ROOTROUTE: HYPOTHESIS FORMATION AND SELECTIVE REVISION FOR MULTI-LABEL TEXT ROUTING

Anonymous authors
Paper under double-blind review

ABSTRACT

Multi-label text routing must combine evidence distributed across local spans into document-level concept decisions. Aggregating the strongest local matches provides an initial estimate of concept support, but the resulting score compresses the evidence that produced it and offers limited information about whether the initial judgment should be retained or revised. We propose ROOTROUTE, which explicitly separates hypothesis formation from evidence-guided revision. The Core learns reusable concept directions from representative training evidence and sparsely aggregates local matches into an initial support score. A verifier then revisits the current document with target-conditioned attention and predicts a constrained correction from the retrieved evidence and the initial support state. We instantiate ROOTROUTE for anatomical routing from clinical reports using frozen BioMedCLIP representations and 128,627 training reports from five sources. Across four medical-report evaluations and a controlled-language suite of 20,480 examples, ROOTROUTE consistently improves both F1 and exact-set agreement over the Core. On seen-source reports, macro-F1 increases from 0.919 to 0.963 and exact-set agreement from 0.767 to 0.876. On 2,000 independently expert-mapped RadGraph-XL–Stanford reports, it reaches 0.928 macro-F1 and 0.683 exact-set agreement. Decision-level analysis further shows that revision changes only a small fraction of Core decisions and that these changes are predominantly corrective, while controlled diagnostics show that revisiting document evidence can recover concept support missed by the initial aggregation. These results support explicit separation of initial support formation from selective evidence-guided revision for multi-label text routing, while making the resulting decision changes directly inspectable. Code will be released after the review process.

1 INTRODUCTION

Multi-label text routing maps a document to concepts supported by its content. Evidence may be concentrated in a few passages, while expressions of the same concept vary across documents. Pooling strong matches between passages and learned concept representations provides an initial estimate of support, but compresses the evidence into a score. However, the score compresses the evidence that produced it and offers limited information for distinguishing relevant support from a misleading association. We therefore separate learning reusable concept support across documents from deciding how current-document evidence should revise it.

ROOTROUTE implements this separation through a Core and a target-conditioned verifier (Figure 1). The Core learns shared concept directions through representative-anchor initialization, cross-document evidence alignment, and report-level fitting. For each document, sparse pooling of local matches produces initial support scores. The verifier then revisits the document and combines current evidence with the fixed Core support to predict a signed correction. A candidate gate selects scores eligible for revision, and concept-specific budgets bound each correction. Both stages reuse frozen text representations. To evaluate this formulation, we apply ROOTROUTE to routing medical reports to 16 anatomical concepts using frozen BioMedCLIP text representations (Zhang et al., 2023). Targets indicate the anatomy discussed, which can be supported by normal findings, negated diseases, or descriptions of postoperative absence. Training uses automatically constructed targets

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

from 128,627 reports; unmarked concepts are encoded as zero. Across four medical-report evaluations, including expert-mapped Stanford reports, and a controlled-language suite, the complete system (Full) improves both F1 and exact-set agreement over Core after separate calibration (Table 1). On seen-source reports, macro-F1 increases from 0.919 to 0.963 and exact-set agreement from 0.767 to 0.876. The evaluation also examines individual decision changes at a common operating point, distinguishing revisions that correct errors from those that introduce them. On seen-source reports, 5,642 of 6,771 decision changes correct an error.

Our contributions are:

- We introduce cross-document concept-support learning that combines representative initialization, evidence alignment, and sparse aggregation into explicit initial hypotheses.
- We formulate correction of the Core hypothesis as target-conditioned selective revision. Signed residual updates preserve the initial support as a reference, while candidate eligibility and concept-specific budgets control whether and how far each score may change.
- We introduce ROOTROUTE as a lightweight and inspectable routing framework with 123,951 task-trained parameters on top of a frozen text encoder. In the medical-report setting, it attains competitive routing performance against strong baselines, while retaining explicit evidence windows, initial support scores, and signed revisions for individual concept decisions.

2 Related Work

Multi-label text classification uses label-specific evidence and relationships among output labels. LEAM derives attention from word–label compatibility, while CAML, LSAN, and LAAT form label-dependent document representations through attention over text (Wang et al., 2018; Mullenbach et al., 2018; Xiao et al., 2019; Vu et al., 2020). For long clinical documents, PLM-ICD combines document segmentation with pretrained medical language representations and label-aware attention (Huang et al., 2022). Label dependencies are modeled explicitly by LDGN through statistical and instance-dependent graphs, and by JointLAAT through coarse-code predictions that inform fine-grained coding (Ma et al., 2021; Vu et al., 2020). For larger label spaces, AttentionXML, X-Transformer, and LightXML use hierarchical or candidate-based prediction (You et al., 2019; Chang et al., 2020; Jiang et al., 2021). Related prompt-based image methods also exploit label co-occurrence (Xie et al., 2026).

Alongside label-aware evidence selection, learned class representations provide a basis for estimating concept support. These representations can summarize labeled examples into representative embeddings (Snell et al., 2017) or incorporate label descriptions (Dahiya et al., 2025; Liu et al., 2022). Label-conditioned text representations support prototype updates and comparisons between clinical notes and diagnosis prototypes (Wang et al., 2025; van Aken et al., 2022). Supervised contrastive objectives encourage similar representations for same-class examples, with multi-label extensions accounting for partial label overlap (Khosla et al., 2020; Zhang & Wu, 2024). Once an initial prediction is available, refinement methods can learn additive corrections from label-score vectors (Xun et al., 2020) or perform successive prediction rounds conditioned on earlier label probabilities (Wang et al., 2021). Input evidence provides another basis for refinement: counterfactual image methods combine full-image and local-patch predictions to adjust class scores (Xie et al., 2024). ROOTROUTE connects support learning across documents with revision grounded in target-relevant evidence from the current document.

In clinical applications, report structuring spans labeling, information extraction, and anatomical mapping. CheXpert derives labels through mention, negation, and uncertainty rules, while CheXbert learns labeling from automatically generated and expert annotations (Irvin et al., 2019; Smit et al., 2020). RadGraph and RadGraphXL represent reports through clinical entities and their relations (Jain et al., 2021; Delbrouck et al., 2024), while RadGenome-Chest CT associates report sentences with anatomical regions (Zhang et al., 2024). Retrieval and generation offer additional routes: E5 supports semantic retrieval (Wang et al., 2024), Llama 3 can generate structured textual targets (AI@Meta, 2024), and BioMedCLIP provides biomedical text representations for similarity-based routing (Zhang et al., 2023). ROOTROUTE uses frozen text representations to route anatomical concepts directly, without requiring an intermediate entity graph or generated target list. Because

the explanatory interpretation of attention depends on how faithfulness is defined and evaluated (Jain & Wallace, 2019; Wiegreffe & Pinter, 2019), its inspectable decision record includes source windows, initial scores, signed revisions, and update constraints.

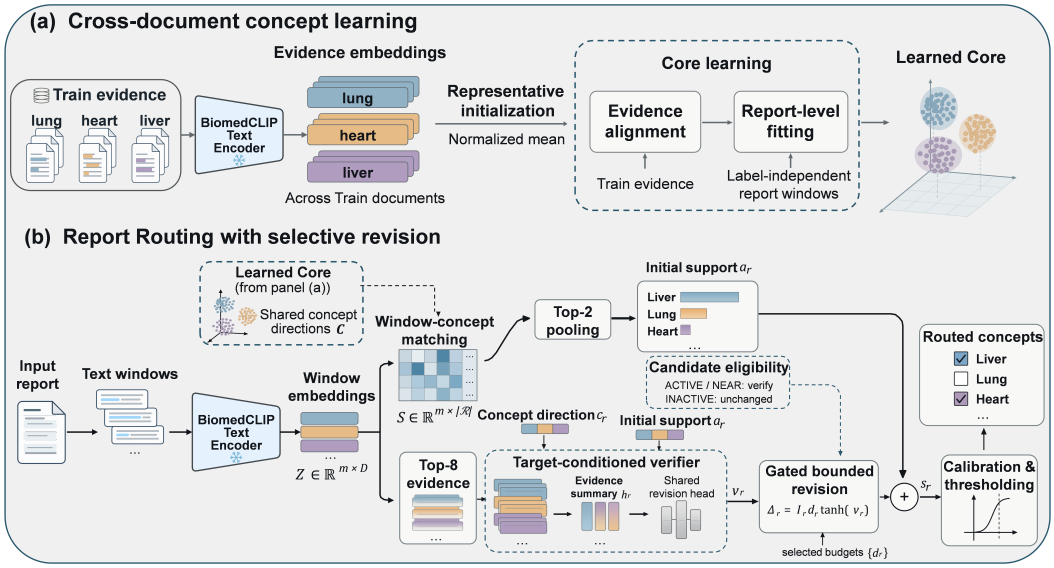

Figure 1: **Overview of ROOTROUTE. a**, Cross-document concept learning through representative initialization, evidence alignment, and report-level Core fitting. **b**, Report routing with initial Core support and target-conditioned, gated, bounded revision.

3 METHOD

3.1 TASK FORMULATION AND OVERVIEW

Let $X = (x_1, \ldots, x_m)$, with $m \geq 1$, denote a document's retained text windows. Multi-label routing predicts the supported subset $\widehat{S}(X) \subseteq \mathcal{R}$ of a fixed concept vocabulary. In our clinical-report application, $\mathcal{R}$ contains 16 anatomical concepts (Roots). The target is the anatomy discussed in the report, rather than only abnormal or physically present structures. Normal findings, negated diseases, and postoperative absence can therefore provide anatomical routing evidence. A frozen lexical label constructor supplies training labels: for report i and concept r, $y_{ir}^{\mathrm{op}} = 1$ denotes an observed-positive association and $y_{ir}^{\mathrm{op}} = 0$ an unmarked entry (Appendix B.2).

Routing uses at most 64 windows per report, constructed independently of concept labels (Appendix A.1). Both stages reuse L2-normalized representations $e_j = \mathrm{normalize}(\mathrm{Enc}(x_j)) \in \mathbb{R}^D$, with $D = 512$, from a frozen BioMedCLIP text encoder (Zhang et al., 2023). ROOTROUTE forms and selectively revises concept support (Figure 1). The Core aggregates matches between window vectors and learned concept directions into initial scores a_r. The verifier revisits target-relevant document evidence to guide signed, bounded updates $s_r = a_r + \Delta_r$ for eligible concepts; ineligible scores remain unchanged. Concept-specific calibration and thresholds produce $\widehat{S}(X)$.

3.2 HYPOTHESIS FORMATION FROM CROSS-DOCUMENT EVIDENCE

Cross-document concept learning. Concept-assigned Train evidence spans are grouped by observed-positive Root for initialization and alignment; report-level fitting and inference use the label-independent routing windows. For each Root, cosine k-medoids selects four representative unit evidence vectors $m_{r1}, \ldots, m_{r4}$, whose normalized mean initializes a single concept direction. Appendix A details window construction, evidence preparation, and medoid selection. With $\bar{c}_r = \mathrm{normalize}(c_r)$, the compatibility between window vector e_j and concept r is:

$$z_{jr} = \langle e_j, \bar{c}_r \rangle + q_r, \qquad q_r = \log \sigma(\rho_r). \tag{1}$$

Here σ is the logistic sigmoid and ρ_r is a learnable reliability logit. Evidence alignment then adapts c_r and ρ_r across training documents. For evidence e assigned to r, let P contain the observed-positive Roots represented by eligible evidence in the same report. With $z_q(e) = \langle e, \bar{c}_q \rangle + \log \sigma(\rho_q)$, the alignment loss is:

$$\ell_{\text{align}}(e, r, P) = -\log \frac{\exp(z_r(e)/T_{\text{align}})}{\sum_{q \in \mathcal{R} \setminus (P \setminus \{r\})} \exp(z_q(e)/T_{\text{align}})}, \tag{2}$$

where $T_{\text{align}} > 0$ is the alignment temperature. The denominator retains the target but excludes other observed-positive Roots, avoiding competition between co-occurring positives. Appendix A details the complete initialization procedure and evidence sampling.

Sparse support aggregation. To form document-level support, the two largest valid window compatibilities for each concept, u_{1r} and u_{2r}, are pooled as

$$a_r(X) = T_{\text{rep}} \log \left[\frac{\exp(u_{1r}/T_{\text{rep}}) + \exp(u_{2r}/T_{\text{rep}})}{2} \right] + b_r, \tag{3}$$

where $T_{\text{rep}} > 0$ is the pooling temperature and b_r is a learned concept bias. For a single-window input, $u_{1r} = z_{1r}$ and $u_{2r} = -1$. Each concept pools its own strongest matches without averaging over all document windows. Appendix A.6 gives the fixed-score pooling properties and their proofs.

Report-level fitting. Starting from the evidence-aligned c_r and ρ_r and zero biases, report-level fitting optimizes c_r, ρ_r, and b_r using concept-weighted binary cross-entropy while keeping the encoder frozen. Section 3.5 summarizes the training stages; Appendix A specifies the loss weights, report/source sampling, and optimization settings.

3.3 Target-Conditioned Verification

The verifier combines current-document evidence with the fixed Core support state to predict a signed revision proposal v_r for each eligible concept.

Candidate eligibility. Let $\pi_r = g_r(a_r)$ denote the output of a concept-specific logistic map, with cutoffs $\eta_r^{\text{near}} \leq \eta_r^{\text{active}}$. Pairs are ACTIVE when $\pi_r \geq \eta_r^{\text{active}}$, NEAR when $\eta_r^{\text{near}} \leq \pi_r < \eta_r^{\text{active}}$, and INACTIVE otherwise. The gate $I_r = \mathbb{I}\{\pi_r \geq \eta_r^{\text{near}}\}$ permits revision for ACTIVE and NEAR pairs. The maps and cutoffs are fitted from report-level fold-excluded Train records; Appendix A.4 specifies their fitting and state-dependent candidate sampling. Eligibility controls permission to revise, while output calibration and decision thresholds are defined separately in Section 3.4.

Target-conditioned evidence. For eligible concept r, let J_r index the $\min(8, m)$ windows with the largest Core compatibilities z_{jr}. Attention logits ℓ_{rj} combine learned query–key matches between $\bar{c}_r$ and e_j with section, position, and Core-compatibility terms. Normalizing over the selected windows:

$$\alpha_{rj} = \frac{\exp(\ell_{rj})}{\sum_{k \in J_r} \exp(\ell_{rk})}, \qquad h_r = \text{normalize}\left(\sum_{j \in J_r} \alpha_{rj} e_j \right). \tag{4}$$

The summary is projected to an evidence representation h_r^{proj}. Appendix A.4 specifies the attention parameterization, metadata encoding, and projection architecture.

Evidence and initial support. The initial Core score a_r summarizes the strongest individual window matches, while absolute target support $\kappa_r = \langle h_r, \bar{c}_r \rangle$ measures the attention-weighted summary's alignment with the target direction. An MLP shared across concepts combines these support measures with the projected evidence to predict the revision proposal $v_r = f_\theta([h_r^{\text{proj}}; a_r; \kappa_r])$. Appendix A.4 specifies the readout architecture.

3.4 Selective Bounded Revision and Final Decisions

Bounded score revision. The proposal v_r is applied to the fixed Core score through a gated, bounded residual:

$$s_r = a_r + \Delta_r, \qquad \Delta_r = I_r d_r \tanh(v_r). \tag{5}$$

The gate I_r controls eligibility, and the nonnegative concept-specific budget d_r bounds the raw-score update. A zero gate or budget preserves the Core score. Budgets are selected on Model-development after checkpoint selection and fixed for evaluation (Appendix A.5; Table 3).

Calibrated routing decisions. With the model and budgets fixed, concept-specific Platt maps fitted on Calibration give $p_r = \sigma(s_r/T_r + \beta_r)$, with $T_r > 0$. Thresholds τ_r then maximize per-concept Calibration F1, yielding $\widehat{S}(X) = \{r \in \mathcal{R} : p_r \geq \tau_r\}$. These outputs are distinct from the Train-fitted eligibility maps g_r in Section 3.3; Appendix B.3 gives the fitting and threshold-selection.

Decision changes at a common threshold. Correction analysis applies Full's fixed operating point to both a_r and s_r, with equivalent raw-score threshold $t_r = T_r[\mathrm{logit}(\tau_r) - \beta_r]$. At this operating point, Core predicts concept r when $a_r \geq t_r$, and Full when $s_r \geq t_r$. Since Equation 5 ensures $|s_r - a_r| \leq I_r d_r$, a decision can change only if $I_r = 1$ and $|a_r - t_r| \leq d_r$ (Appendix A.6). Performance comparisons instead use separately fitted Core and Full operating points. The selected source windows, initial support, signed update, and final decision form an inspectable revision record.

3.5 Stage-Wise Learning

After Core fitting (Section 3.2), the verifier is trained on fixed Core states from five report-group folds. Each fold is excluded from its corresponding report-level Core fit, while Train-wide initialization remains shared. For eligible training pairs, the verifier minimizes:

$$\mathcal{L} = \mathcal{L}_{\mathrm{route}} + \lambda_{\mathrm{direction}}\mathcal{L}_{\mathrm{direction}} + \lambda_{\mathrm{sparse}}\mathcal{L}_{\mathrm{sparse}} + \lambda_{\mathrm{bound}}\mathcal{L}_{\mathrm{bound}}. \tag{6}$$

The routing term uses concept-weighted binary cross-entropy, while the direction term favors label-consistent corrections when one of a target's fixed nearest-concept comparisons carries a different operational label in the same report. The two regularization terms penalize residual magnitude and saturation, while Equation 5 imposes the hard revision bound. Appendix A.4 details the training residual scale, neighbor construction, loss coefficients, and sampling. Verifier checkpoints are selected by macro average precision (AP) on Model-development; after checkpoint selection, concept-specific revision budgets are chosen by per-concept AP on the same split (Appendix A.5). Final output maps and thresholds are fitted on Calibration (Appendix B.3).

4 Experiments

4.1 Evaluation Setting

Data and targets. We use reports from RadGenome-ChestCT, Merlin, MR-RATE, AbdomenAtlas3.0, and M3D-Cap-public, with 128,627 reports for training, 13,927 for model development, 15,904 for calibration, and 17,496 for seen-source testing; source-unseen evaluation uses 29,756 reports from X-Raydar and 2,955 from IU-Xray. Routing targets for these collections are generated by a frozen lexical mapper; a complementary evaluation uses 2,000 Stanford reports with expert Anatomy and organ-identifying Observation annotations. Data sources, group-based partitions, and target construction are detailed in Appendix B. The language suite evaluates ten wording conditions for each of 2,048 held-out base windows, yielding 20,480 probes (Appendix D.1). A separate diagnostic examines evidence distributed across two distinct windows (Section 4.4; Appendix D.2).

Metrics and operating points. Natural-report F1 is macro-averaged over all 16 Roots; Language uses micro-F1. Exact-set agreement requires equality of the predicted and target sets, with unmarked entries encoded as zero under the operational evaluation definition. Performance comparisons use each system's separately fitted or selected operating point, frozen before evaluation. Revision analyses instead apply Full's operating point to both Core and Full. Paired 95% confidence intervals use 10,000 bootstrap resamples. Appendix B.3 specifies the statistical protocol and metric definitions.

4.2 Core Learning and Revision

Core learning. With Core capacity and subsequent report-level training held fixed, representative initialization and evidence alignment contribute little to seen-source F1 but substantially improve transfer (Figure 2(a)). Together, they raise F1 by only 0.4 percentage points on seen-source reports, compared with 6.3 points on Stanford, 7.9 on IU-Xray, and 15.0 on Language. The two stages contribute differently across these conditions: representative initialization accounts for nearly all of the Stanford gain (0.830→0.892→0.893), whereas Language benefits from both initialization and subsequent alignment (0.514→0.630→0.664). Alignment also contributes more on IU-Xray than on seen-source reports, adding 2.8 versus 0.1 F1 points. These gain profiles indicate that

the Core-learning stages primarily improve performance beyond the seen-source distribution rather than further fitting it. The effect is not uniform across metrics. On IU-Xray, the complete learning sequence raises F1 from 0.311 to 0.390 while Exact decreases from 0.425 to 0.408. Thus, stronger concept-level recognition does not necessarily translate into more completely correct target sets. Appendix C.2 specifies the controlled training protocol.

Core-to-Full revision. Once the initial support is formed, revisiting current-document evidence provides an additional but highly non-uniform gain (Table 1). Full improves both F1 and Exact over its corresponding Core in all five evaluations, with the largest gains appearing on the two source-unseen report collections. Exact-set agreement increases by 20.2 points on X-Raydar and 39.9 points on IU-Xray, compared with only 0.9 point on Language; F1 shows the same contrast, improving by 9.0 and 8.6 points on the two source-unseen collections but by only 1.1 points on Language. Thus, the additional value of revision is concentrated more strongly under the tested source shifts than under controlled wording variation. All paired 95% confidence intervals for the Core-to-Full differences remain above zero for both metrics across all five evaluations (Table 6). Revision also improves both metrics with Top-1 and Top-4 pooling (Appendix C.4).

Together with the preceding Core-learning results, the two stages exhibit different gain profiles: Core learning contributes substantially to Language performance before verification, whereas revision provides its largest additional gains on source-unseen reports. This pattern is consistent with complementary roles for reusable support formation and document-specific evidence revision.

Table 1: Overall routing performance and model size. Results use each system's own operating point. F1 is macro-averaged for natural reports and micro-averaged for Language; Exact denotes exact-set agreement. Within the Core-to-Full and capacity-reference block, darker red indicates higher scores. †RadGraphXL is a source-linked reference on Stanford.

System	Params.	Seen-source		X-Raydar		IU-Xray		Stanford		Language	
		F1	Exact	F1	Exact	F1	Exact	F1	Exact	F1	Exact
External application references											
E5-Mistral	7.00 B	0.907	0.647	0.651	0.460	0.353	0.506	0.820	0.545	0.649	0.462
RadGraphXL	156.08 M	0.953	0.772	0.848	0.752	0.518	0.743	0.979†	0.870†	0.586	0.412
LLaMA-3-8B	8.00 B	0.861	0.719	0.747	0.742	0.446	0.842	0.820	0.548	0.765	0.619
Core-to-Full and capacity reference											
Core	109.72 M	0.919	0.767	0.690	0.561	0.390	0.408	0.893	0.596	0.664	0.483
Matched head	109.84 M	0.977	0.924	0.772	0.791	0.512	0.712	0.920	0.622	0.601	0.423
Full (Ours)	109.83 M	0.963	0.876	0.780	0.763	0.476	0.807	0.928	0.683	0.675	0.492

Baseline comparisons. The external interfaces exhibit dataset- and metric-dependent trade-offs rather than a uniform ranking (Table 1). Full outperforms E5-Mistral on both F1 and Exact across all five evaluations, but the comparisons with stronger structured and generative interfaces are mixed. On both source-unseen collections, RadGraphXL attains higher F1 whereas Full achieves higher Exact; on IU-Xray, the comparison with LLaMA-3-8B shows the opposite trade-off. These reversals highlight that stronger concept-level recognition does not necessarily translate into more completely correct target sets, motivating the use of both metrics. The relative strengths also change with the evaluation conditions. LLaMA-3-8B performs best on both Language metrics despite achieving lower F1 than Full on all four natural-report evaluations. Thus, no external interface dominates across datasets and metrics, and ROOTROUTE's relative strength on natural reports does not extend uniformly to controlled wording variation. Implementation protocols and the source-linked status of RadGraphXL on Stanford are detailed in Appendix C.1.

The capacity-matched attention head provides a more controlled comparison with direct classification, using the same frozen representations and a similar number of task-trained parameters (Appendix C.1). It performs better on seen-source reports, whereas Full performs better on Stanford and Language; X-Raydar and IU-Xray again split by metric. The comparison therefore provides no consistent advantage for either direct classification or the two-stage formulation.

ROOTROUTE nevertheless operates at a substantially smaller total model scale than the LLM-based interfaces. Its 109.83M parameters correspond to approximately 1.4–1.6% of the parameter count of the 7–8B baselines, while remaining competitive on the source-unseen evaluations. On the expert-mapped Stanford set, it achieves the highest F1 and Exact among the source-independent comparison interfaces.

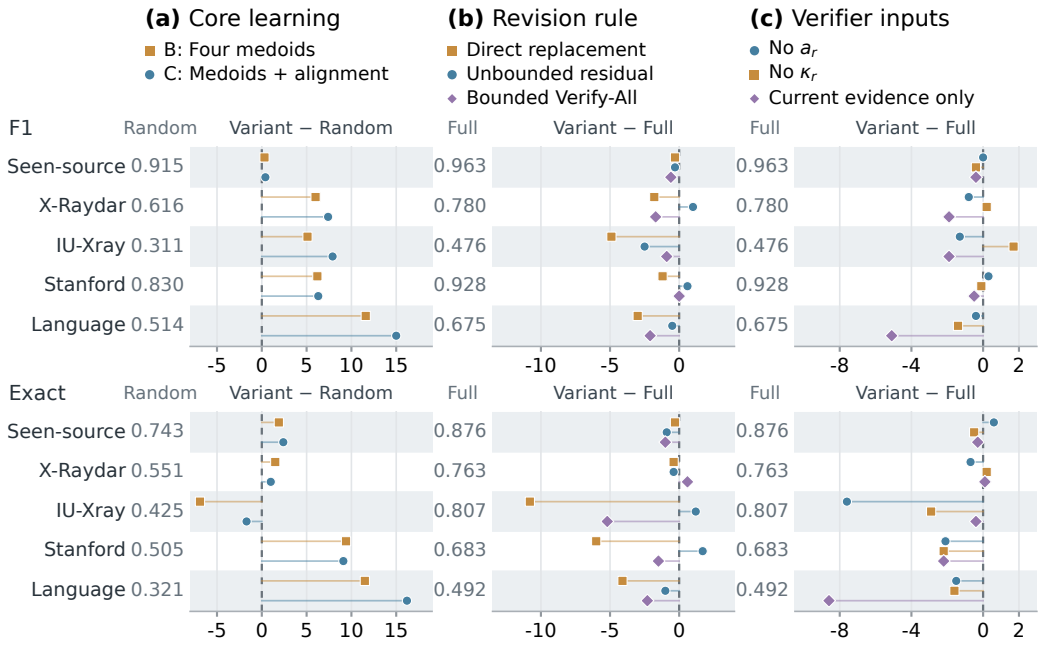

Figure 2: Stage-wise model analysis. Differences from Random in (a) and Full in (b,c) are shown in percentage points; gray numbers give baseline scores.

4.3 Design of Selective Revision

Revision rules. All controls fix the Core, retrain the verifier, and fit separate operating points on Calibration (Appendix C.3). Direct replacement retains the verifier inputs and candidate gate but assigns verifier outputs directly to eligible scores. Its lower F1 and Exact across all five evaluations (Figure 2(b)) favor using the verifier to adjust an existing support state under the tested configurations. The Exact difference is largest on IU-Xray, where Full reaches 0.807 compared with 0.699 for replacement, followed by Stanford at 0.683 versus 0.623.

Expanding the scope of revision does not produce consistent gains. Bounded Verify-All permits every Root to be revised, yet Full achieves higher Exact on four evaluations, with X-Raydar as the exception. Additional opportunities to intervene therefore do not necessarily improve complete-set prediction. Removing the hard bound produces a different pattern, with advantages that vary across evaluations and metrics. The evidence thus supports selective intervention more consistently than a general accuracy advantage from bounding its magnitude. The hard bound provides an explicit limit on permitted score changes, while its predictive benefit remains dependent on the evaluation setting.

Verifier inputs. Initial support also contributes beyond providing the residual starting score. The Current evidence only control removes a_r and κ_r while retaining target-conditioned attention, the Core residual, candidate eligibility, and revision budgets. Its lower F1 across all five evaluations (Figure 2(c)) shows that keeping the Core in the residual path does not eliminate the value of exposing its support state to the verifier. The largest effect occurs on Language: including the support scalars raises F1 from 0.624 to 0.675 and Exact from 0.406 to 0.492. The corresponding F1 gains on natural reports range from 0.4 to 1.9 percentage points. The input comparison therefore identifies wording variation as a setting where the verifier benefits particularly from explicit support inputs. Removing either scalar individually has mixed effects, so these results do not establish that each scalar is individually necessary.

4.4 Decision Changes and Recovery Limits

Decision changes. To isolate the effect of revision from recalibration, we apply Full's fixed calibration and thresholds to both Core and Full and track the resulting binary decision changes. Revision alters only 1.6–2.9% of report–Root decisions across the five evaluations, yet corrected errors outnumber newly introduced errors in every case (Table 2). On seen-source reports, 5,642 of 6,771 changed decisions fix a Core error. By contrast, the balance is much narrower on Stanford and Language, where corrections account for 57.0% and 51.9% of changes, respectively. On Language in particular, 2,801 corrected decisions are nearly offset by 2,596 newly introduced errors, leaving only 205 fewer binary errors at the common operating point.

The direction of correction also depends on the evaluation. On seen-source reports, most beneficial changes remove extra predictions, whereas on the other four evaluations they more often recover concepts missed by the Core. Revision therefore does not exhibit a fixed tendency to add or remove labels; its net effect follows the dominant error structure of the initial predictions. At report level, more examples enter than leave exact-set agreement in every evaluation. Thus, although revision changes only a small fraction of individual decisions, its aggregate effect remains net corrective for both Root-level predictions and complete target sets at the shared operating point.

Table 2: Decision transitions at Full's fixed operating point. Flip rates use all report–Root decisions; other entries are counts. Into/Out count reports entering/leaving exact-set agreement.

Evaluation	Flip (%)	Corrected errors		Introduced errors		Exact	
		FN→TP	FP→TN	TP→FN	TN→FP	Into	Out
Seen-source	2.4	1,364	4,278	724	405	2,786	313
X-Raydar	1.9	3,930	2,633	1,412	918	4,921	1,449
IU-Xray	2.9	869	301	179	16	957	137
Stanford	2.9	354	178	280	121	227	139
Language	1.6	1,822	979	1,026	1,570	1,964	1,369

Remaining Core errors. Full corrects 77.7% of initial Core errors on seen-source reports, 53.2% on X-Raydar, 73.5% on IU-Xray, and 48.9% on Stanford. An error can survive for three distinct reasons: the candidate gate excludes it, the permitted revision range cannot reach the correct side of the boundary, or the verifier fails to produce a successful update despite having sufficient range.

For the four natural-report evaluations, missed eligibility is relatively rare: the gate covers 95.0–99.5% of initial Core errors while admitting only 16.0–42.7% of all report–Root pairs. Revision budgets impose a more visible constraint in some settings. On Stanford, 20.3% of initial Core errors are eligible but cannot reach the correct decision within the permitted range, compared with only 5.8% on seen-source reports.

The dominant remaining failure, however, occurs even when correction is both eligible and geometrically permitted. Among uncorrected initial errors, 50.5–83.2% fall into this category across the natural-report evaluations. For 82.2–87.2% of these cases, the realized update moves away from the correct decision; the remainder moves in the correct direction but does not cross the threshold. Thus, for errors already admitted by the gate and reachable within the revision budget, choosing an effective update direction is a larger limitation than insufficient update magnitude. Language exhibits an additional bottleneck: the gate itself excludes 31.1% of initial Core errors, so failures arise from both missed revision opportunities and ineffective updates. Appendix C.6 defines the decomposition and update-direction criteria, and Table 7 reports the complete results.

Controlled support recovery. We vary which members of a Root pair are supported to examine whether revision recovers the appropriate subset. Two held-out templates for each of 120 unordered pairs produce 960 views: neither target, either target alone, or both targets in independently encoded windows. Neither-target inputs mention another anatomical concept. Figure 3 evaluates Core and Full at Full's fixed operating point, scoring the two tested Roots jointly.

With both targets supported, accuracy increases from 0.554 to 1.000. Core misses one target in 95 views and both targets in 12; Full recognizes both in all 240 views. With one target supported,

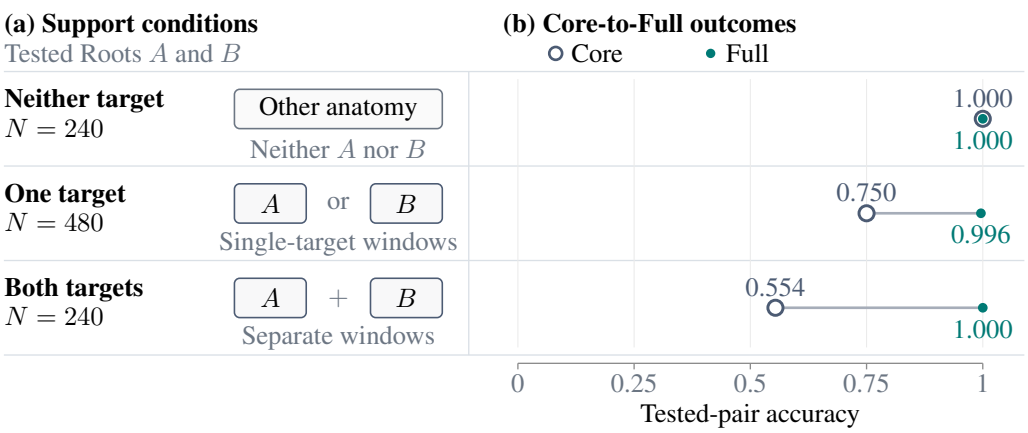

Figure 3: Controlled recovery of the supported Root subset. **(a)** Inputs support neither tested Root, one Root, or both Roots in separate windows; single-target conditions are pooled. **(b)** Core and Full use Full's fixed calibration and thresholds. Tested-pair accuracy requires both designated predictions to match their labels; the other 14 Roots are excluded.

accuracy increases from 0.750 to 0.996. Core omits the target in 120 of 480 views. Full includes the supported target in every view, but also selects its unsupported partner in two. When neither target is supported, both systems reject the pair in all 240 views. Together, these outcomes show recovery of omitted support alongside selection of the appropriate subset in the tested constructions. Appendix D.2 specifies input construction, encoding, and scoring.

Language sensitivity. The controlled-language suite reveals that ROOTROUTE is robust to some surface variation but remains sensitive when the anatomical target becomes less directly recoverable from the wording. Across the same 2,048 base windows, fixed spelling and morphological changes leave target recall essentially unchanged relative to the original wording, indicating that the routing decision is not tied to an exact surface form. Performance degrades instead when the target is expressed through more indirect or compressed language, with the largest failures occurring for abbreviations, typographic corruption, and postoperative/absent descriptions.

These failures are predominantly omissions rather than indiscriminate over-routing. In the three weakest conditions, target recall falls to 0.250–0.270, while the number of additional predicted Roots remains only 0.063–0.074 per probe on average. The main effect of difficult wording is therefore loss of the intended association, not broad activation of unrelated concepts. This interpretation is reinforced by the contrast between negated-pathology and postoperative/absent expressions: both retain the anatomical target under the routing definition, yet the former is recovered reliably whereas the latter is not. The limitation is thus not negation itself, but whether the relevant anatomy remains accessible to the learned support representation under a given linguistic form. Full condition-wise results and constructions are reported in Appendix D.1.

5 Conclusion

ROOTROUTE formulates multi-label text routing as explicit initial support formation followed by selective evidence-guided revision. The Core learns reusable concept support across documents, while the verifier revisits current-document evidence to decide whether that initial judgment should be retained or corrected. Across the evaluated settings, the two stages exhibit different roles: Core learning primarily strengthens the initial support under source and wording changes, while revision provides additional gains by correcting the retained hypothesis. Decision-level analysis further shows that revision changes only a small fraction of predictions and is net corrective overall. Controlled diagnostics show that it can recover support omitted by the initial aggregation without simply activating more concepts, while the remaining errors reveal that allowing a correction does not guarantee an effective update. These findings support hypothesis formation followed by selective evidence-guided revision as a useful formulation for multi-label text routing, while motivating more reliable revision under difficult language variation and broader target spaces.

AI USE STATEMENT

Generative AI tools were used to assist with and refine the writing. Experimental results and specific implementation details were provided by the authors. The authors are responsible for the final manuscript, references, and research conclusions.

REFERENCES

AI@Meta. Llama 3 model card, 2024. URL https://huggingface.co/meta-llama/Meta-Llama-3-8B-Instruct.

Fan Bai, Yuxin Du, Tiejun Huang, Max Q.-H. Meng, and Bo Zhao. M3D: Advancing 3D medical image analysis with multi-modal large language models. *arXiv preprint arXiv:2404.00578*, 2024. URL https://arxiv.org/abs/2404.00578.

Pedro R. A. S. Bassi, Mehmet Can Yavuz, Kang Wang, Xiaoxi Chen, Wenxuan Li, Sergio Decherchi, Andrea Cavalli, Yang Yang, Alan Yuille, and Zongwei Zhou. RadGPT: Constructing 3D image–text tumor datasets. *arXiv preprint arXiv:2501.04678*, 2025. URL https://arxiv.org/abs/2501.04678.

Louis Blankemeier et al. Merlin: A vision language foundation model for 3D computed tomography. *arXiv preprint arXiv:2406.06512*, 2024. URL https://arxiv.org/abs/2406.06512v1.

Wei-Cheng Chang, Hsiang-Fu Yu, Kai Zhong, Yiming Yang, and Inderjit S. Dhillon. Taming pretrained transformers for extreme multi-label text classification. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 3163–3171, 2020. doi: 10.1145/3394486.3403368. URL https://doi.org/10.1145/3394486.3403368.

Yashin Dicente Cid, Matthew Macpherson, Louise Gervais-Andre, Yuanyi Zhu, Giuseppe Franco, Ruggiero Santeramo, et al. Development and validation of open-source deep neural networks for comprehensive chest x-ray reading: A retrospective, multicentre study. *The Lancet Digital Health*, 6(1):e44–e57, 2024. doi: 10.1016/S2589-7500(23)00218-2. URL https://doi.org/10.1016/S2589-7500(23)00218-2.

Kunal Dahiya, Diego Ortego, and David Jimenez-Cabello. Prototypical extreme multi-label classification with a dynamic margin loss. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 10709–10727, 2025. doi: 10.18653/v1/2025.naacl-long.537. URL https://aclanthology.org/2025.naacl-long.537/.

Jean-Benoit Delbrouck, Pierre Chambon, Zhihong Chen, Maya Varma, Andrew Johnston, Louis Blankemeier, Dave Van Veen, Tan Bui, Steven Truong, and Curtis Langlotz. RadGraph-XL: A large-scale expert-annotated dataset for entity and relation extraction from radiology reports. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 12902–12915, 2024. doi: 10.18653/v1/2024.findings-acl.765. URL https://aclanthology.org/2024.findings-acl.765/.

Dina Demner-Fushman, Marc D. Kohli, Marc B. Rosenman, Sonya E. Shooshan, Laritza Rodriguez, Sameer Antani, George R. Thoma, and Clement J. McDonald. Preparing a collection of radiology examinations for distribution and retrieval. *Journal of the American Medical Informatics Association*, 23(2):304–310, 2016. doi: 10.1093/jamia/ocv080. URL https://doi.org/10.1093/jamia/ocv080.

Forithmus. MR-RATE: A vision-language foundation model and dataset for magnetic resonance imaging. Official dataset and code release, 2026. URL https://huggingface.co/datasets/Forithmus/MR-RATE.

Chao-Wei Huang, Shang-Chi Tsai, and Yun-Nung Chen. PLM-ICD: Automatic ICD coding with pretrained language models. In *Proceedings of the 4th Clinical Natural Language Processing Workshop*, pp. 10–20, 2022. doi: 10.18653/v1/2022.clinicalnlp-1.2. URL https://aclanthology.org/2022.clinicalnlp-1.2/.

Jeremy Irvin, Pranav Rajpurkar, Michael Ko, Yifan Yu, Silviana Ciurea-Ilcus, Chris Chute, Henrik Marklund, Behzad Haghgoo, Robyn Ball, Katie Shpanskaya, Jayne Seekins, David A. Mong, Safwan S. Halabi, Jesse K. Sandberg, Ricky Jones, David B. Larson, Curtis P. Langlotz, Bhavik N. Patel, Matthew P. Lungren, and Andrew Y. Ng. CheXpert: A large chest radiograph dataset with

uncertainty labels and expert comparison. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(1):590–597, 2019. doi: 10.1609/aaai.v33i01.3301590. URL https://ojs.aaai.org/index.php/AAAI/article/view/3834.

Saahil Jain, Ashwin Agrawal, Adriel Saporta, Steven Q. H. Truong, Du Nguyen Duong, Tan Bui, Pierre Chambon, Yuhao Zhang, Matthew P. Lungren, Andrew Y. Ng, Curtis P. Langlotz, and Pranav Rajpurkar. RadGraph: Extracting clinical entities and relations from radiology reports. *arXiv preprint arXiv:2106.14463*, 2021. URL https://arxiv.org/abs/2106.14463.

Sarthak Jain and Byron C. Wallace. Attention is not explanation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 3543–3556, 2019. doi: 10.18653/v1/N19-1357. URL https://aclanthology.org/N19-1357/.

Ting Jiang, Deqing Wang, Leilei Sun, Huayi Yang, Zhengyang Zhao, and Fuzhen Zhuang. LightXML: Transformer with dynamic negative sampling for high-performance extreme multi-label text classification. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(9): 7987–7994, 2021. doi: 10.1609/aaai.v35i9.16974. URL https://ojs.aaai.org/index.php/AAAI/article/view/16974.

Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. In *Advances in Neural Information Processing Systems*, volume 33, pp. 18661–18673, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/d89a66c7c80a29b1bdbab0f2a1a94af8-Abstract.html.

Han Liu, Feng Zhang, Xiaotong Zhang, Siyang Zhao, Junjie Sun, Hong Yu, and Xianchao Zhang. Label-enhanced prototypical network with contrastive learning for multi-label few-shot aspect category detection. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 1079–1087, 2022. doi: 10.1145/3534678.3539340. URL https://doi.org/10.1145/3534678.3539340.

Qianwen Ma, Chunyuan Yuan, Wei Zhou, and Songlin Hu. Label-specific dual graph neural network for multi-label text classification. In *Proceedings of ACL-IJCNLP*, pp. 3855–3864, 2021. doi: 10.18653/v1/2021.acl-long.298. URL https://aclanthology.org/2021.acl-long.298/.

James Mullenbach, Sarah Wiegreffe, Jon Duke, Jimeng Sun, and Jacob Eisenstein. Explainable prediction of medical codes from clinical text. In *Proceedings of NAACL-HLT*, pp. 1101–1111, 2018. doi: 10.18653/v1/N18-1100. URL https://aclanthology.org/N18-1100/.

Akshay Smit, Saahil Jain, Pranav Rajpurkar, Anuj Pareek, Andrew Y. Ng, and Matthew P. Lungren. Combining automatic labelers and expert annotations for accurate radiology report labeling using BERT. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, pp. 1500–1519, 2020. doi: 10.18653/v1/2020.emnlp-main.117. URL https://aclanthology.org/2020.emnlp-main.117/.

Jake Snell, Kevin Swersky, and Richard S. Zemel. Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems*, volume 30, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/cb8da6767461f2812ae4290eac7cbc42-Abstract.html.

Betty van Aken, Jens-Michalis Papaioannou, Marcel G. Naik, Georgios Eleftheriadis, Wolfgang Nejdl, Felix A. Gers, and Alexander Löser. This patient looks like that patient: Prototypical networks for interpretable diagnosis prediction from clinical text. In *Proceedings of AACL-IJCNLP (Volume 1: Long Papers)*, pp. 172–184, 2022. doi: 10.18653/v1/2022.aacl-main.14. URL https://aclanthology.org/2022.aacl-main.14/.

Thanh Vu, Dat Quoc Nguyen, and Anthony Nguyen. A label attention model for ICD coding from clinical text. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence*, pp. 3335–3341, 2020. doi: 10.24963/ijcai.2020/461. URL https://www.ijcai.org/proceedings/2020/461.

Gang Wang, Yajun Du, and Yurui Jiang. LSPCL: Label-specific supervised prototype contrastive learning for multi-label text classification. *Knowledge-Based Systems*, 309:112887, 2025. doi: 10.1016/j.knosys.2024.112887. URL https://www.sciencedirect.com/science/article/abs/pii/S0950705124015211.

Guoyin Wang, Chunyuan Li, Wenlin Wang, Yizhe Zhang, Dinghan Shen, Xinyuan Zhang, Ricardo Henao, and Lawrence Carin. Joint embedding of words and labels for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 2321–2331, 2018. doi: 10.18653/v1/P18-1216. URL https://aclanthology.org/P18-1216/.

Liang Wang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder, and Furu Wei. Improving text embeddings with large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 11897–11916, 2024. doi: 10.18653/v1/2024.acl-long.642. URL https://aclanthology.org/2024.acl-long.642/.

Ran Wang, Robert Ridley, Xi'ao Su, Weiguang Qu, and Xinyu Dai. A novel reasoning mechanism for multi-label text classification. *Information Processing & Management*, 58(2):102441, 2021. doi: 10.1016/j.ipm.2020.102441. URL https://www.sciencedirect.com/science/article/abs/pii/S0306457320309341.

Sarah Wiegreffe and Yuval Pinter. Attention is not not explanation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 11–20, 2019. doi: 10.18653/v1/D19-1002. URL https://aclanthology.org/D19-1002/.

Lin Xiao, Xin Huang, Boli Chen, and Liping Jing. Label-specific document representation for multi-label text classification. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 466–475, 2019. doi: 10.18653/v1/D19-1044. URL https://aclanthology.org/D19-1044/.

Ming-Kun Xie, Jia-Hao Xiao, Pei Peng, Gang Niu, Masashi Sugiyama, and Sheng-Jun Huang. Counterfactual reasoning for multi-label image classification via patching-based training. In *Proceedings of ICML*, volume 235, pp. 54576–54589, 2024. URL https://proceedings.mlr.press/v235/xie24i.html.

Ming-Kun Xie, Zhiqiang Kou, Zhongnian Li, Gang Niu, and Masashi Sugiyama. Unlocking the power of co-occurrence in CLIP: A DualPrompt-driven method for training-free zero-shot multi-label classification. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/cb825b4e5d0665a339e8d1aa08d1ca7e-Abstract-Conference.html.

Guangxu Xun, Kishlay Jha, Jianhui Sun, and Aidong Zhang. Correlation networks for extreme multi-label text classification. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 1074–1082, 2020. doi: 10.1145/3394486.3403151. URL https://dl.acm.org/doi/10.1145/3394486.3403151.

Ronghui You, Zihan Zhang, Ziye Wang, Suyang Dai, Hiroshi Mamitsuka, and Shanfeng Zhu. AttentionXML: Label tree-based attention-aware deep model for high-performance extreme multi-label text classification. In *Advances in Neural Information Processing Systems*, volume 32, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/9e6a921fbc428b5638b3986e365d4f21-Abstract.html.

Pingyue Zhang and Mengyue Wu. Multi-label supervised contrastive learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(15):16786–16793, 2024. doi: 10.1609/aaai.v38i15.29619. URL https://ojs.aaai.org/index.php/AAAI/article/view/29619.

Sheng Zhang, Yanbo Xu, Naoto Usuyama, Hanwen Xu, Jaspreet Bagga, Robert Tinn, Sam Preston, Rajesh Rao, Mu Wei, Naveen Valluri, et al. BioMedCLIP: A multimodal biomedical

foundation model pretrained from fifteen million scientific image–text pairs. *arXiv preprint arXiv:2303.00915*, 2023. URL https://arxiv.org/abs/2303.00915.

Xiaoman Zhang, Chaoyi Wu, Ziheng Zhao, Jiayu Lei, Ya Zhang, Yanfeng Wang, and Weidi Xie. RadGenome-Chest CT: A grounded vision–language dataset for chest CT analysis. *arXiv preprint arXiv:2404.16754*, 2024. URL https://arxiv.org/abs/2404.16754.

A Reproducibility Details

This appendix details report preprocessing and the three learning stages of ROOTROUTE: Core initialization and evidence alignment, report-level Core fitting, and verifier training. Checkpoint and revision-budget selection are described separately from output calibration (Appendix B.3). Analytical properties of the score operators are given in Appendix A.6.

A.1 Report preprocessing and frozen representations

Section boundaries are defined by colon-terminated headers at line starts. Windows end at a period, exclamation mark, question mark, newline, or text end; each window is prefixed with its lower-case section name except when the section is unknown. For reports containing more than 64 windows, retention prioritizes Impression/Conclusion, followed by Findings and then other sections; selected windows are restored to their original document order. This selection is independent of routing labels. Frozen BioMedCLIP encodes each retained window with a 256-token input limit into a 512-dimensional unit vector. Concept-assigned Train evidence is used for Core initialization and evidence alignment, whereas label-independent windows are used for report fitting and inference.

A.2 Core initialization and evidence alignment

Representative initialization. For each Root, we deduplicate eligible Train evidence by report group, template, and normalized text, and select candidates with source-aware quotas. Cosine k-medoids selects four representative evidence vectors using multiple seeded restarts; we retain the minimum-cost solution. The normalized mean of the four medoids initializes the Core direction in Section 3.2. The Core reliability logits ρ_r are initialized to 4 before evidence alignment. Evidence alignment uses $T_{\text{align}} = 0.07$; report pooling uses $T_{\text{rep}} = 0.05$.

Evidence sampling and optimization. Alignment uses 418,015 observed-positive evidence rows from 40,393 Train reports, with replacement sampling and 418,015 draws per epoch. An evidence row in Root/source cell (r, s) receives weight proportional to $[\sqrt{n_{rs}} \sum_{s'} \sqrt{n_{rs'}}]^{-1}$, where n_{rs} counts eligible evidence in that cell. The Core alignment objective is 0.5 times the sampled mini-batch mean of Equation 2. AdamW runs for 20 epochs with learning rate 3×10^{-3}, batch size 2,048, weight decay 10^{-4}, and seed 20260807; the last checkpoint is retained.

A.3 Report-level Core fitting

The report-level loss averages concept-weighted binary cross-entropy over sampled reports and all Roots:

$$\ell_{\text{wbce}}(s, y; r) = -w_r y \log \sigma(s/T_\ell) - (1 - y) \log \sigma(-s/T_\ell), \tag{7}$$

where $T_\ell = 0.1$, $M_{\text{pos},r}$ and $M_{\text{unlab},r}$ count observed-positive and unmarked reports for Root r in the corresponding training subset, respectively, and $w_r = \text{clip}[(M_{\text{unlab},r}/M_{\text{pos},r})^{1/2}, 1, 20]$ is fixed from that subset. Reports are sampled with replacement. For a report i from source $s(i)$ containing an observed-positive Root, the unnormalized sampling weight is:

$$\widetilde{w}_i = \sum_{r:\, y_{ir}^{\text{op}}=1} \frac{1}{|\mathcal{R}| \sqrt{n_{r,s(i)}} \sum_{s'} \sqrt{n_{r,s'}}}, \tag{8}$$

where $n_{r,s}$ counts observed-positive reports for Root r and source s in the training subset. When both groups occur, reports with at least one observed-positive label receive half the sampling mass after normalization; all-unmarked reports share the other half uniformly. The default number of draws per epoch equals the number of reports in the subset. Report-level fitting uses AdamW with batch size 128 and a learning-rate search over $\{10^{-4}, 3\times10^{-4}, 10^{-3}\}$. Model-development macro average precision (AP) controls early stopping and selects the checkpoint and training configuration. The selected optimizer learning rate is 10^{-3}, with 10^{-4} for Core directions.

A.4 VERIFIER ARCHITECTURE AND TRAINING

Evidence attention and readout. The target-conditioned attention in Section 3.3 matches each Root direction to up to eight selected windows:

$$\ell_{rj} = \frac{(W_Q \bar{c}_r)^\top (W_K e_j)}{\sqrt{64}} + b^{\mathrm{sec}}_{\mathrm{sec}(j)} + g_{\mathrm{pos}}(\xi_j) + \lambda_{\mathrm{score}} z_{jr}. \tag{9}$$

The section bias b^{sec} has 12 entries, indexed by the window's section $\mathrm{sec}(j)$. Text before the first recognized header uses `preamble`; reports without a recognized header and section names outside the fitted vocabulary use `unknown`. The position adapter g_{pos} receives a two-dimensional vector ξ_j containing the sentence index clipped to $[0, 127]$ and divided by 127, and the retained-window rank clipped to $[0, 63]$ and divided by 63. The learned coefficient λ_{score} starts at 0.25, and invalid slots are masked before softmax. The normalized evidence summary is projected and concatenated with the initial Core score a_r and target-support scalar κ_r before entering the shared readout. The final readout layer is zero-initialized, so verifier training starts with unchanged Core scores.

Component	Configuration
Q/K projections	$512 \to 64$; bias-free
Evidence projection	$512 \to 64$; LayerNorm, then GELU
Position adapter	$2 \to 16 \to 1$; GELU after the hidden layer
Readout	$66 \to 128 \to 64 \to 1$; LayerNorm after the first linear layer, GELU after each hidden layer
Dropout	0.1 after the first readout GELU

Fold-excluded Core states and candidate eligibility. Verifier training uses five report-group folds: each report is scored by a Core whose report-level fit excludes its fold, while Train-wide medoid/alignment initialization and training comparison sets remain shared. Fold-specific Core fits use seed 20260807 and three report-level optimization epochs each. For each target fold, eligibility maps use the other folds' fold-excluded records. Each eligibility map $g_r(a) = \sigma(w_r^{\mathrm{gate}} a + b_r^{\mathrm{gate}})$ is fitted by minimizing summed, unweighted binary cross-entropy with the slope penalty $\frac{10^{-6}}{2}(w_r^{\mathrm{gate}})^2$. The slope and intercept are optimized jointly by Newton updates, with an unconstrained slope and an unpenalized intercept. The ACTIVE cutoff maximizes training F1; the NEAR cutoff targets 0.995 observed-positive recall and is capped at the ACTIVE cutoff. Final eligibility maps use all Train fold-excluded records, not Calibration.

Training objective and sampling. For target r, the fixed comparison set $\mathcal{N}(r)$ contains the four other final Train-fitted Core directions with the highest cosine similarity; equal similarities are ordered by Root index. For eligible candidate i, $\chi_i = 1$ when a comparison concept has a different operational label in the same report. This affects loss weighting and candidate sampling, not verifier inputs. For a batch of B candidates, $s_i = a_{r_i} + \Delta_i$ and $\Delta_i = 0.15 \tanh(v_i)$ during training. Equation 6 uses:

$$\begin{aligned}
\mathcal{L}_{\mathrm{route}} &= \frac{1}{B} \sum_i \ell_{\mathrm{wbce}}(s_i, y_i; r_i), \\
\mathcal{L}_{\mathrm{direction}} &= \frac{\sum_i \chi_i \,\mathrm{softplus}(0.5 - (2y_i - 1)v_i)}{\max(\sum_i \chi_i, 1)}, \\
\mathcal{L}_{\mathrm{sparse}} &= \frac{1}{B} \sum_i \omega_i |\Delta_i|, \\
\mathcal{L}_{\mathrm{bound}} &= \frac{1}{B} \sum_i [\max(|\tanh(v_i)| - 0.95, 0)]^2,
\end{aligned} \tag{10}$$

with $\omega_i = 0.25$ when $\chi_i = 1$ and 1 otherwise, and coefficients $(\lambda_{\mathrm{direction}}, \lambda_{\mathrm{sparse}}, \lambda_{\mathrm{bound}}) = (0.30, 0.02, 0.01)$. Sampling multiplies the weights of positive non-ACTIVE and unmarked ACTIVE candidates by 3, positive ACTIVE candidates by 1.25, and candidates with differing comparison labels by 1.5. A square-root inverse-frequency correction across Roots is then applied and normalized. The verifier uses AdamW with seed 20260807, learning rate 10^{-3}, batch size 512, 200,000 candidate samples per epoch, weight decay 10^{-4}, and gradient clipping 5. Training stops after 20

epochs without a Model-development macro-AP improvement greater than 10^{-5}, with a maximum of 200 epochs.

A.5 Checkpoint and revision-budget selection

The verifier checkpoint maximizes Model-development macro average precision (AP), averaged over Roots containing both classes. With weights fixed, each budget is selected by Root AP from $\{0, 0.025, \ldots, 0.15\}$: choose zero unless a candidate improves on Core by more than 10^{-4}, otherwise choose the smallest value within 10^{-4} of the best AP. Table 3 lists the resulting budgets. Output calibration is fitted separately as specified in Appendix B.3.

Table 3: Root-16 vocabulary and Model-development-selected revision budgets d_r.

Root	d_r	Root	d_r	Root	d_r	Root	d_r
liver	0.050	lung	0.125	gallbladder	0.025	intestine	0.125
spleen	0.000	heart	0.050	adrenal gland	0.025	bladder	0.025
pancreas	0.025	esophagus	0.100	colon	0.150	prostate	0.075
kidney	0.025	stomach	0.150	duodenum	0.075	rectum	0.125

A.6 Analytical properties

Sparse support aggregation. For fixed window compatibilities, Equation 3 is invariant to window order because it uses only the two largest values. Away from ties at the Top-2 selection boundary, its derivative with respect to each unselected compatibility is zero. For an input with at least two windows, adding windows while preserving all existing windows and their scores cannot decrease a_r: neither of the two largest values can decrease, and log-sum-exp is increasing in both arguments. Under these conditions, a_r remains unchanged if no added compatibility exceeds the original second-largest value.

Selective bounded revision. Equation 5 preserves the Core score when $I_r = 0$ and bounds the update by $|s_r - a_r| \leq I_r d_r$. At the common raw-score threshold t_r defined in Section 3.4, a decision change requires the threshold to lie between the initial and revised scores. Consequently,

$$\widehat{y}_r^{\mathrm{Core}} \neq \widehat{y}_r^{\mathrm{Full}} \quad \Longrightarrow \quad I_r = 1 \quad \text{and} \quad |a_r - t_r| \leq |s_r - a_r| \leq d_r.$$

The realized update's direction and magnitude determine whether the threshold is crossed; the target label determines whether the resulting change corrects an error.

B Data and Evaluation Protocol

B.1 Datasets and partitions

Table 4 identifies the report collections, their imaging domains, and their roles in model fitting and evaluation. The five development sources contribute separate Train, Model-development, Calibration, and seen-source test partitions; X-Raydar, IU-Xray, and Stanford are used only for evaluation.

Model-visible text. The model input is RadGenome-ChestCT's region-grounded `Sentence` text on base rows, Merlin's `findings`, MR-RATE's `report`, AbdomenAtlas3.0's `narrative report`, M3D-Cap-public's `response`, X-Raydar's `text`, or IU-Xray's `report`. Two one-character MR-RATE records produce no windows and are excluded from Train. The M3D loader reads native train, validation, and test files in that order and retains the first occurrence of each distinct response text; all 17,493 retained rows consequently have train provenance before group allocation.

Report groups. Before splitting, we group likely duplicate or templated reports using patient identifiers when available, exact normalized-text matching across sources, within-source template matching, and near-duplicate text matching. Only MR-RATE supplies patient identifiers; record identifiers are not grouping keys. Near-duplicate matching normalizes de-identification spans and numbers and requires a token-count ratio of at least 0.75 and character-five-gram Jaccard similarity of at least 0.90.

Table 4: Report sources and partitions. Dashes denote unused roles. The five development sources and X-Raydar/IU-Xray use lexical targets; Stanford uses projected expert annotations. Counts refer to reports with model-visible text.

Source	Domain	Train	Model-dev	Calibration	Test
RadGenome-ChestCT (Zhang et al., 2024)	Chest CT	18,193	2,346	2,375	2,778
Merlin (Blankemeier et al., 2024)	Primarily abdominal CT	13,746	5,059	1,560	5,126
MR-RATE (Forithmus, 2026)	Brain and spine MRI	79,239	3,807	9,149	5,822
AbdomenAtlas3.0 (Bassi et al., 2025)	Abdominal CT	5,309	920	1,088	1,944
M3D-Cap-public (Bai et al., 2024)	Public 3D medical CT	12,140	1,795	1,732	1,826
Development-source total		128,627	13,927	15,904	17,496
X-Raydar (Cid et al., 2024)	Chest radiography	—	—	—	29,756
IU-Xray-REVA-PO (Demner-Fushman et al., 2016)	Chest radiography	—	—	—	2,955
RadGraphXL–Stanford (Delbrouck et al., 2024)	Abdomen/pelvis CT, chest CT, chest radiography, brain MRI	—	—	—	2,000

Partition assignment. Native or external test membership takes precedence; native validation becomes test when no native test exists, and Model-development otherwise. Remaining groups follow Train/Model-development/Calibration allocations of 80/10/10 for RadGenome-ChestCT and Train/Calibration allocations of 90/10 for Merlin and MR-RATE; AbdomenAtlas3.0 and retained M3D-Cap-public groups use 70/10/10/10 for Train/Model-development/Calibration/test. Linked groups move towards the stricter role in the order test, Calibration, Model-development, Train. Groups identified by these rules do not cross partitions. X-Raydar and IU-Xray are entirely source-unseen.

B.2 Operational routing targets

The Root-16 vocabulary is listed in Table 3. A fixed lexical mapper provides training and evaluation targets for the five development sources and the two source-unseen collections. The frozen ontology and extractor are `organ16_v2.0.0` and `organ_semantic_extractor_v2.0.0`, using 66 alias rules over the cleaned input text. Hepatic and splenic flexure map to colon; large bowel maps to colon, small bowel and bowel loops to intestine, and generic bowel remains unmapped. Rectosigmoid maps to both colon and rectum; urinary-bladder rules exclude gall-bladder variants, and ambiguous fundus or antrum mentions require gastric context. For RadGenome-ChestCT, native Anatomy fields are grounding metadata; sentence-level lexical events supply the routing targets. Normal findings, negated disease, uncertainty, and postoperative absence retain the anatomical association; ambiguous or excluded mentions remain unmarked. These are routing targets, not abnormality labels or verified anatomical-presence labels.

Stanford instead projects existing expert Anatomy and organ-identifying Observation spans, including explicit findings and procedures, to Root-16. Compound mentions and anatomical homonyms are resolved using adjacent modifiers and local context; normal, absent, and uncertain assertions retain the association. All 2,000 reports are used, with text reconstructed from native sentence tokens, giving 8,332 positive report–Root pairs. Stanford is evaluation-only, but its annotations also underlie the source-linked RadGraphXL reference. For every collection, observed or mapped positives are encoded as one and all unmarked entries as zero; zero does not denote a verified negative.

B.3 METRICS, CALIBRATION, AND UNCERTAINTY

Metrics and target interpretation. Natural-report macro-F1 averages all 16 Roots, including those without positives; Root F1 is zero when $2TP + FP + FN = 0$. IU-Xray contains observed positives for 11 of the 16 Roots, so its macro-F1 includes five Roots with zero F1 under this definition. Language micro-F1 pools the complete 16-Root prediction vectors. Language micro-F1 pools the complete 16-Root prediction vectors. Exact agreement requires the predicted set to equal the observed or expert-mapped target set. Extra and missing labels are the predicted-minus-target and target-minus-predicted set sizes. All correctness and transition counts use this operational target definition.

Output calibration and thresholds. After model and budget selection, Root-specific Platt maps with positive temperature are fitted on Calibration by unweighted binary cross-entropy. Per-Root thresholds maximize Calibration F1 over distinct probability boundaries, retaining the largest threshold in a tie; predictions use inclusive $\geq$. Core and Full use separately fitted operating points in performance comparisons. Decision-transition analysis instead applies Full's fixed map and threshold to both raw scores, as defined in Section 3.4. External-interface calibration follows Appendix C.

Evaluation-sample uncertainty. Paired 95% percentile-bootstrap intervals use 10,000 resamples with seed 20270815 and fitted models and operating points fixed. Seen-source, X-Raydar, and IU-Xray resample report groups (14,078, 28,157, and 2,195 groups); Stanford resamples individual reports. Language resamples the 2,048 base windows within target Root, keeping all ten transformations of each selected base together. These intervals quantify evaluation-sample uncertainty rather than training-run variability.

C ADDITIONAL EXPERIMENTS AND CONTROLS

The comparison protocols below support Table 1; the Core and verifier controls support Figure 2. Pooling sensitivity and paired uncertainty provide supplementary checks, followed by the fixed-state failure decomposition used at the end of Section 4.4.

C.1 COMPARISON INTERFACES

Matched attention head. The 126,510-parameter classifier uses the same frozen windows, operational labels, weighted report loss, and report/source-balanced sampling as the Core. A $512 \to 231$ Linear–LayerNorm–GELU projection feeds 16 learned queries for label-conditioned attention. For projected windows h_j and query q_r, attention is $\alpha_{rj} = \mathrm{softmax}_j(h_j^\top q_r/\sqrt{231})$. Root-specific affine readouts are combined with a shared $1 \to 34 \to 1$ GELU calibrator through a learned gate and global temperature. The head starts from random initialization and uses AdamW with learning rate 10^{-3}, weight decay 10^{-4}, at most 36 epochs, and patience 8. Training uses seed 20260806; Model-development macro-AP selects the checkpoint, and Calibration fits its operating point.

E5-Mistral. Frozen E5-Mistral embeds the same report windows and retrieves relevant passages using Root-specific queries. Root queries use `Instruct: [instruction] Query: [Root]`, with the instruction "Given an anatomical concept, retrieve radiology-report passages that describe that anatomical region." Each Root's report score is the maximum window cosine similarity; per-Root thresholds maximize F1 on the same Calibration set as ROOTROUTE.

RadGraphXL. The extraction interface uses a 400-token input limit. Root-relevant entities and relations are aggregated into standardized feature vectors for per-Root logistic classifiers fitted on Calibration. Per-Root thresholds maximize Calibration F1. On Stanford, a separate native interface maps extracted Anatomy entities directly to Roots without fitted ranking heads; this is the source-linked reference in Table 1.

LLaMA-3-8B. The interface uses deterministic zero-shot generation with one user message and no system message:

> Given the following full radiology report, identify the anatomical organs discussed in the report. Return only organ names separated by commas. Return NONE if no anatomical organ is discussed. Do not include explanations. Report: [report text]

Inputs retain at most 64 section-aware windows; long reports use fixed head–tail truncation. The parser accepts whole-word canonical Root names in singular or plural form. Empty or unrecognized outputs remain in evaluation as empty predictions.

C.2 Core-learning controls

The Core-only controls in panel (a) of Figure 2 change initialization (unit-normalized Gaussian versus four-medoid mean) and whether evidence alignment is applied. Report-level loss, sampling, windows, and Top-2 pooling remain fixed; each Core receives its own Calibration-fitted operating point. Both unaligned variants initialize reliability logits at 4 and report biases at zero; Model-development selects their checkpoints, while the evidence-aligned condition reuses the primary Core.

C.3 Revision controls

Update rules. These controls fix the primary Core and retrain the verifier. Direct replacement uses $s_r = (1 - I_r)a_r + I_r v_r$, removes the residual scale, and drops the sparsity and bound penalties. Unbounded residual uses $s_r = a_r + I_r d_r v_r$, with training scale 0.15 and all original loss terms, including the saturation penalty on $\tanh(v_r)$; its selected d_r is a multiplier, not a hard bound. Bounded Verify-All sets $I_r = 1$ for all Roots during both training and inference.

Verifier inputs. Current evidence only removes both support scalars; the two single-input controls remove either a_r or κ_r. Their first-layer input widths are 64 and 65 rather than 66, with other widths unchanged. Target-conditioned attention, the Core residual starting score, gate, and bounded update remain in place. All modified verifiers use Model-development macro AP for checkpoint selection and separate Calibration-fitted operating points; applicable residual multipliers follow the same budget-selection rule as Full.

C.4 Pooling sensitivity

For Table 5, Top-1 and Top-4 retrain the complete pipeline at seed 20260807, including fold-specific states, verifier, budgets, and calibration; Top-2 reuses the primary fit.

Table 5: Sparse-pooling sensitivity. Cells report F1 / exact agreement; Language uses micro-F1. Top-2 uses the final Core and Full. Top-1 and Top-4 retrain the complete correction pipeline.

Pooling and stage	Seen-source	X-Raydar	IU-Xray	Stanford	Language
Top-1 Core	0.924/0.775	0.706/0.596	0.392/0.458	0.898/0.604	0.674/0.486
Top-1 Corrected	0.962/0.879	0.780/0.759	0.447/0.662	0.923/0.638	0.682/0.501
Top-2 Core	0.919/0.767	0.690/0.561	0.390/0.408	0.893/0.596	0.664/0.483
Top-2 Full	0.963/0.876	0.780/0.763	0.476/0.807	0.928/0.683	0.675/0.492
Top-4 Core	0.914/0.759	0.667/0.588	0.397/0.428	0.891/0.613	0.644/0.469
Top-4 Corrected	0.963/0.878	0.767/0.742	0.492/0.692	0.930/0.652	0.684/0.507

C.5 Paired Core-to-Full uncertainty

Table 6 reports paired differences for the fixed primary models at their separately fitted operating points. The intervals use the resampling units and protocol in Appendix B.3, with the fitted models and operating points held fixed.

Table 6: **Paired Core-to-Full performance differences.** Full minus Core using aligned fixed predictions; each stage uses its own Calibration-fitted operating point. Entries report differences with unadjusted 95% percentile-bootstrap intervals. F1 is macro-averaged for natural reports and micro-averaged for Language.

Evaluation	ΔF1 [95% CI]	ΔExact [95% CI]
Seen-source	+0.044 [0.041, 0.047]	+0.110 [0.102, 0.118]
X-Raydar	+0.090 [0.071, 0.110]	+0.202 [0.192, 0.214]
IU-Xray	+0.085 [0.040, 0.161]	+0.399 [0.353, 0.445]
Stanford	+0.035 [0.027, 0.043]	+0.088 [0.069, 0.107]
Language	+0.011 [0.005, 0.016]	+0.009 [0.002, 0.015]

C.6 Revision-failure decomposition

Scope and denominators. This analysis uses saved Core and Full scores with Full's frozen calibration and thresholds; candidate gates and revision budgets remain fixed. Initial errors include both false negatives and false positives under the operational evaluation targets. Table 7 partitions only these initial Core errors, not all errors made by Full. New errors introduced from initially correct Core decisions are counted separately in Table 2.

Relation to the decision ledger. The outcome percentages in Table 7 use initial Core errors as their denominator. Table 2 reports transition counts and a flip rate over all report–Root decisions; the fraction of changes that correct an error instead uses all changed decisions as its denominator.

Eligibility and permitted range. For each initial Core error, gate closure is checked first. For an eligible false negative, the permitted interval contains the correct positive decision when $a_r + d_r \geq t_r$. For an eligible false positive, it contains a correct negative decision when $a_r - d_r < t_r$. The strict inequality in the latter condition follows from the inclusive positive decision rule. Eligible errors failing the relevant condition are budget-blocked, including those with zero budget. The remaining errors are divided into corrected and within-range uncorrected cases according to Full's prediction. The four categories are mutually exclusive and exhaustive.

Table 7: Disposition of initial Core errors at Full's fixed operating point. The four outcome columns partition these errors and report percentages of the Core-error count. Newly introduced errors are excluded; they are reported separately in Table 2.

Evaluation	Core errors	Gate closed (%)	Budget blocked (%)	Within range, uncorrected (%)	Corrected (%)
Seen-source	7,258	2.4	5.8	14.1	77.7
X-Raydar	12,330	2.9	16.3	27.6	53.2
IU-Xray	1,592	0.5	4.0	22.0	73.5
Stanford	1,088	5.0	20.3	25.8	48.9
Language	11,693	31.1	16.4	28.5	24.0

Candidate coverage and unrecovered errors. Across the four natural-report evaluations, the gate admits 16.0–42.7% of all report–Root decisions while covering 95.0–99.5% of initial Core errors. Among the initial errors that remain uncorrected, 50.5–83.2% are eligible and have a permitted interval containing the correct decision. These percentages use unrecovered initial errors as their denominator, whereas the outcome percentages in Table 7 use all initial Core errors.

Direction of the realized update. For a false negative, $\Delta_r > 0$ points towards the correct decision; for a false positive, $\Delta_r < 0$ does so. Among eligible, within-range but uncorrected errors in the four natural-report evaluations, 82.2–87.2% receive updates directed away from the correct decision. The remaining updates have the correct direction but insufficient magnitude to change the prediction. On Language, 31.1% of initial errors are excluded by the gate, and only 45.7% of eligible errors within the permitted range are corrected. Newly introduced errors also offset most corrections at this common operating point, as recorded in Table 2.

Interpretation. Range permission is a geometric condition, not a guarantee that the verifier produces the required update. The analysis separates errors blocked by the fixed revision rules from ineffective realized updates, but does not estimate how changing the gate, changing the budgets, or retraining the verifier would affect performance.

D CONTROLLED DIAGNOSTICS

D.1 LANGUAGE VARIATIONS

Construction. The suite selects 128 held-out seen-source windows per Root, with one base per report and exactly one detected target per base. Each of the 2,048 bases produces an unchanged control and nine fixed Root-specific transformations, yielding 20,480 probes (Table 8). Negated and postoperative expressions retain the anatomy-discussed target; they are not abnormality or physical-presence labels.

Condition-wise evaluation. The table reports the primary Full fit. Recall requires the target, Exact requires it alone, and Extra counts other predicted Roots. Overall micro-F1 pools all 16-Root vectors across conditions, rather than averaging condition-wise F1. Section 4.4 analyzes these Full-system errors; the results do not isolate either stage.

Table 8: **Controlled-language construction and performance.** Primary-fit Full results with 2,048 probes per condition. Recall concerns the designated target; Exact requires that target and no others; Extra is the mean number of additional Roots. All ten conditions contribute to Overall (20,480 probes).

Category	Construction	Recall	Exact	Micro-F1	Extra
Unchanged control	Unchanged source window.	0.839	0.683	0.819	0.211
Synonym/ adjectival	Longest target match replaced by a synonym/adjective.	0.766	0.584	0.741	0.300
Spelling/ morphology	Fixed spelling or morphological variant.	0.841	0.667	0.806	0.245
Cross sentence	Professional term followed by a cross-sentence reference.	0.643	0.604	0.753	0.065
Professional terminology	Fixed professional term.	0.572	0.452	0.638	0.221
Implicit/descriptive	Implicit clinical description.	0.440	0.340	0.544	0.177
Negated pathology	No focal abnormality in the target anatomy.	0.938	0.938	0.968	0.000
Postoperative/ absent	Prior Root-specific surgery; postoperative bed included in examination.	0.250	0.188	0.381	0.063
Abbreviation	Fixed abbreviation.	0.270	0.234	0.402	0.072
Typo/misspelling	Fixed typographic variant.	0.260	0.229	0.390	0.074
Overall	All ten categories pooled.	0.582	0.492	0.675	0.143

D.2 SUPPORT-FORM CONSTRUCTION AND OUTCOME SCORING

Pairs and labels. Two held-out templates for each of 120 unordered Root pairs instantiate neither target, either target alone, and both in separate windows, yielding 960 views. Labels include every explicitly named Root. Neither-target inputs mention another anatomical concept, which retains its positive label while the tested pair is labeled zero.

Encoding and operating point. The two texts in both-target views are encoded independently, not concatenated into one window. Both use the unknown-section identifier and positional indices $(0, 1)$. Core and Full share Full's frozen calibration and thresholds.

Tested-pair scoring. Accuracy requires both tested entries to match their labels: single-target success accepts one member and rejects its partner; neither-target success rejects both. Single-target

conditions are pooled for reporting. The other 14 predictions are excluded, unlike complete-set agreement.

Scope. There is no otherwise matched two-target, joint-window condition. The diagnostic therefore establishes subset recovery in these constructions, not that window separation causes Core errors or that all 16 decisions are correct.