R2QCIF: Behavioral-Trace-as-Supervision for Radiology Report Quality Control

Anonymous ACL submission

Abstract

Errors in radiology reports can propagate through clinical decisions and lead to missed diagnoses, misdiagnoses, inappropriate treatment, and patient-safety risks. Although large language models (LLMs) offer new opportunities for automated quality control, medical AI faces a supervision granularity gap: fine-grained training labels are scarce, whereas hospital information systems accumulate abundant clinical revision traces that implicitly externalize physicians' correction knowledge. To bridge this gap, we propose **R2QCIF**, a behavioral-trace-as-supervision framework that treats draft-to-final physician revisions as naturally occurring implicit annotations and converts them into usable fine-grained supervision through iterative self-training. As the core of R2QCIF, **IF2** verifies whether model-generated pseudo-labels are supported by revision-derived evidence. Experiments show that R2QCIF outperforms conventional self-training, approaches manual-annotation-based supervised training, and transfers to editing datasets with implicit-annotation structures, validating its effectiveness in bridging the supervision granularity gap.

1 Introduction

Errors in radiology reports are not uncommon (Renfrew et al., 1992; Fitzgerald, 2001; Brady, 2017), ranging from textual and speech transcription errors (Prevedello et al., 2014; Motyer et al., 2016) to contextual inconsistencies and incorrect medical relationships. Such errors may propagate through downstream clinical decision-making, leading to missed diagnoses, misdiagnoses, inappropriate treatment, delayed follow-up and patient-safety risks. Therefore, radiology report quality control is an important clinical safeguard in the diagnostic workflow, but current manual review places a heavy burden on physicians. Recent LLMs have shown potential for assisting radiologists in report generation and error detection (Gertz et al., 2024), enabling fine-grained decision support such as error localization, classification, and rationale generation. Nevertheless, online LLM services may introduce risks of clinical data leakage and patient privacy disclosure (Wang et al., 2024). As a result, lightweight and locally deployable LLMs have become a more practical direction for radiology report quality control (Salam et al., 2025; Moll et al., 2025; Nowak et al., 2025).

The central obstacle is not the absence of medical expertise, but its usable form. LLM training requires explicit fine-grained labels, such as error-location annotations, error categories, and rationales, which are costly and difficult to obtain at scale. In contrast, hospital information systems naturally accumulate abundant clinical behavioral traces, including report revision histories and version differences. These traces record physicians' correction behavior and externalize medical expertise, but are often stored and ignored as operational logs rather than used as supervision signals. We define this mismatch as a supervision granularity gap: LLM training requires explicit fine-grained supervision, whereas clinical workflows produce implicit, revision-level, and behavioral supervision.

We focus on draft-to-final radiology report revisions as a representative behavioral trace. A physician's draft-to-final revision can be regarded as a naturally occurring implicit annotation, signaling that part of the draft required correction and that the final report reflects a clinically preferred state. However, it does not directly specify which sentence is erroneous, what type of error it contains, or why the content is clinically inappropriate. Manually converting such revisions into explicit labels is possible, but it defeats the purpose of using naturally accumulated traces and reintroduces high annotation costs. Teacher–student distillation may assist annotation (Xia et al., 2025), but closed-source online teachers raise privacy and compli-

ance concerns for clinical texts (Zhong et al., 2025). Conventional self-training can reduce annotation cost by expanding pseudo-label supervision, but it mainly relies on the model's own confidence or consistency to generate and select pseudo labels. Such model-internal signals may not reflect medical correctness and, more importantly, do not exploit the externalized medical knowledge embedded in physician revision traces.

To bridge the supervision granularity gap, we propose **R2QCIF** (**R**evision-**t**o-**Q**uality-**C**ontrol via **I**mplicit-to-**F**ine-grained Formalization), a behavioral-trace-as-supervision framework that converts physician revision traces into fine-grained pseudo-label supervision through revision-aware self-training. As shown in Figure 2, its core mechanism, **IF2** (**I**mplicit-to-**F**ine-grained **F**ormalization) aligns candidate pseudo labels with revision-derived evidence, using edit-location alignment, confidence estimation, and counterfactual state verification to verify whether model predictions are supported by the observed revision trace. Our contributions are summarized as follows: 1) We introduce a behavioral-trace-as-supervision paradigm and define the supervision granularity gap between scarce fine-grained labels and abundant revision traces that serve as implicit annotations. 2) We propose R2QCIF, a revision-aware iterative self-training framework, and design IF2 as its key mechanism to convert revision-derived implicit annotations into reliable fine-grained pseudo-label supervision. 3) We construct Reportif, a synthetic dataset containing report revisions, and evaluate R2QCIF on Reportif, MEDEC-MS (Ben Abacha et al., 2025), and IteraTeR (Kim et al., 2022), validating its effectiveness and transferability across implicit-annotation scenarios.

2 Related Work

Existing radiology NLP studies mainly focus on automatic report generation from images (Bu et al., 2024; Chen et al., 2025; Gu et al., 2025), report evaluation metrics (Huang et al., 2024; Ostmeier et al., 2024), and information extraction from reports (Le Guellec et al., 2024; Builtjes et al., 2025). These studies help reduce radiologists' workload, but they do not eliminate the need for physician review, since radiologists remain responsible for the clinical correctness and safety of final reports. Therefore, practical radiology applications require not only report generation and evaluation, but also report proofreading and error detection.

Research on radiology report proofreading remains relatively limited. Recent efforts, such as MEDIQA-CORR 2024 (Ben Abacha et al., 2025) and radiology report error correction (Wu et al., 2025), provide useful benchmarks and methods for clinical text correction. However, these approaches typically rely on gold-standard annotations or expert-maintained resources, which are costly to obtain and difficult to scale in real clinical environments. In contrast, our work focuses on naturally accumulated report revisions, treating physician correction behavior as an implicit supervision source rather than relying solely on explicit labels.

Semi-supervised learning and iterative self-training have been widely explored to reduce annotation dependence in LLM training. Existing self-training methods (Dong et al., 2024; Ye et al., 2024) and SSRM (He et al., 2024) expand supervision by generating pseudo labels from limited labeled data. However, these methods mainly rely on model-internal signals, such as confidence or consistency, to select pseudo labels, and they usually treat unlabeled samples independently. In revision-based report quality control, the available weak supervision is instead expressed as paired report versions before and after physician revision. Therefore, the central challenge is not merely how to generate pseudo labels, but how to determine whether a candidate fine-grained label is supported by the externalized clinical knowledge embedded in the observed revision.

3 Reportif: Synthetic Revision Dataset

3.1 Dataset Construction

To study the supervision granularity gap between revision-level behavioral traces and fine-grained quality-control labels in a controllable setting, we construct **Reportif**, a synthetic revision dataset derived from MIMIC-CXR (Johnson et al., 2019). We select 3,800 reports with complete Findings and Impression sections, inject synthetic errors into 1,900 reports using a locally deployed DeepSeek-V3 (Liu et al., 2024) model, and retain the remaining 1,900 reports as error-free controls. This construction simulates a draft-to-final revision process: the corrupted report serves as the pre-revision draft, while the original report serves as the post-revision quality-controlled version. The injected samples are also associated with fine-grained reference an-

notations, including the erroneous sentence, error category, and possible explanation, which support evaluation of error localization, classification, and reasoning.

We define seven error categories: typographical error, ASR-like homophone error, template residue, diagnostic presence/absence error, severity misjudgment, recommendation–finding mismatch, and lesion information conflict. These categories are derived from physician summaries at collaborating hospitals and related work on synthetic data generation. Typographical and ASR-like homophone errors are rule-based, while the remaining types are generated by the model with category-specific prompts. All synthesized samples are manually reviewed and corrected for clinical plausibility, label correctness, and consistency with radiology reporting conventions.

Due to the data use agreement of MIMIC-CXR, we do not directly release the synthesized dataset. Instead, we provide the error-injection rules, prompt templates, and construction details in the Appendix A to support reproducibility for authorized MIMIC-CXR users.

3.2 Task Difficulty Analysis

Following PhysioNet's guidance on the use of MIMIC data with LLMs and online services, we evaluate several locally deployed open-source LLMs on the Reportif test set to examine whether it provides a non-trivial fine-grained quality-control benchmark, as shown in Table 1.

Model	Pos_acc	Type_acc	Rea_bleurt	All_acc
DeepSeek-V3.2	26.58	27.63	46.21	19.47
DeepSeek-V4-Flash	38.16	35.00	46.61	29.74
QwQ-32B	30.79	27.11	43.48	22.37
Qwen3.6-35B-A3B	40.11	39.31	44.75	32.98

Table 1: Task difficulty validation on the Reportif test set. All LLMs are evaluated with the same prompt.

As shown in Figure 1, we further analyze the performance of DeepSeek-V3.2 (Liu et al., 2025), DeepSeek-V4-Flash (DeepSeek-AI, 2026), QwQ-32B (Qwen Team, 2025; Yang et al., 2024) and Qwen3.6-35B-A3B (Qwen Team, 2026) across error categories. All these models were deployed locally to avoid transmitting protected data to external online services. General-purpose LLMs achieve relatively better results on surface-level errors, such as typographical errors, ASR-like homophone errors, and template residues, but perform poorly on context-dependent errors requiring cross-sentence reasoning. This suggests that Reportif is not a trivial synthetic benchmark and can evaluate fine-grained quality-control capabilities beyond surface error detection.

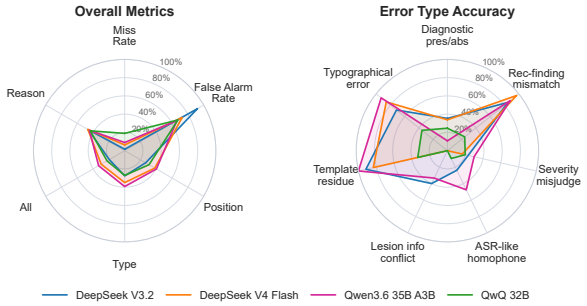

Figure 1: General-purpose LLM performance on Reportif. The left panel shows overall metrics, and the right panel shows category-level accuracy.

4 Methodology

R2QCIF is built on the assumption that physician revisions are behavioral traces of clinical correction decisions. A draft-to-final report pair provides an implicit annotation: the draft report reflects a potentially problematic state, while the final report reflects a clinically preferred state after physician revision. However, such behavioral traces are not directly usable as model supervision because they do not explicitly specify the error location, error type, or rationale. R2QCIF bridges this supervision granularity gap by generating candidate fine-grained pseudo labels from pre-revision reports and using IF2 to verify whether they are supported by the observed revision trace.

4.1 Task Definition

The goal of R2QCIF is to recover explicit fine-grained quality-control supervision from revision-level behavioral traces. In radiology report quality control, given a report X, the model $\mathcal{M}$ is expected to produce a fine-grained quality-control result E. We denote the report as $X = (s_1, s_2, \ldots, s_j)$, where s_j is the j-th sentence. Following our experimental setting, each instance contains at most one target error, and the error location is defined at the sentence level. The output is formulated as $E = (P, T, R)$, where $P = s_k$ denotes the erroneous sentence, T denotes the error type from the predefined categories, and R denotes the rationale supporting the quality-control decision.

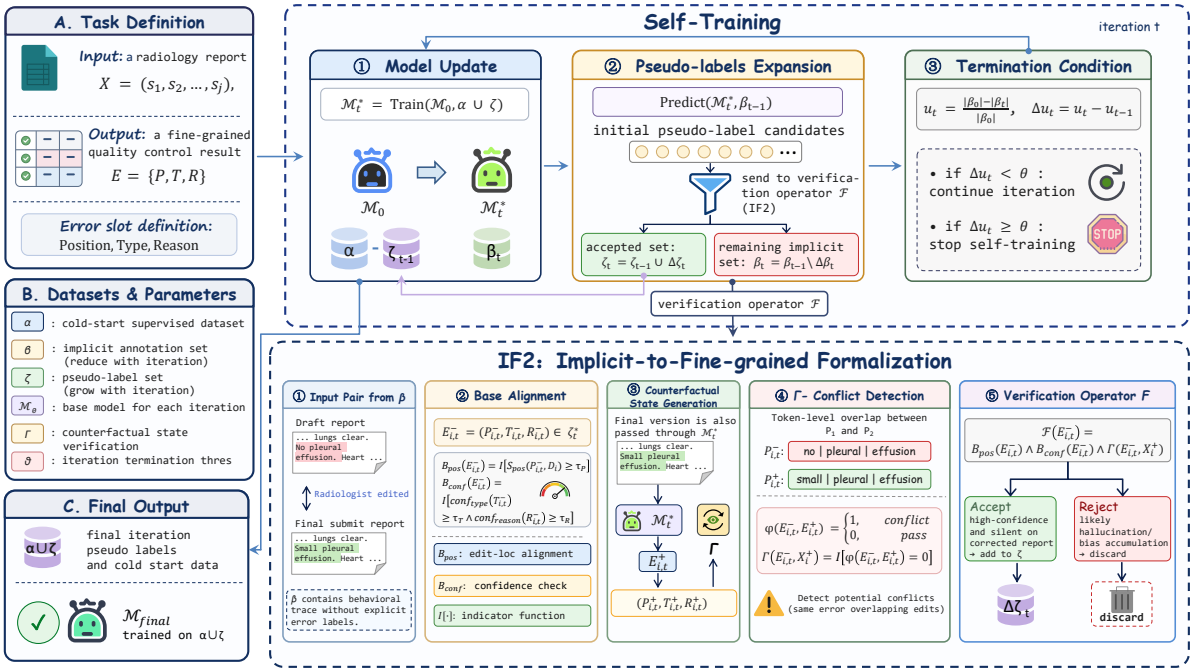

Figure 2: Overview of R2QCIF. R2QCIF follows the behavioral-trace-as-supervision paradigm by treating draft-to-final physician revisions as implicit annotations and converting them into fine-grained pseudo-label supervision through revision-aware self-training. Its key mechanism, IF2, aligns candidate pseudo labels with revision-derived evidence through edit-location alignment, confidence estimation, and counterfactual state verification, so that only pseudo labels supported by physician revision traces are added to the training set.

4.2 Revision-Aware Self-Training Pipeline

We define the revision-derived behavioral trace set as β, where $\beta = \{(X_i^-, X_i^+)\}_{i=1}^N$. X_i^- denotes the pre-revision draft report, and X_i^+ denotes the post-revision quality-controlled report. Each pair represents an implicit annotation produced by physician revision behavior, but it does not provide explicit fine-grained labels. In addition, we use a small supervised dataset with explicit quality-control annotations as the cold-start dataset, denoted by α. The pseudo-label set verified by IF2 is denoted by ζ.

At the t-th iteration, R2QCIF updates $\mathcal{M}_t^*$ using α and the revision-supported pseudo-label set ζ_{t-1}. The superscript $*$ denotes a candidate temporary state, and the subscript t denotes the t-th iteration. The updated model then generates candidate pseudo labels on the remaining behavioral trace set β_{t-1}. IF2 verifies whether these candidates are supported by the corresponding revision traces, resulting in an expanded pseudo-label set ζ_t and an updated unused trace set β_t.

Model Update At the t-th iteration, the model is initialized from $\mathcal{M}_0$, the untrained base model, and fine-tuned on α and ζ_{t-1}, where $\zeta_0 = \varnothing$. This re-initialization reduces the accumulation of iteration-specific pseudo-label noise:

$$\mathcal{M}_t^* = \text{Train}(\mathcal{M}_0, \alpha \cup \zeta_{t-1}) \tag{1}$$

Pseudo-Label Expansion The updated model is used to make predictions on the pre-revision reports in β_{t-1}. The unverified pseudo-label set generated at the t-th iteration is defined as:

$$\zeta_t^* = \text{Predict}(\mathcal{M}_t^*, \{X_i^- \mid (X_i^-, X_i^+) \in \beta_{t-1}\}) \tag{2}$$

The newly added pseudo-label set $\Delta\zeta_t$ consists of the candidates in ζ_t^* that pass IF2 verification. The formal definition of the verification criterion is introduced in the IF2 section.

Thus, the pseudo-label set obtained at the t-th iteration is:

$$\zeta_t = \zeta_{t-1} \cup \Delta\zeta_t \tag{3}$$

We define the behavioral trace data used at the t-th iteration as $\Delta\beta_t = \text{Source}(\Delta\zeta_t)$, where $\text{Source}(\Delta\zeta_t)$ denotes the set of revision pairs corresponding to the pseudo labels that pass IF2 verification at iteration t. Accordingly, the remaining unused behavioral traces are:

$$\beta_t = \beta_{t-1} \setminus \Delta\beta_t \tag{4}$$

Iteration Termination Let $|\cdot|$ denote the cardinality of a set. We define the utilization rate of behavioral traces as:

$$u_t = \frac{|\beta_0| - |\beta_t|}{|\beta_0|} \tag{5}$$

β_0 denotes the initial behavioral trace set. The utilization gain is defined as $\Delta u_t = u_t - u_{t-1}$. Given an iteration termination threshold θ, the iterative process continues when $\Delta u_t \geq \theta$. When $\Delta u_t < \theta$, the model is considered to have reached the boundary of efficiently utilizing revision-derived behavioral traces, and self-training is terminated. The final model $\mathcal{M}_{final}$ is then retrained using the final pseudo-label set. Assuming that the process terminates at the k-th iteration, we have:

$$\mathcal{M}_{final} = \mathrm{Train}\big(\mathcal{M}_0,\, \alpha \cup \zeta_k\big) \tag{6}$$

4.3 IF2: Aligning Revision Evidence with Pseudo Labels

IF2 is designed to convert revision-derived implicit annotations into reliable fine-grained pseudo-label supervision. Instead of accepting pseudo labels based only on model-internal confidence, IF2 asks whether a candidate label is supported by the physician's observed revision behavior. Specifically, a reliable pseudo label should satisfy three conditions: it should align with the edited region, be sufficiently stable under confidence estimation, and disappear in the revised counterfactual state. We implement these conditions with the verification operator $\mathcal{F}$.

We denote an arbitrary candidate pseudo label in the candidate quality-control results ζ_t^* generated by the model at iteration t as:

$$E_{i,t}^- = \left(P_{i,t}^-, T_{i,t}^-, R_{i,t}^-\right) \in \zeta_t^* \tag{7}$$

$P_{i,t}^-$ denotes the predicted error location, $T_{i,t}^-$ denotes the predicted error type, and $R_{i,t}^-$ denotes the predicted rationale. The superscript $-$ indicates that the pseudo label is derived from the pre-revision report.

4.3.1 Edit-Location Alignment

Physician revisions provide direct behavioral evidence about where the report was changed. Therefore, a candidate pseudo label is more likely to be revision-supported if its predicted error location overlaps with the physician-edited region.

Let D_i denote the edit-location difference in the i-th revision pair, and let s_k denote the actually modified sentence. We extract the physician-edited location by comparing the report before and after revision:

$$D_i = \mathrm{Diff}\left(X_i^-, X_i^+\right) = s_k \in X_i^- \tag{8}$$

$\mathrm{Diff}(\cdot)$ denotes a difference function used to compare textual differences. For a candidate pseudo label $E_{i,t}^-$, we compute the token-level overlap between its predicted position $P_{i,t}^-$ and the physician-edited position s_k:

$$S_{\mathrm{pos}}\left(P_{i,t}^-, D_i\right) = \mathrm{IoU}_{\mathrm{tok}}\left(P_{i,t}^-, s_k\right) \tag{9}$$

The token-level IoU is defined as:

$$\mathrm{IoU}_{\mathrm{tok}}(A, B) = \frac{|\mathrm{token}(A) \cap \mathrm{token}(B)|}{|\mathrm{token}(A) \cup \mathrm{token}(B)|} \tag{10}$$

Here, $\mathrm{token}(\cdot)$ denotes a tokenization function that converts an input text span into a set of tokens.

Let τ_p denote the similarity threshold for edit locations. The edit-location alignment function is defined as:

$$B_{\mathrm{pos}}\left(E_{i,t}^-\right) = \mathbb{I}\left[S_{\mathrm{pos}}\left(P_{i,t}^-, D_i\right) \geq \tau_p\right] \tag{11}$$

Here, $\mathbb{I}[\cdot]$ denotes the indicator function, which takes the value 1 when the condition inside the brackets holds and 0 otherwise.

4.3.2 Confidence-Based Stability Check

Confidence scores are used to remove clearly unstable pseudo labels, rather than to determine medical correctness. For a candidate pseudo label $E_{i,t}^-$, we compute confidence scores for the error type and rationale from the token probabilities of the generated outputs, denoted as $\mathrm{Conf}_{\mathrm{type}}\left(T_{i,t}^-\right)$ and $\mathrm{Conf}_{\mathrm{reason}}\left(R_{i,t}^-\right)$, respectively. Specifically, for each generated token y_l, we compute the softmax probability assigned to the actually generated token:

$$p_l = p(y_l \mid X_i^-, y_{<l}) = \mathrm{softmax}(\mathbf{z}_l)_{y_l} \tag{12}$$

where $\mathbf{z}_l$ denotes the logits at the l-th decoding step.

For the error type and rationale, their confidence scores are computed as the mean probabilities of the corresponding generated tokens. Let τ_T and τ_R denote the confidence thresholds for the error type and rationale. The confidence-based stability function is defined as:

$$B_{\mathrm{conf}}(E_{i,t}^-) = \mathbb{I}\big[\mathrm{Conf}_{\mathrm{type}}(T_{i,t}^-) \geq \tau_T \wedge \mathrm{Conf}_{\mathrm{reason}}(R_{i,t}^-) \geq \tau_R\big] \tag{13}$$

4.3.3 Counterfactual State Verification

Counterfactual state verification uses the post-revision report as a behavioral reference state. If a candidate pseudo label correctly captures an error corrected by the physician, then the same error should no longer be predicted at the corresponding location in X_i^+. Otherwise, the candidate is more likely to reflect a model-side hallucination or systematic bias rather than revision-supported medical knowledge.

Specifically, we use the same model $\mathcal{M}_t^*$ from the current iteration to make a prediction on the revised report X_i^+, obtaining the counterfactual quality-control result:

$$\begin{aligned} E_{i,t}^+ &= \text{Predict}(\mathcal{M}_t^*, X_i^+) \\ &= (P_{i,t}^+, T_{i,t}^+, R_{i,t}^+). \end{aligned} \tag{14}$$

We define the conflict function between a candidate pseudo label $E_{i,t}^-$ and a counterfactual prediction $E_{i,t}^+$ as:

$$\phi(E_{i,t}^-, E_{i,t}^+) = \begin{cases} 1, & \begin{array}{l} T_{i,t}^- = T_{i,t}^+ \wedge \\ \text{IoU}_{\text{tok}}\left(P_{i,t}^-, P_{i,t}^+\right) > \gamma \end{array} \\ 0, & \text{otherwise} \end{cases} \tag{15}$$

Here, γ is the conflict detection threshold. When two predictions have the same error type and the token-level IoU between their predicted locations exceeds γ, they are considered to be in conflict.

Based on the conflict function, the counterfactual state verification function is defined as:

$$\Gamma\left(E_{i,t}^-, X_i^+\right) = \mathbb{I}\left[\phi\left(E_{i,t}^-, E_{i,t}^+\right) = 0\right] \tag{16}$$

When $\Gamma(E_{i,t}^-, X_i^+) = 1$, the model does not predict an error conflicting with the candidate pseudo label in the revised report, and thus the candidate pseudo label passes counterfactual state verification. When $\Gamma(E_{i,t}^-, X_i^+) = 0$, the candidate pseudo label conflicts with the prediction on the revised report and should therefore be rejected.

4.3.4 IF2 Verification Operator

By integrating edit-location alignment, confidence-based stability checking, and counterfactual state verification, IF2 verifies whether a candidate pseudo label is supported by revision-derived implicit supervision. The verification result for a single candidate pseudo label $E_{i,t}^-$ is defined as:

$$\begin{aligned} \mathcal{F}(E_{i,t}^-) = B_{\text{pos}}(E_{i,t}^-) \wedge B_{\text{conf}}(E_{i,t}^-) \\ \wedge\ \Gamma(E_{i,t}^-, X_i^+) \end{aligned} \tag{17}$$

Therefore, the set of newly added pseudo labels that pass IF2 verification at the t-th iteration is:

$$\Delta\zeta_t = \left\{E_{i,t}^- \in \zeta_t^* \mid \mathcal{F}\left(E_{i,t}^-\right) = 1\right\} \tag{18}$$

5 Experiments

5.1 Experimental Setup

To evaluate R2QCIF under resource-constrained settings, we fine-tune 8B-scale models with LoRA in each iteration. We use a data gain threshold of $\theta = 0.01$ and a maximum of 10 self-training iterations. Pseudo-label generation uses temperature 0.7 and top_p 0.8, while final evaluation uses greedy decoding with `do_sample=False`. To reduce model-specific effects, we evaluate both Llama3.1-8B-Instruct (Grattafiori et al., 2024) and Qwen3-8B (Yang et al., 2025). Additional training details are provided in Appendix B.

5.1.1 Evaluation Metrics

We evaluate model performance using structured accuracy and semantic similarity. Structured metrics include position accuracy, type accuracy, and overall accuracy, which measure whether the model correctly predicts the erroneous sentence, error category, and complete structured output, respectively. For free-text rationales or corrections, we use BLEURT-20 (Sellam et al., 2020), multiplied by 100 for readability.

5.1.2 Datasets

We evaluate R2QCIF on Reportif, MEDEC-MS, and IteraTeR. For each dataset, we split the data into a cold-start set α, an implicit annotation set β, and a test set. Although reference annotations are available, labels in β are hidden during R2QCIF training; only paired versions are used for IF2 verification, simulating the supervision granularity gap between revision traces and fine-grained labels. Dataset sizes and error-sample proportions (F_ratio) are shown in Table 2:

Dataset	α set	β set	Test Data	F_ratio
Reportif	380	3040	380	50%
MEDEC-MS	574	2189	597	52.09%
IteraTeR Dataset	442	3745	428	60%

Table 2: Dataset statistics for splits.

MEDEC-MS provides an external medical text correction setting beyond Reportif. Since it does not contain correction-rationale annotations, we

Model	Method	Reportif				MEDEC-MS			
		Pos_acc	Type_acc	Rea_bleurt	All_acc	Pos_acc	Type_acc	Cor_bleurt	All_acc
Llama3.1-8B-Instruct	Base	12.63	38.95	41.84	6.32	26.30	23.79	40.37	8.21
	Cold Start	63.16	63.68	52.56	56.05	51.93	**71.38**	62.46	40.20
	FullSFT†	91.58	93.16	61.26	90.79	55.61	64.95	64.69	44.56
	SSRM	86.84	86.05	55.86	82.89	57.79	63.34	61.70	46.23
	LMSI	66.05	66.58	**57.53**	62.89	50.25	38.20	62.99	37.69
	R2QCIF	**89.74**	**89.21**	57.17	**86.05**	**59.30**	68.81	**63.01**	**46.57**
Qwen3-8B	Base	26.84	38.42	47.14	11.58	43.05	34.17	52.07	30.65
	Cold Start	54.21	51.58	52.01	43.16	53.10	45.90	63.35	45.06
	FullSFT†	91.05	91.84	60.62	89.47	59.46	61.09	64.83	49.58
	SSRM	86.05	84.74	55.94	81.58	58.29	**56.91**	64.08	48.41
	LMSI	60.79	60.53	**59.47**	60.26	51.76	40.54	64.61	39.87
	R2QCIF	**90.26**	**86.58**	55.90	**84.74**	**62.14**	51.09	**65.81**	**50.59**

Table 3: Quality-control performance on medical datasets. Bold indicates the best label-scarce result. FullSFT† uses gold labels from β as a fully supervised reference, where the supervision granularity gap is explicitly resolved.

Model	Method	IteraTeR Dataset			
		Edit Pos_acc	Intent_acc	Edit_bleurt	All_acc
Llama3.1-8B-Instruct	Base	7.48	0	15.41	0
	Cold Start	23.60	38.55	11.83	19.39
	FullSFT†	43.69	59.11	14.72	40.19
	SSRM	39.95	44.36	7.56	35.75
	LMSI	37.85	42.76	13.53	36.21
	R2QCIF	**55.14**	**57.94**	**17.69**	**45.09**
Qwen3-8B	Base	10.75	0.47	15.59	0.47
	Cold Start	26.64	44.86	8.49	25.47
	FullSFT†	41.36	56.31	8.71	38.79
	SSRM	44.56	38.52	13.67	38.32
	LMSI	38.55	46.50	14.12	32.94
	R2QCIF	**59.81**	**57.48**	**19.39**	**48.13**

Table 4: Results on the IteraTeR edit-intention dataset.

replace the rationale-based metric with an error-sentence correction metric adapted to MEDEC-MS, allowing us to evaluate whether revision-derived supervision can support medical correction beyond our synthetic dataset. IteraTeR is an edit-intent recognition dataset, which we use as an extended setting to examine whether the proposed revision-to-supervision framework transfers beyond medical quality control. We use its human-annotated document-level data, and split samples with multiple edits into separate instances.

For MEDEC-MS and IteraTeR, we do not use the complete datasets due to practical constraints. Instead, we use only the currently fully open-source portions.

5.1.3 Baselines

We compare R2QCIF with three training settings that reflect different levels of supervision. Base uses the original model without task-specific training, Cold Start trains only on the explicitly labeled set α, and FullSFT trains on gold labels from both α and β. FullSFT is not label-scarce, it serves as a fully supervised reference where the supervision granularity gap is manually resolved. All settings use the same prompts, temperature, and top_p. In Table 3, they are denoted as Base, Cold Start, and FullSFT.

For self-training baselines, we compare with SSRM and LMSI (Huang et al., 2023). SSRM is a non-cumulative self-training method that predicts pseudo labels with the model from the previous iteration and retrains on newly generated pseudo labels. LMSI samples pseudo labels multiple times and retains the one with the highest vote. Unlike R2QCIF, these baselines select pseudo labels using model-internal signals rather than revision-derived evidence. All methods use the same prompts and parameter settings.

5.2 Overall Performance

Tables 3 and 4 show that R2QCIF consistently improves over Base and Cold Start across datasets and backbone models. On Reportif, R2QCIF improves All_acc from 56.05 to 86.05 with Llama3.1-8B-Instruct and from 43.16 to 84.74 with Qwen3-8B over Cold Start. Similar gains are observed on MEDEC-MS, where R2QCIF improves All_acc from 40.20 to 46.57 and from 45.06 to 50.59, respectively. These results indicate that revision traces can provide effective supervision beyond the limited explicit labels in α.

Compared with self-training baselines, R2QCIF achieves stronger performance on most accuracy metrics. On Reportif, it improves All_acc over SSRM from 82.89 to 86.05 with Llama3.1-8B-Instruct and from 81.58 to 84.74 with Qwen3-8B. On MEDEC-MS, R2QCIF also obtains the best label-scarce All_acc for both backbones. This suggests that pseudo labels verified by revision evidence are more reliable than those selected only by model-internal confidence, consistency, or voting.

R2QCIF also narrows the gap to FullSFT, which uses gold labels from β. On Reportif, its position accuracy reaches 89.74 and 90.26, close to FullSFT scores of 91.58 and 91.05. This indicates that draft-to-final revisions provide strong behav-

Model	Method	Reportif				MEDEC-MS			
		Pos_acc	Type_acc	Rea_bleurt	All_acc	Pos_acc	Type_acc	Cor_bleurt	All_acc
Llama3.1-8B-Instruct	Cold Start	63.16 (-26.58)	63.68 (-25.53)	52.56 (-4.61)	56.05 (-30.00)	51.93 (-7.37)	**71.38** (+2.57)	62.46 (-0.55)	40.20 (-6.37)
	R2QCIF w/o Bpos&Γ	87.89 (-1.85)	87.37 (-1.84)	56.79 (-0.38)	83.95 (-2.10)	53.94 (-5.36)	41.54 (-27.27)	62.56 (-0.45)	40.54 (-6.03)
	R2QCIF w/o Γ	88.95 (-0.79)	88.16 (-1.05)	57.16 (-0.01)	85.26 (-0.79)	**61.81** (+2.51)	53.10 (-15.71)	60.80 (-2.21)	**51.76** (+5.19)
	R2QCIF	**89.74**	**89.21**	**57.17**	**86.05**	59.30	68.81	**63.01**	46.57

Table 5: Ablation study of revision-supported pseudo-label verification. Parenthesized values denote $\Delta = \text{current} - \text{R2QCIF}$; negative values indicate degradation relative to the full model.

ioral evidence for error localization. However, the remaining gap in type accuracy and rationale similarity shows the boundary of implicit supervision: revision traces reveal what changed, but they do not always explicitly provide the error category or reasoning. Results on IteraTeR further support this conclusion, where R2QCIF achieves the best label-scarce All_acc for both backbones, demonstrating that revision-to-supervision conversion can transfer beyond radiology report quality control.

5.3 Ablation Experiments

Table 5 shows that removing revision-based verification weakens R2QCIF, especially on Reportif. This confirms that IF2 improves self-training by checking whether candidate pseudo labels are supported by revision traces, rather than simply expanding the pseudo-label pool.

The MEDEC-MS results also show that counterfactual verification Γ controls pseudo-label quality. Removing Γ increases All_acc but sharply reduces Type_acc and Cor_bleurt, suggesting that accepting more pseudo labels may improve coarse-grained correctness while harming fine-grained reliability. Thus, IF2 is designed to transform reliable revision evidence, not to maximize pseudo-label utilization.

5.4 Iteration Analysis

Taking the performance of R2QCIF on Reportif with Llama3.1-8B-Instruct as an example, Figures 3 and 4 show that implicit-data utilization and pseudo-label pass rate gradually converge during self-training. This indicates that R2QCIF first converts easier revision traces into pseudo-label supervision, while later iterations face harder cases whose revision evidence cannot be reliably aligned with model-generated labels.

This convergence highlights the boundary of behavioral-trace-as-supervision: revision traces are useful because they encode correction behavior, but they are not equivalent to complete manual annotations. R2QCIF therefore improves training by selectively converting reliable traces into fine-grained supervision.

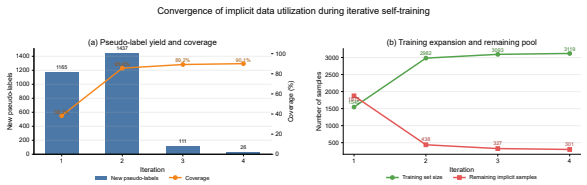

Figure 3: Implicit data utilization during self-training.

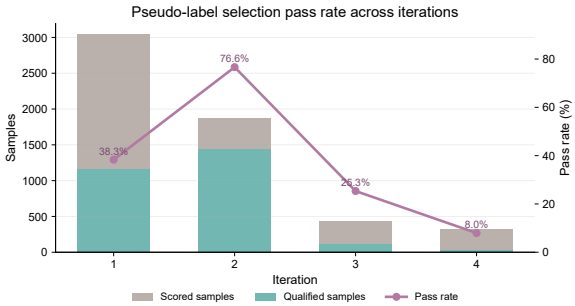

Figure 4: Pseudo-label pass rate across iterations.

Conclusion

In this paper, we propose R2QCIF, a behavioral-trace-as-supervision framework for radiology report quality control. R2QCIF addresses the supervision granularity gap between scarce explicit fine-grained labels and abundant clinical revision traces by treating draft-to-final physician revisions as naturally occurring implicit annotations. Through IF2, R2QCIF verifies whether model-generated pseudo labels are supported by revision evidence via edit-location alignment, confidence estimation, and counterfactual state verification, thereby converting implicit annotations into usable fine-grained supervision. Experiments on Reportif and MEDEC-MS show that R2QCIF improves lightweight LLMs under limited annotation, privacy, and computational-resource constraints. Extended experiments on IteraTeR further demonstrate that the proposed framework can transfer to broader versioned editing scenarios with implicit-annotation structures. These results suggest that clinical behavioral traces, when properly verified, can serve as a scalable supervision source for medical AI.

Limitations

Although R2QCIF demonstrates the value of using clinical revision traces as supervision, several limitations remain. First, R2QCIF reduces but does not eliminate the need for manual annotations. It still requires a small set of high-quality fine-grained labels for cold-start training, which provides the initial quality-control capability and mitigates error accumulation during self-training. Second, iterative self-training introduces additional time costs because it requires multiple rounds of pseudo-label generation, IF2 verification, and model updating. Although the method can be run under relatively limited computational resources, the total training time increases with the number of iterations. Third, our current experiments mainly focus on single-error samples. More complex scenarios, where one report may contain multiple interacting errors, remain underexplored. In such cases, error localization, revision attribution, and pseudo-label verification may become more challenging.

Future work will explore more annotation-efficient cold-start strategies, improve the efficiency of revision-aware self-training, and extend R2QCIF to multi-error radiology report quality control.

Ethical Considerations

This study focuses on radiology report quality control based on clinical revision traces and LLM self-training. R2QCIF is intended to assist report quality assessment and error detection, rather than replace radiologists or provide automated clinical diagnosis. Any practical deployment should undergo institutional validation and remain under the supervision of qualified medical professionals.

Our work uses MIMIC-CXR, a publicly available and de-identified dataset of chest X-ray images and radiology reports. The dataset has been de-identified according to HIPAA Safe Harbor requirements, and its access and use are governed by the PhysioNet data use agreement. We use the data only for research purposes, do not attempt to re-identify patients, and do not disclose protected health information.

Although R2QCIF reduces reliance on online teacher models and explicit large-scale annotation, LLM-based quality-control systems may still produce erroneous, biased, or overconfident outputs. The model may miss clinically important report errors or incorrectly flag accurate reports as problematic. If used without human oversight, such outputs may increase review burden, undermine user trust, or mislead downstream decisions. Therefore, R2QCIF should only be used as an auxiliary quality-control tool, and final clinical judgment should remain with qualified medical professionals.

In addition, models trained from clinical reports and revision traces may inherit dataset-specific biases, including hospital-specific reporting styles, patient population differences, disease distributions, and imaging protocols. Before deployment in different hospitals, languages, or clinical workflows, external validation, subgroup analysis, and continuous monitoring are necessary.

References

Asma Ben Abacha, Wen-wai Yim, Yujuan Fu, Zhaoyi Sun, Meliha Yetisgen, Fei Xia, and Thomas Lin. 2025. MEDEC: A benchmark for medical error detection and correction in clinical notes. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 22539–22550, Vienna, Austria. Association for Computational Linguistics.

Adrian P Brady. 2017. Error and discrepancy in radiology: inevitable or avoidable? *Insights into imaging*, 8(1):171–182.

Shenshen Bu, Taiji Li, Yuedong Yang, and Zhiming Dai. 2024. Instance-level expert knowledge and aggregate discriminative attention for radiology report generation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 14194–14204.

Luc Builtjes, Joeran Bosma, Mathias Prokop, Bram van Ginneken, and Alessa Hering. 2025. Leveraging open-source large language models for clinical information extraction in resource-constrained settings. *JAMIA open*, 8(5):ooaf109.

Weixing Chen, Yang Liu, Ce Wang, Jiarui Zhu, Guanbin Li, Cheng-Lin Liu, and Liang Lin. 2025. Cross-modal causal representation learning for radiology report generation. *IEEE Transactions on Image Processing*.

DeepSeek-AI. 2026. Deepseek-v4: Towards highly efficient million-token context intelligence.

Hanze Dong, Wei Xiong, Bo Pang, Haoxiang Wang, Han Zhao, Yingbo Zhou, Nan Jiang, Doyen Sahoo, Caiming Xiong, and Tong Zhang. 2024. Rlhf workflow: From reward modeling to online rlhf. *Preprint*, arXiv:2405.07863.

Richard Fitzgerald. 2001. Error in radiology. *Clinical radiology*, 56(12):938–946.

Roman Johannes Gertz, Thomas Dratsch, Alexander Christian Bunck, Simon Lennartz, Andra-Iza Iuga, Martin Gunnar Hellmich, Thorsten Persigehl, Lenhard Pennig, Carsten Herbert Gietzen, Philipp Fervers, David Maintz, Robert Hahnfeldt, and Jonathan Kottlors. 2024. Potential of gpt-4 for detecting errors in radiology reports: Implications for reporting accuracy. *Radiology*, 311(1):e232714. PMID: 38625012.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, and 1 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Difei Gu, Yunhe Gao, Yang Zhou, Mu Zhou, and Dimitris Metaxas. 2025. Radalign: Advancing radiology report generation with vision-language concept alignment. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 484–494. Springer.

Yifei He, Haoxiang Wang, Ziyan Jiang, Alexandros Papangelis, and Han Zhao. 2024. Semi-supervised reward modeling via iterative self-training. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 7365–7377, Miami, Florida, USA. Association for Computational Linguistics.

Alyssa Huang, Oishi Banerjee, Kay Wu, Eduardo Pontes Reis, and Pranav Rajpurkar. 2024. Fineradscore: A radiology report line-by-line evaluation technique generating corrections with severity scores. *Preprint*, arXiv:2405.20613.

Jiaxin Huang, Shixiang Gu, Le Hou, Yuexin Wu, Xuezhi Wang, Hongkun Yu, and Jiawei Han. 2023. Large language models can self-improve. In *Proceedings of the 2023 conference on empirical methods in natural language processing*, pages 1051–1068.

Alistair E. W. Johnson, Tom J. Pollard, Seth J. Berkowitz, Nathaniel R. Greenbaum, Matthew P. Lungren, Chih-ying Deng, Roger G. Mark, and Steven Horng. 2019. Mimic-cxr, a de-identified publicly available database of chest radiographs with free-text reports. *Scientific Data*, 6(1):317.

Zae Myung Kim, Wanyu Du, Vipul Raheja, Dhruv Kumar, and Dongyeop Kang. 2022. Improving iterative text revision by learning where to edit from other revision tasks. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 9986–9999.

Bastien Le Guellec, Alexandre Lefèvre, Charlotte Geay, Lucas Shorten, Cyril Bruge, Lotfi Hacein-Bey, Philippe Amouyel, Jean-Pierre Pruvo, Gregory Kuchcinski, and Aghiles Hamroun. 2024. Performance of an open-source large language model in extracting information from free-text radiology reports. *Radiology: Artificial Intelligence*, 6(4):e230364.

Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, and 1 others. 2024. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*.

Aixin Liu, Aoxue Mei, Bangcai Lin, Bing Xue, Bingxuan Wang, Bingzheng Xu, Bochao Wu, Bowei Zhang, Chaofan Lin, Chen Dong, and 1 others. 2025. Deepseek-v3. 2: Pushing the frontier of open large language models. *arXiv preprint arXiv:2512.02556*.

Johannes Moll, Louisa Fay, Asfandyar Azhar, Sophie Ostmeier, Sergios Gatidis, Tim Lueth, Curtis P. Langlotz, and Jean-Benoit Delbrouck. 2025. Structuring radiology reports: Challenging LLMs with lightweight models. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 7707–7724, Suzhou, China. Association for Computational Linguistics.

RE Motyer, S Liddy, WC Torreggiani, and O Buckley. 2016. Frequency and analysis of non-clinical errors made in radiology reports using the national integrated medical imaging system voice recognition

dictation software. *Irish Journal of Medical Science (1971-)*, 185(4):921–927.

Sebastian Nowak, Benjamin Wulff, Yannik C Layer, Maike Theis, Alexander Isaak, Babak Salam, Wolfgang Block, Daniel Kuetting, Claus C Pieper, Julian A Luetkens, and 1 others. 2025. Privacy-ensuring open-weights large language models are competitive with closed-weights gpt-4o in extracting chest radiography findings from free-text reports. *Radiology*, 314(1):e240895.

Sophie Ostmeier, Justin Xu, Zhihong Chen, Maya Varma, Louis Blankemeier, Christian Bluethgen, Arne Edward Michalson, Michael Moseley, Curtis Langlotz, Akshay S Chaudhari, and Jean-Benoit Delbrouck. 2024. GREEN: Generative radiology report evaluation and error notation. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 374–390, Miami, Florida, USA. Association for Computational Linguistics.

Luciano M. Prevedello, Stephen Ledbetter, Cameron Farkas, and Ramin Khorasani. 2014. Implementation of speech recognition in a community-based radiology practice: Effect on report turnaround times. *Journal of the American College of Radiology*, 11(4):402–406.

Qwen Team. 2025. Qwq-32b: Embracing the power of reinforcement learning.

Qwen Team. 2026. Qwen3.6-35B-A3B: Agentic coding power, now open to all.

DL Renfrew, EA Franken Jr, KS Berbaum, FH Weigelt, and MM Abu-Yousef. 1992. Error in radiology: classification and lessons in 182 cases presented at a problem case conference. *Radiology*, 183(1):145–150.

Babak Salam, Claire Stüwe, Sebastian Nowak, Alois M Sprinkart, Maike Theis, Dmitrij Kravchenko, Narine Mesropyan, Tatjana Dell, Christoph Endler, Claus C Pieper, and 1 others. 2025. Large language models for error detection in radiology reports: a comparative analysis between closed-source and privacy-compliant open-source models. *European Radiology*, 35(8):4549–4557.

Thibault Sellam, Dipanjan Das, and Ankur P Parikh. 2020. Bleurt: Learning robust metrics for text generation. In *Proceedings of ACL*.

Leyao Wang, Zhiyu Wan, Congning Ni, Qingyuan Song, Yang Li, Ellen Clayton, Bradley Malin, and Zhijun Yin. 2024. Applications and concerns of chatgpt and other conversational large language models in health care: Systematic review. *J Med Internet Res*, 26:e22769.

Jinge Wu, Zhaolong Wu, Ruizhe Li, Tong Chen, Abul Hasan, Yunsoo Kim, Jason P. Y. Cheung, Teng Zhang, and Honghan Wu. 2025. Error correction in radiology reports: A knowledge distillation-based multi-stage framework. *Preprint*, arXiv:2406.15045.

Mingxuan Xia, Haobo Wang, Yixuan Li, Zewei Yu, Jindong Wang, Junbo Zhao, and Runze Wu. 2025. Prompt candidates, then distill: A teacher-student framework for llm-driven data annotation. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2750–2770.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, and 23 others. 2024. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*.

Chenlu Ye, Wei Xiong, Yuheng Zhang, Hanze Dong, Nan Jiang, and Tong Zhang. 2024. Online iterative reinforcement learning from human feedback with general preference model. In *Advances in Neural Information Processing Systems*, volume 37, pages 81773–81807. Curran Associates, Inc.

Xiaoying Zhong, Siyi Li, Zhao Chen, Long Ge, Dongdong Yu, Shijia Wang, Liangzhen You, and Hongcai Shang. 2025. Considerations for patient privacy of large language models in health care: scoping review. *Journal of Medical Internet Research*, 27:e76571.

A Dataset Details

In this appendix, we provide additional details about the construction of REPORTIF, including the error injection rules and prompt templates used for synthetic error generation. Due to the access restrictions and data use agreement of MIMIC-CXR, we do not directly release synthetic samples derived from MIMIC-CXR reports. Instead, we provide the construction protocol, including the error taxonomy, injection rules, prompt templates, and validation procedure. Researchers with authorized access to MIMIC-CXR can reproduce REPORTIF on their local authorized copy of the dataset.

A.1 Error Injection Rules

Each synthetic sample in REPORTIF is constructed by injecting exactly one predefined error into a clean radiology report. The injection is restricted to clinically relevant sections, including `FINDINGS`, `IMPRESSION`, and `RECOMMENDATION`. All unrelated content, including section headers, punctuation, capitalization, spacing, and line breaks, is preserved whenever possible. The complete error taxonomy is shown in Table 6.

For error types with multiple realization methods, we define sub-methods and assign samples approximately uniformly across them. For example, typographical errors are balanced across character omission, character transposition, character duplication, and keyboard-neighbor replacement. Template-residue errors are balanced across sentence duplication and unresolved placeholders. Recommendation–finding mismatches are balanced across wrong-modality and wrong-body-region recommendations. This design reduces repetitive generation patterns and improves the diversity of synthetic errors.

A.2 Prompt Templates

For error types that cannot be reliably generated by deterministic rules, we use an LLM-based injection procedure. The model is instructed to inject exactly one predefined error while preserving all unrelated report content verbatim. The prompt consists of a global instruction template, an error-specific instruction, and output validation requirements, as shown in Tables 7–9. The placeholder `{REPORT_TEXT}` denotes the input report and is not released with this paper.

A.3 Licenses and Intended Use

Licenses and access. MIMIC-CXR is used under the PhysioNet data use agreement, we do not redistribute any MIMIC-CXR-derived synthetic samples. MEDEC-MS and IteraTeR are used through their original distributions and in accordance with their original release terms.

Consistency with intended use. All datasets are used only for research evaluation and method development. We do not use them for clinical diagnosis, autonomous clinical decision-making, or deployment in real clinical workflows. We do not attempt to re-identify patients or disclose protected health information. The use of MIMIC-CXR-derived data is restricted to local experiments, and no MIMIC-CXR reports or derived samples are transmitted to third-party online model services.

B Training Details

This section provides the detailed training and inference settings of R2QCIF.

R2QCIF is trained through an iterative self-training pipeline. In each iteration, the current model is first fine-tuned on the available supervised data, including the cold-start annotations and the pseudo-labeled samples selected from previous iterations. The updated model then generates pseudo labels for the remaining revision-trace samples. These candidates are verified by IF2 through edit-location alignment, confidence-based stability checking, and counterfactual state verification. The accepted pseudo-labeled samples are added to the training set for the next iteration.

The self-training process stops when the utilization gain of revision-trace samples is lower than 0.01 or when the maximum number of iterations is reached. After the iterative process terminates, we retrain the final model using the union of the cold-start data and all accepted pseudo-labeled samples.

B.1 Experimental Settings

All experiments use LoRA fine-tuning for 8B-scale backbone models. In each self-training iteration, pseudo-label generation uses temperature 0.7 and top_p 0.8. For final evaluation, we use greedy decoding with `do_sample=False` to ensure deterministic outputs. The data gain threshold is set to $\theta = 0.01$, and the maximum number of self-training iterations is set to 10. Additional implementation details follow the settings described in the main experiments.

Code	Error Type	Target Section	Injection Operation	Constraints
0-1	Typographical error	FINDINGS or IMPRESSION	Create one spelling error by deleting one internal character, swapping adjacent characters, repeating one character, or replacing one character with a neighboring QWERTY key.	Edit exactly one token once. Do not modify numbers, units, dates, punctuation, proper nouns, or stop words. Keep the sentence readable and clinically interpretable.
0-2	ASR-like homophone error	FINDINGS or IMPRESSION	Replace one word with a homophone or near-homophone that preserves local fluency but changes lexical identity.	Edit exactly one word. Do not modify numbers, units, measurements, dates, or laterality markers. Prefer substitutions plausible in speech-recognition scenarios.
0-3	Template residue	FINDINGS or IMPRESSION	Duplicate one generic sentence immediately after itself, or replace one concrete detail with a placeholder such as [DEGREE], [DATE], [SIDE], or [SIZE].	Use only one method. Only one localized edit is allowed. Do not alter unrelated sentences.
1-1A	Diagnostic omission	IMPRESSION	Remove a conclusion corresponding to one explicit key finding in FINDINGS.	Do not change FINDINGS. Do not delete all IMPRESSION content. Keep the remaining IMPRESSION fluent and realistic.
1-1B	Diagnostic misinterpretation	IMPRESSION	Replace one conclusion with a clinically plausible but incorrect interpretation of the same finding.	Keep FINDINGS unchanged. Do not use generic normal or negative templates. Modify only one conclusion statement.
1-1C	Evidence deletion	FINDINGS	Delete one sentence that serves as the sole evidence for a conclusion in IMPRESSION.	Do not rewrite other FINDINGS sentences. Ensure that the deleted evidence is not redundantly stated elsewhere. Keep IMPRESSION unchanged.
1-2	Severity misjudgment	IMPRESSION	Change qualitative severity or acuity wording, such as mild to severe, acute to chronic, or stable to progressive.	Modify descriptors only. Keep entities and sentence structure largely unchanged. Do not change explicit numbers or units.
1-3	Recommendation–finding mismatch	IMPRESSION or RECOMMENDATION	Modify or add one management statement so that it is plausible in form but mismatched to the report context.	Introduce exactly one management mismatch, such as a wrong imaging modality or a wrong anatomical region. Do not alter factual findings elsewhere.
1-4	Lesion information conflict	Preferably FINDINGS	Modify one lesion attribute in one section, such as laterality, anatomical location, size, count, or unit.	The same lesion entity must be explicitly present in both FINDINGS and IMPRESSION. Change only one attribute in one section. Ensure that the contradiction is detectable from text alone.

Table 6: Error injection rules used to construct REPORTIF. Each sample contains exactly one injected error.

Global prompt for error injection
You are now a specialist in radiology, and you are given a clean radiology report. Your task is to inject EXACTLY ONE predefined error into the report based on the provided definition.
Strict edit constraints: 1. Modify EXACTLY ONE section: FINDINGS, IMPRESSION, or RECOMMENDATION. Do not touch any other sections. To create cross-section inconsistency, still edit only one section. 2. Make ONLY ONE localized edit in total: modify, add, or delete one sentence. Do not introduce unprompted errors, findings, or diagnoses. 3. Choose exactly one action type: Modify, Add, or Delete. Never combine actions. 4. The edited section must not become empty. Preserve all unrelated text verbatim, including punctuation, spacing, and line breaks. The modified report must remain fluent and realistic. 5. If deletion would make the chosen section empty, return ["", "", "", ""].
Input format: Error Type Code: {ERROR_TYPE_CODE} Sub-method Constraint: {SUB_METHOD_INSTRUCTION} Original report: {REPORT_TEXT}

Table 7: Global prompt template used for LLM-based error injection.

Code	Error-specific instruction
0-3	Create exactly one template-residue error in FINDINGS or IMPRESSION. Use either sentence duplication or placeholder replacement. Use only one method and do not alter unrelated sentences.
1-1A	Create evidence–conclusion inconsistency by omission. Edit IMPRESSION only: remove the conclusion corresponding to one explicit key finding in FINDINGS. Do not change FINDINGS.
1-1B	Create diagnostic misinterpretation in IMPRESSION while keeping FINDINGS unchanged. Replace one conclusion with a clinically plausible but incorrect interpretation of the same finding.
1-1C	Create evidence–conclusion inconsistency by removing key evidence. Edit FINDINGS only: delete one sentence that serves as the sole evidence for a conclusion in IMPRESSION. Keep IMPRESSION unchanged.
1-2	Create severity misjudgment in IMPRESSION. Change qualitative severity or acuity wording without changing anatomy, numbers, or units.
1-3	Create recommendation–finding mismatch in management content. Modify or add one follow-up statement so that it is plausible in form but mismatched to the report context.
1-4	Create lesion information conflict between FINDINGS and IMPRESSION. Modify one lesion attribute in only one section, preferably FINDINGS. The contradiction must be detectable from text alone.

Table 8: Error-specific instructions inserted into the LLM prompt. Deterministic rules are used first for 0-1 and 0-2; the LLM is used as fallback when rule-based injection fails.

Output format and validation requirements
The model must output a valid JSON array and nothing else: `["Full modified report text", "Raw sentence", "Modified sentence", "Wrong reason"]`
Field rules: 1. `Full modified report text`: the complete report after the edit. The final report must not contain special markers such as `[DELETED]` or `[NONE]`. 2. For deletion, `Raw sentence` is the exact sentence removed from the original report, and `Modified sentence` is `[DELETED]`. 3. For addition, `Raw sentence` is `[NONE]`, and `Modified sentence` is the exact new sentence. 4. For modification, `Raw sentence` is the exact original sentence, and `Modified sentence` is the exact changed sentence. 5. `Wrong reason` must provide a professional explanation of the error based only on the modified report text, without mentioning injection or prompt instructions.
Automatic validation: The output is accepted only if it is a JSON array with exactly four string fields; the modified report is non-empty and different from the original report; the reported raw and modified sentences are consistent with the original and modified reports; and replacing, deleting, or adding the reported sentence recovers the full modified report up to whitespace normalization.

Table 9: Output format and automatic validation requirements for LLM-generated samples.

Hyperparameter	Value
Backbones	Llama3.1-8B-Instruct, Qwen3-8B
Max length / precision	2048 / bfloat16
LoRA r/α/dropout	16 / 16 / 0
LR / epochs	2×10^{-4} / 3
Batch / grad. accum.	2 / 4
Optimizer / scheduler	AdamW 8-bit / cosine
Max iterations	10
Stopping threshold	0.01
Selection thresholds	0.4 / 0.8 / 0.8
Counterfactual threshold	0.4
Generation	T = 0.7, top_p = 0.8
Evaluation	Greedy decoding
Seed	42

Table 10: Main training and inference settings for R2QCIF.

All experiments are conducted on a single NVIDIA RTX 3090 GPU with 24 GB memory. The backbone models have approximately 8B parameters. Each self-training run takes approximately 5 GPU hours, and the total computational budget ranges from approximately 20 to 48 GPU hours. Unless otherwise specified, all reported results are from a single run with seed 42; final evaluation uses greedy decoding to ensure deterministic decoding.

B.2 Implementation Details of IF2

During pseudo-label generation, model outputs are parsed using the predefined field markers `Error Type`, `Error Sentence`, and `Reason`. Outputs that do not contain all required fields, predict multiple error instances, or produce an error type outside the predefined taxonomy are treated as invalid error candidates and rejected before IF2 filtering. For samples with no revision difference, a prediction of "No error found" is treated as a negative pseudo label. For samples with a non-empty revision signal, however, "No error found" does not provide a valid error position, type, or reason, and therefore is not passed to IF2 as an error pseudo-label. Malformed outputs are not manually repaired.

For edit-location filtering, reports are segmented into sentences and the edited location D_i is extracted by sentence-level differencing between the pre-revision report X_i^- and the revised report X_i^+.

Both the predicted error sentence and the edited sentence are normalized by lowercasing and collapsing consecutive whitespaces before tokenization. Token-level IoU is then computed between the normalized token sets. In the synthetic single-error setting, each accepted implicit-feedback instance contains one edited source sentence by construction. Instances for which a unique edited sentence cannot be identified are discarded.

For confidence filtering, we record token probabilities during decoding. The token spans corresponding to the error type and reason are obtained by aligning the decoded response with the parsed field values. The confidence of each field is computed as the mean probability of the generated tokens in that field. If a parsed field cannot be aligned with the generated tokens, the candidate is rejected. The thresholds (τ_p, τ_T, τ_R) are set to $(0.4, 0.8, 0.8)$, and the counterfactual conflict threshold γ is set to 0.4. These thresholds are empirically set and kept fixed across all datasets and backbone models.

For counterfactual state verification, the same model is applied to the revised report X_i^+. If the model predicts "No error found", the candidate passes this verification step. Otherwise, the candidate is rejected only when the prediction on X_i^+ has the same error type as the candidate pseudo label and the token-level IoU between their predicted positions exceeds γ.

B.3 Inference Prompt

Table 11 shows the prompt used during pseudo-label generation and final evaluation. The same inference prompt is used across all backbone models and experimental settings. During pseudo-label generation, the model samples responses with temperature 0.7 and top-p 0.8 to improve output diversity. During final evaluation, we use greedy decoding to obtain deterministic results.

B.4 Training Prompt

Table 12 shows the prompt template used to construct supervised fine-tuning samples. For samples with errors, the assistant response consists of the error type, erroneous sentence, and error reason. For samples without errors, the target response is simply "No error found." This design keeps the training and inference formats consistent.

For Llama3.1-8B-Instruct, we wrap the prompt-response pair with the Llama chat template. For Qwen3-8B, we use the tokenizer chat template with the user and assistant roles, and disable the thinking mode during inference.

Inference prompt used in R2QCIF
You are now a professional radiologist responsible for identifying basic issues and logical problems in radiology reports, and you must provide detailed descriptions of the error type, the erroneous sentence, and the reason for the error. Please carefully analyze each report to ensure accurate quality control analysis.
The radiology report you need to review is shown below: — `{report}` —
Each report contains at most one problem. The possible problem types include: Typographical error, ASR-like homophone error, Template residue, Diagnostic presence/absence error, Severity misjudgment, Recommendation–finding mismatch, Lesion information conflict. If there is no error in the report, please output "No error found." Otherwise, please return your quality control analysis in the following format:
Error Type: [type of error] Error Sentence: [the sentence with error] Reason: [explanation of the error]

Table 11: Inference prompt used for pseudo-label generation and final evaluation. The placeholder `{report}` is replaced by the input radiology report.

Training prompt used for supervised fine-tuning
System prompt: You are now a professional radiologist responsible for identifying basic issues and logical problems in radiology reports, and you must provide detailed descriptions of the error type, the erroneous sentence, and the reason for the error. Please carefully analyze each report to ensure accurate quality control analysis.
Task prompt: The radiology report you need to review is shown below: — `{false_text}` —
Each report contains at most one problem. The possible problem types include: Typographical error, ASR-like homophone error, Template residue, Diagnostic presence/absence error, Severity misjudgment, Recommendation–finding mismatch, Lesion information error. If there is no error in the report, please output "No error found." Otherwise, please return your quality control analysis in the following format:
Error Type: [type of error] Error Sentence: [the sentence with error] Reason: [explanation of the error]
Training response for error samples: Error Type: `{error_type}` Error Sentence: `{false_sentence}` Reason: `{reason}`
Training response for no-error samples: No error found.

Table 12: Training prompt template used to construct supervised fine-tuning samples.