and exclude data-loading overhead. Budget-sweep timings correspond to the full offline evaluation on the test split with 101 budget thresholds. Entries in the final column are reported as per-study speedup / budget-sweep speedup relative to exact projection.

Dataset	Method	Total ms/study	Budget Sweep sec/split	Slowdown vs. fast RPO	Speedup vs. Projection (study / sweep)
VinDr-CXR	BR-L2D+RPO	0.225	0.699	1.00×	1074× / 1287×
	BR-L2D+RPO+Exact Decode	1.178	97.21	5.25×	205× / 9.25×
	BR+Projection	241.71	899.36	1074× / 1287×	1.00×
CheXpert	BR-L2D+RPO	0.611	0.300	1.00×	138× / 123×
	BR-L2D+RPO+Exact Decode	1.045	4.891	1.71×	80× / 7.5×
	BR+Projection	84.287	36.862	138× / 123×	1.00×

These timings show that exact TBP decoding occupies a useful middle ground. It is slower than the fast marginal decoder, especially for full budget-sweep evaluation, but it remains dramatically cheaper than exact projection while eliminating incoherence exactly. This supports the main practical interpretation of the paper: fast RPO is the preferred deployment-time operating point because it delivers strong utility at near-BR cost while maintaining negligible incoherence, and exact TBP decoding provides a coherent deterministic decoder under the learned TBP joint model that is far less expensive than projection.

K Extension Beyond Trees: DAG Taxonomies

The main paper assumes that the taxonomy is a tree. This assumption is useful because it makes coherence local and gives exact linear-time dynamic programs for both Bayes decoding and coherent projection. Many medical vocabularies, however, are more naturally represented as directed acyclic graphs (DAGs), where a finding may have multiple parents. This appendix sketches how the same conceptual framework can be extended to DAG taxonomies, and clarifies which parts of the method transfer directly and which require new inference machinery.

K.1 DAG label space and action coherence

Let

$$\mathcal{G} = (\mathcal{V}, \mathcal{E})$$

be a directed acyclic taxonomy, where an edge $(p \to c) \in \mathcal{E}$ means that child concept c implies parent concept p. As in the tree setting, labels obey upward implication:

$$y_c = 1 \Rightarrow y_p = 1 \qquad \forall (p \to c) \in \mathcal{E}. \tag{38}$$

Equivalently, an absent parent forces all of its descendants absent. The taxonomy-consistent binary label space becomes

$$\mathcal{Y}_{\mathcal{G}} := \left\{ y \in \{0,1\}^{|\mathcal{V}|} : \forall (p \to c) \in \mathcal{E},\ y_c = 1 \Rightarrow y_p = 1 \right\}.$$

For an action vector $a \in \mathcal{A}^{|\mathcal{V}|}$, the compatibility set is defined exactly as before:

$$\mathsf{Comp}(a) := \{ y \in \mathcal{Y}_{\mathcal{G}} : \forall v \in \mathcal{V},\ a_v \neq \perp \Rightarrow y_v = a_v \}.$$

Thus a is taxonomically satisfiable if $\mathsf{Comp}(a) \neq \emptyset$.

Under Selective-Exclusion, a deferred parent may selectively exclude descendants, but may not allow a positive child assertion that commits the deferred parent. For a DAG, the natural edge-wise contract is

$$a_p = \bot \Rightarrow a_c \in \{0, \bot\} \qquad \forall (p \to c) \in \mathcal{E}. \tag{39}$$

Together with deductive closure, this yields the following DAG analogue of the tree constraints:

$$\begin{array}{ll} a_p = 0,\ a_c = 1 & \text{taxonomic contradiction,} \\ a_p = \bot,\ a_c = 1 & \text{delegation violation,} \qquad \forall (p \to c) \in \mathcal{E}. \\ a_p = 0,\ a_c = \bot & \text{deductive defect,} \end{array} \tag{40}$$

These are the same local forbidden parent–child patterns as in Proposition 2. The difference is that, in a DAG, a node may have multiple parents, so its feasible action is determined by the conjunction of all parent constraints.

K.2 Multiple-parent feasibility

Let $P(v) = \{p : (p \to v) \in \mathcal{E}\}$ denote the set of parents of node v. In the tree case, the parent action $a_{pa(v)}$ determines the feasible child-action set through

$$\Gamma_{\mathrm{SE}}(0) = \{0\}, \qquad \Gamma_{\mathrm{SE}}(1) = \mathcal{A}, \qquad \Gamma_{\mathrm{SE}}(\bot) = \{0, \bot\}.$$

For a DAG, the corresponding feasible set for node v is the intersection of all parent-imposed feasible sets:

$$\Gamma_{\mathrm{DAG}}(a_{P(v)}) := \bigcap_{p \in P(v)} \Gamma_{\mathrm{SE}}(a_p). \tag{41}$$

Thus:

$$\Gamma_{\mathrm{DAG}}(a_{P(v)}) = \begin{cases} \{0\}, & \exists p \in P(v) : a_p = 0, \\ \{0, \bot\}, & \nexists p : a_p = 0 \text{ and } \exists p : a_p = \bot, \\ \mathcal{A}, & \forall p \in P(v) : a_p = 1. \end{cases}$$

This has the desired semantics. If any parent is asserted absent, then v is forced absent. If no parent is absent but at least one parent is deferred, then v may be absent or deferred, but cannot be asserted present. Only when all parents are asserted present may v freely choose among $0, 1, \bot$.

The coherent DAG action set can therefore be written as

$$\mathcal{C}_{\mathrm{SE}}^{\mathrm{DAG}} = \left\{ a \in \mathcal{A}^{|\mathcal{V}|} : \forall v \in \mathcal{V},\ a_v \in \Gamma_{\mathrm{DAG}}(a_{P(v)}) \right\}, \tag{42}$$

with the convention that roots have no parent restriction and may take any action in $\mathcal{A}$.

K.3 Coherent projection on a DAG

The exact projection objective transfers directly. Given local primitive distributions $\boldsymbol{\eta}_v(a \mid x)$, the DAG coherent projection is

$$\hat{\mathbf{a}}_{\mathrm{DAG}}^{\mathrm{Proj}}(x) \in \arg \max_{\mathbf{a} \in \mathcal{C}_{\mathrm{SE}}^{\mathrm{DAG}}} \sum_{v \in \mathcal{V}} \log \boldsymbol{\eta}_v(a_v \mid x). \tag{43}$$

The distinction is computational. On a tree, the factor graph induced by the constraints has treewidth one, so dynamic programming gives exact linear-time decoding. On a general DAG, multiple parents create factor loops in the moralised constraint graph, and exact MAP inference need not decompose into independent subtrees.

Several practical routes are available.

Bounded-treewidth exact decoding. If the moralised graph of the taxonomy has small treewidth, Eq. (43) can be solved exactly by junction-tree or variable-elimination methods. The local factors are small because each node still has only three possible actions. This would retain the exact zero-incoherence guarantee, with complexity exponential in graph treewidth rather than in taxonomy size.

Integer-programming decoding. For arbitrary DAGs, coherent projection can be formulated as a small integer linear program. Introduce binary variables

$$z_{v,a} \in \{0,1\}, \qquad v \in \mathcal{V},\ a \in \mathcal{A},$$

where $z_{v,a} = 1$ means $a_v = a$, and impose

$$\sum_{a \in \mathcal{A}} z_{v,a} = 1 \qquad \forall v \in \mathcal{V}.$$

The three forbidden edge patterns in Eq. (40) become linear constraints:

$$z_{p,0} + z_{c,1} \le 1, \tag{44}$$
$$z_{p,\perp} + z_{c,1} \le 1, \tag{45}$$
$$z_{p,0} + z_{c,\perp} \le 1, \tag{46}$$

for all $(p \to c) \in \mathcal{E}$. The projection problem is then

$$\max_{z} \sum_{v \in \mathcal{V}} \sum_{a \in \mathcal{A}} z_{v,a} \log \boldsymbol{\eta}_v(a \mid x) \quad \text{s.t. Eqs. (44)–(46).} \tag{47}$$

This formulation gives an exact coherent decoder for DAGs and is especially attractive for offline evaluation, moderate-sized taxonomies, or safety-critical settings where deterministic zero incoherence is required.

Approximate message passing or local repair. For large DAGs, one could use loopy max-product, dual decomposition, or Lagrangian relaxation to obtain approximate coherent decodes. A simpler engineering alternative is to first decode nodewise, then iteratively repair violated edges until all constraints in Eq. (40) are satisfied. Such repairs would be heuristic unless paired with an exact optimisation certificate, but they may be useful as fast deployment-time approximations.

K.4 TBP-style models on a DAG

The tree version of TBP defines a top-down Markov action model in which each child has one parent. A direct DAG analogue must combine evidence from multiple parents. One natural extension is to replace the single-parent transition by the multiple-parent feasible set in Eq. (41). Let $\boldsymbol{\eta}_v(\cdot \mid x)$ be the local primitive at node v. Conditional on the parent actions $a_{P(v)}$, define

$$P(a_v = j \mid a_{P(v)}, x) = \frac{\boldsymbol{\eta}_v(j \mid x)\, \mathbb{1}\{j \in \Gamma_{\mathrm{DAG}}(a_{P(v)})\}}{\sum_{k \in \Gamma_{\mathrm{DAG}}(a_{P(v)})} \boldsymbol{\eta}_v(k \mid x)}. \tag{48}$$

This conditional is well-defined because $0 \in \Gamma_{\mathrm{SE}}(a_p)$ for every parent action a_p, and therefore $0 \in \Gamma_{\mathrm{DAG}}(a_{P(v)})$ for every parent-action configuration. Since $\boldsymbol{\eta}_v(0 \mid x) > 0$ under the softmax parameterisation, the normalising denominator is strictly positive.

This is the same renormalisation principle used in the tree TBP transition kernel: local primitive mass is projected onto the admissible action face induced by the parents. For roots, $P(a_v \mid x) = \boldsymbol{\eta}_v(a_v \mid x)$. Since $\mathcal{G}$ is acyclic, these conditionals define a valid Bayesian-network-style joint action model:

$$P_{\mathrm{DAG}}(\mathbf{a} \mid x) = \prod_{v \in \mathcal{V}} P(a_v \mid a_{P(v)}, x), \tag{49}$$

where root nodes use their primitive distributions.

By construction, Eq. (49) assigns zero probability to every action vector outside $\mathcal{C}_{\mathrm{SE}}^{\mathrm{DAG}}$. Thus the coherent-support property of TBP extends naturally from trees to DAGs. However, marginal computation is no longer a single top-down pass, because a node's parents may be statistically dependent through shared ancestors. Exact marginal inference in Eq. (49) would again require junction-tree or variable-elimination methods, while approximate inference could use loopy belief propagation, variational methods, or sampling.

K.5 RPO for DAG taxonomies

Recursive Policy Optimisation also extends at the objective level. Stage I would still learn local primitives with a nodewise L2D loss:

$$\mathcal{J}_{\mathrm{StageI}} = \sum_{v \in \mathcal{V}} \mathcal{L}_\phi\big(\boldsymbol{\eta}_v(x), y_v, m_v\big).$$

Stage II would replace the tree-composed marginals $\boldsymbol{\mu}_v$ with marginals under the DAG joint model:

$$\boldsymbol{\mu}_v(a \mid x) := P_{\mathrm{DAG}}(a_v = a \mid x),$$

and optimise

$$\mathcal{J}_{\mathrm{RPO}}^{\mathrm{DAG}} = \sum_{v \in \mathcal{V}} \mathcal{L}_\phi\big(\boldsymbol{\mu}_v(x), y_v, m_v\big). \tag{50}$$

When exact marginals are tractable, this gives a direct DAG analogue of RPO. When they are not, Eq. (50) could be optimised with differentiable approximate inference. Alternatively, one could train against coherent MAP actions or structured surrogate losses derived from the ILP decoder, though this would require handling the non-smoothness of discrete optimisation.

K.6 Practical interpretation

The tree assumption in the main paper is therefore not a conceptual requirement of deferral coherence. The three central ideas all extend to DAGs:

1. deferral coherence is still defined over ternary handoff actions rather than binary labels;
2. Selective-Exclusion still forbids positive child assertions under absent or deferred parents;
3. coherent projection and coherent-support joint action models can still be formulated exactly.

What changes is the inference problem. Trees permit simple linear-time dynamic programs. DAGs require either exact bounded-treewidth inference, integer-programming decoders, or approximate structured inference. Thus, extending the present framework to realistic DAG medical vocabularies is a natural direction for future work: the semantics of coherent handoff are essentially unchanged, but the computational machinery must move from tree recursions to more general structured prediction.

L Additional Experiments and Ablations

L.1 Extension to Multiple Experts

Our primary focus in the main paper is the single-expert setting. We include this subsection to show that the same coherent hierarchical recipe extends naturally to multiple experts in practice. We emphasise that we view this as an empirical extension of the framework, rather than as a new theoretical result.

Let E denote the number of available experts. We replace the single deferral action $\perp$ with expert-indexed deferral actions $\perp_1, \ldots, \perp_E$, so that each node acts on the augmented action space

$$A_E \;:=\; \{0, 1, \perp_1, \ldots, \perp_E\}.$$

The corresponding coherent Bayes rule is obtained from Eq. (4) by replacing the single deferral risk with expert-indexed risks

$$\rho_v(\perp_e \mid x) = P(m_{v,e} \neq Y_v \mid x) + \lambda_{v,e},$$

and optimising over the multi-expert coherent action set.

At each node $v \in V$, we use a dedicated head

$$f_\theta^{(v)}(x) \in \mathbb{R}^{2+E},$$

which induces the local primitive policy

$$\pi_v(a \mid x) \;:=\; \mathrm{Softmax}\big(f_\theta^{(v)}(x)\big)_a, \qquad a \in A_E.$$

Stage I: local multi-expert primitives. In Stage I, we train each node independently with the multi-expert classifier–rejector surrogate. Let $y_v \in \{0, 1\}$ denote the ground-truth label at node v, and let $m_{v,1}, \ldots, m_{v,E} \in \{0, 1\}$ denote the corresponding predictions of the E experts. We define the local loss

$$\mathcal{L}_{\mathrm{ME}}^{(v)}(x, y_v, m_{v,1:E}) = -\log \pi_v(y_v \mid x) - \sum_{e=1}^{E} \mathbb{I}(m_{v,e} = y_v)\, \log \pi_v(\perp_e \mid x),$$

and optimise the Stage I objective

$$J^{\mathrm{ME}}_{\mathrm{base}} = \sum_{v \in V} \mathbb{E}\Big[\mathcal{L}^{(v)}_{\mathrm{ME}}(x, y_v, m_{v,1:E})\Big].$$

Multi-expert Selective-Exclusion contract. We retain the same Selective-Exclusion semantics as in the single-expert setting, but treat the deferral contract as expert-agnostic. Concretely, for every edge $(\mathrm{pa}(v) \to v)$,

$$a_{\mathrm{pa}(v)} \in \{\bot_1, \ldots, \bot_E\} \implies a_v \in \{0, \bot_1, \ldots, \bot_E\}.$$

Together with taxonomic satisfiability, this implies that any positive child assertion must have a positive parent assertion:

$$a_v = 1 \implies a_{\mathrm{pa}(v)} = 1.$$

Thus, if a parent is deferred to any expert, a child may still be excluded or deferred, but it may not be asserted present.

This is an expert-agnostic handoff contract: a parent deferred to one expert may have a child deferred to another expert. A stricter same-expert contract would instead require

$$a_{\mathrm{pa}(v)} = \bot_e \Rightarrow a_v \in \{0, \bot_e\},$$

which may be preferable when responsibility for a finding family should remain with one reader. We leave this workflow choice to deployment.

Multi-expert TBP. For each edge $(\mathrm{pa}(v) \to v)$, we define a transition kernel

$$T_v(x) \in [0,1]^{(2+E)\times(2+E)},$$

with rows and columns indexed by $A_E = \{0, 1, \bot_1, \ldots, \bot_E\}$. The kernel is specified by the following cases:

$$(T_v)_{0,0} = 1, \qquad (T_v)_{0,a} = 0 \ \text{ for all } a \neq 0,$$
$$(T_v)_{1,a} = \pi_v(a \mid x), \qquad a \in A_E,$$

and for each deferred parent state $\bot_e$,

$$(T_v)_{\bot_e,1} = 0, \qquad (T_v)_{\bot_e,a} = \frac{\pi_v(a \mid x)}{Z_v(x)}, \qquad a \in \{0, \bot_1, \ldots, \bot_E\},$$

where

$$Z_v(x) = \pi_v(0 \mid x) + \sum_{e'=1}^{E} \pi_v(\bot_{e'} \mid x).$$

When $Z_v(x) = 0$, we set $(T_v)_{\bot_e,0} = 1$ and $(T_v)_{\bot_e,a} = 0$ for all $a \neq 0$.

Let $\Pi_v(x) \in \Delta^{1+E}$ denote the unconditional marginal action distribution at node v. As in the single-expert setting, TBP propagates these marginals top-down via

$$\Pi_v(x) = \Pi_{\mathrm{pa}(v)}(x)^\top T_v(x).$$

By construction, this recursion assigns zero probability to both taxonomic contradictions (positive child under absent parent) and delegation violations (positive child under deferred parent).

Stage II: recursive policy optimisation. In Stage II, we fine-tune end-to-end through the TBP recursion using the same surrogate applied to the propagated unconditional marginals. Writing $\Pi_v(a \mid x)$ for the a-coordinate of $\Pi_v(x)$, we optimise

$$J^{\mathrm{ME}}_{\mathrm{RPO}} = \sum_{v \in V} \mathbb{E}\left[-\log \Pi_v(y_v \mid x) - \sum_{e=1}^{E} \mathbb{I}(m_{v,e} = y_v)\, \log \Pi_v(\bot_e \mid x)\right].$$

This is the direct multi-expert analogue of the Stage II fine-tuning procedure in the main method: Stage I learns strong local primitives, and Stage II calibrates them under the inference-time hierarchy semantics induced by TBP.

Inference. At test time, we decode top-down using the admissible TBP transitions rather than taking independent nodewise argmaxes of the unconditional marginals. This yields a coherent joint action assignment by construction under the expert-agnostic Selective-Exclusion contract.

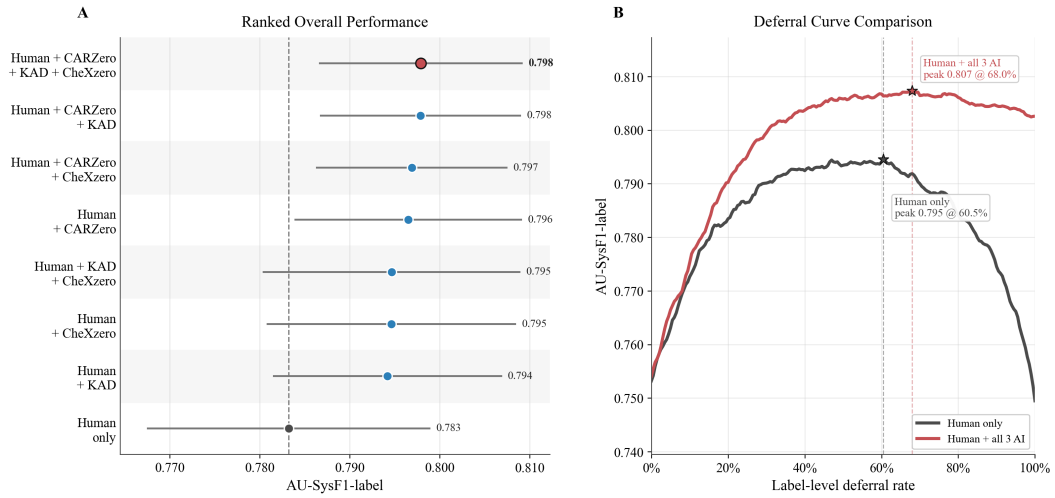


Figure 7: **CheXpert multiple-expert results. (A)** Overall F1-Label for eight human/AI expert combinations across 15 runs per combination (5 readers $\times$ 3 seeds); points show means and bars show ± 1 SD. **(B)** Mean label-level deferral curves comparing Human only and Human + CARZero + KAD + CheXzero; stars mark the peak of each mean curve.

Setup. On CheXpert, we treat one radiologist as the designated human expert and allow deferral to a selected subset of three AI experts: CARZero [32], KAD [33], and CheXzero [34]. This yields $2^3 = 8$ expert combinations, ranging from Human only to Human + CARZero + KAD + CheXzero. For each chosen expert set, we instantiate the multi-expert extension above using the same taxonomy, the same Selective-Exclusion contract, and the same two-stage training recipe as in the single-expert setting. We evaluate all eight combinations across all five radiologists and three random seeds, yielding 15 runs per combination. We report F1-Label, and for the best-performing configuration we additionally plot label-level deferral curves against the human-only baseline.

Results. All AI-augmented combinations outperform the human-only baseline of 0.783 F1-Label. The best mean performance is achieved by Human + CARZero + KAD + CheXzero at 0.798, narrowly ahead of Human + CARZero + KAD at 0.798. Figure 7(B) further shows that the full four-expert system yields a stronger deferral curve than Human only, with a peak F1-Label of 0.807 at 68.0% label-level deferral, compared with 0.795 at 60.5% for Human only. These results suggest that the coherent hierarchical L2D framework transfers cleanly to multiple-expert routing and can improve system utility in this setting.

L.2 Sensitivity to the Stage I objective

Tables 15 and 16 test whether the RPO gains depend on the particular Stage I objective. On VinDr-CXR expert `R10`, RPO is robust across the tested initialisations: MC + RPO, Focal + RPO, and BR-L2D + RPO all achieve strong utility while reducing edge-weighted incoherence to near zero.

The comparison also separates RPO from two alternative explanations. Simply continuing Stage I training does not remove incoherence and gives weaker utility than RPO, showing that the effect is not just longer optimisation. Projection guarantees zero incoherence for every Stage I objective, but can reduce utility substantially when the local primitives are poorly aligned with the coherent decoder, most notably for Focal + Projection. Overall, Stage I provides a useful initialisation, while Stage II RPO is the component that aligns the learned primitives with the TBP inference semantics and improves the utility–coherence trade-off.

Table 15: **Sensitivity to the Stage I objective on VinDr-CXR expert R10: utility.** We compare alternative Stage I objectives and post-Stage I training strategies over five random seeds for reader `R10`. Values are mean ± SD. Higher is better for all metrics.

Method	AU-SysF1(I) ↑	AU-SysF1(L) ↑	AU-SysBalAcc ↑
RPO			
MC + RPO	**0.5199 ± 0.0050**	0.7892 ± 0.0009	0.8530 ± 0.0018
Focal + RPO	0.5198 ± 0.0052	**0.7899 ± 0.0024**	0.8536 ± 0.0023
BR-L2D + RPO	0.5191 ± 0.0052	0.7835 ± 0.0032	0.8503 ± 0.0032
Continue			
MC + Continue	0.5154 ± 0.0047	0.7831 ± 0.0025	0.8434 ± 0.0025
Focal + Continue	0.5131 ± 0.0055	0.7818 ± 0.0028	0.8431 ± 0.0043
BR (cont.)	0.5145 ± 0.0056	0.7817 ± 0.0021	0.8422 ± 0.0012
Projection			
MC + Projection	0.5170 ± 0.0049	0.7529 ± 0.0112	0.8541 ± 0.0035
Focal + Projection	0.3786 ± 0.0072	0.4736 ± 0.0107	0.7221 ± 0.0101
BR-L2D + Projection	0.5158 ± 0.0041	0.7691 ± 0.0077	0.8449 ± 0.0053
Stage I only			
MC Pretrain	0.5168 ± 0.0056	0.7589 ± 0.0083	0.8515 ± 0.0044
Focal Pretrain	0.5117 ± 0.0052	0.7596 ± 0.0060	**0.8602 ± 0.0048**
BR-L2D	0.5148 ± 0.0038	0.7683 ± 0.0074	0.8439 ± 0.0057

Table 16: **Sensitivity to the Stage I objective on VinDr-CXR expert R10: edge-weighted incoherence.** Values are mean ± SD over five random seeds for reader `R10`. Lower is better for all metrics. Tax = taxonomic contradiction, Ded = deductive defect, Del = delegation violation, and Any = any incoherence.

Method	Tax ↓	Ded ↓	Del ↓	Any ↓
RPO				
MC + RPO	0.0000 ± 0.0000	0.0006 ± 0.0002	0.0001 ± 0.0000	0.0007 ± 0.0002
Focal + RPO	0.0000 ± 0.0000	0.0007 ± 0.0001	0.0013 ± 0.0002	0.0020 ± 0.0003
BR-L2D + RPO	0.0000 ± 0.0000	0.0002 ± 0.0001	0.0001 ± 0.0001	0.0003 ± 0.0001
Continue				
MC + Continue	0.0000 ± 0.0000	0.0780 ± 0.0041	0.0114 ± 0.0020	0.0894 ± 0.0044
Focal + Continue	0.0000 ± 0.0000	0.0755 ± 0.0049	0.0129 ± 0.0024	0.0885 ± 0.0066
BR (cont.)	0.0000 ± 0.0000	0.0639 ± 0.0143	0.0144 ± 0.0039	0.0783 ± 0.0177
Projection				
MC + Projection	0.0000 ± 0.0000	0.0000 ± 0.0000	0.0000 ± 0.0000	0.0000 ± 0.0000
Focal + Projection	0.0000 ± 0.0000	0.0000 ± 0.0000	0.0000 ± 0.0000	0.0000 ± 0.0000
BR-L2D + Projection	0.0000 ± 0.0000	0.0000 ± 0.0000	0.0000 ± 0.0000	0.0000 ± 0.0000
Stage I only				
MC Pretrain	0.0004 ± 0.0002	0.0777 ± 0.0264	0.0001 ± 0.0000	0.0781 ± 0.0264
Focal Pretrain	0.0000 ± 0.0000	0.0020 ± 0.0015	0.0040 ± 0.0011	0.0060 ± 0.0025
BR-L2D	0.0000 ± 0.0000	0.0433 ± 0.0180	0.0142 ± 0.0039	0.0575 ± 0.0185

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction state the contract-constrained formulation, the two proposed solutions, and the empirical utility–coherence trade-off, and these claims match the theory, experiments, and conclusion (Abstract, Section 1, Section 5).

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The paper includes a dedicated "Limitations and future work" subsection discussing the tree-taxonomy assumption, single-expert main method, contract choice, retrospective evaluation, simulated experts in ADPv2, and the gap between descriptive AU metrics and deployment-specific policies (Section 6).

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: Section 3 states the tree-taxonomy, open-world, and Selective-Exclusion assumptions, and the appendix provides formal proofs for the TBP properties and the non-separability construction, with a proof sketch in the main text where appropriate (Appendix E, Appendix F).

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: The paper specifies the datasets, splits, taxonomy construction, preprocessing, architectures, optimization, stopping rules, seeds, metric definitions, statistical tests, and runtime protocols used for the main experiments (Section 5, Appendices H.1–J).

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the

case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: We provide a link to code repository in the abstract.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: The appendix gives the dataset construction and splits, label handling, architectures, preprocessing, optimizer, learning rate, weight decay, batch size, stopping rules, seeds, and Stage I/II schedules needed to interpret the experiments (Appendices H.1, H.3, H.8).

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: The main results table marks statistically significant wins, and the appendix reports the paired one-sided Wilcoxon signed-rank tests used for the primary pooled VinDr-CXR and CheXpert comparisons, together with additional variability summaries for ablations (Table 2, Appendix H.6).

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.

- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The paper now reports the workstation CPU/GPU/RAM configuration used for the reported runs, together with software versions and representative runtime measurements for the structured decoders (Appendix H.8, Appendix J).

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: To the best of our knowledge, the work conforms to the NeurIPS Code of Ethics: it studies retrospective medical-imaging datasets and synthetic experts, and we do not report any deviation from the applicable ethical requirements.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

Justification: See Appendix A.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper does not release high-risk generative models, scraped web datasets, or other dual-use assets of the kind targeted by this question; it studies discriminative deferral policies on existing medical-imaging datasets.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: Appendix G.1 lists the license or data-use terms for VinDr-CXR, CheXpert, PadChest, and ADPv2. The code release does not redistribute any dataset files. Users must obtain restricted medical datasets from the official providers under the applicable research-use agreements.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.

- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: The paper introduces methods and empirical results, but it does not release a new dataset, model, or benchmark asset as part of the submission.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The work does not recruit participants or run crowdsourcing experiments; it uses existing retrospective reader annotations and synthetic experts only.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

Justification: The work does not involve prospective human-subject interaction or new data collection by the authors; it uses existing retrospective datasets and synthetic experts.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: LLMs are not an important or non-standard component of the method, theory, or experiments in this research.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.