SELF-EDITING AND PREFERENCE DISTILLATION FOR CHEST X-RAY REPORT GENERATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Reinforcement learning has advanced chest X-ray report generation, but assigning a single report-level reward can suppress correct findings in low-reward reports and reinforce local errors in high-reward ones. Self-refinement offers a way to address such local errors, yet existing approaches have limitations: inference-time methods can be unreliable without external feedback, while training-time methods rely on externally derived corrections or refinement supervision rather than explicitly optimizing the generation model itself to perform self-refinement. We propose ED-SR (Editing and Distillation for Self-Refinement), a training-time reinforcement learning framework in which a single vision-language model serves as both generator and editor. The generator produces candidate reports, while the editor re-examines the images and its generated drafts to identify potential errors, explain why they are problematic, and propose sentence-level revisions, deletions, and additions. ED-SR augments report-level policy optimization with two losses: (i) a *counterfactual edit-reward* loss that assigns fine-grained credit to self-generated editing actions using the reward change from applying each edit alone, and (ii) a *correction-distillation preference* loss that trains the generation policy to prefer reward-improving revisions over their original drafts. Because the generator and editor share parameters, edit-level supervision directly updates the same model used for report generation, while gradient conflict resolution manages competing report-level and preference updates. At inference time, ED-SR generates reports in a single pass without an additional refinement stage. Experiments demonstrate state-of-the-art performance across natural language generation and clinical efficacy metrics, achieving the best results on 9 of 11 metrics on MIMIC-CXR and all evaluated metrics on CheXpert Plus. Code will be released upon acceptance.

1 INTRODUCTION

Chest X-ray report generation (CXR-RG) aims to produce clinically accurate reports from chest X-rays, reducing radiologists' workload and improving reporting consistency. Recent studies have applied reinforcement learning (RL) to this field with promising results. However, most of these methods compute a reward for the complete report and assign the resulting report-level advantage uniformly to all generated tokens (Guo et al., 2026; Ding et al., 2025; Xiao et al., 2025; Yi et al., 2025). Because a report contains multiple findings of varying correctness, uniform reward assignment provides only coarse credit, which can suppress correct findings in low-reward reports and reinforce local errors in high-reward ones without indicating which sentences should be modified.

Self-correction and refinement provide a way to revisit local errors, but both inference-time and training-time approaches have limitations. Several CXR-RG methods revise draft reports at inference time (Park et al., 2025; Dong et al., 2026; Liu et al., 2024). However, proposed revisions are not uniformly beneficial, and without reliable external feedback, self-correction can introduce new errors or overwrite correct content (Kamoi et al., 2024; Xu et al., 2024; Huang et al., 2024; Zhang et al., 2025c). This has motivated training-time approaches that exploit reference reports to construct improved supervision (Zhang et al., 2025b; Liu et al., 2026c; Zhou et al., 2026). However, these methods either rely on reference-derived corrections or use external models to identify errors or construct refined targets, rather than explicitly optimizing the generation model itself to perform self-refinement during training.

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.

To address this gap, we propose **ED-SR**, which combines self-editing and preference distillation to explicitly train the model to identify potential errors in its own output, explain why they are problematic, and propose targeted revisions. By learning from this error-correction process, the model can reduce similar errors in subsequent generations and improve its first-pass generation capabilities. Specifically, a single vision-language model (VLM) serves as both generator and editor. During RL, the generator produces candidate reports, while the editor re-examines the images and generated reports to propose structured edits. Each edit specifies a sentence-level revision, deletion, or addition together with a rationale for the proposed change. ED-SR augments report-level policy optimization with two objectives derived from these self-proposed edits: a counterfactual edit-reward loss and a correction-distillation preference loss.

(i) The *counterfactual edit-reward* loss evaluates each edit individually by measuring the reward change obtained by applying that edit alone to the original draft. These reward changes provide fine-grained credit that reinforces beneficial edits and penalizes harmful ones. Because the generator and editor share model parameters, the edit-level objective updates the same model used for report generation. (ii) The *correction-distillation preference* loss directly optimizes the generation policy: when jointly applying reward-improving edits produces a better report, the revised report is preferred over the original candidate, thereby distilling verified revisions into the generation policy. Gradient conflict resolution manages conflicts between the preference and report-level objectives. At inference time, ED-SR generates reports in a single pass without an additional editing stage.

Our main contributions are summarized as follows:

- We propose ED-SR, a training-time self-refinement framework in which a single VLM both generates reports and proposes local edits. A counterfactual edit-reward loss assigns fine-grained credit to individual editing actions.
- We introduce a correction-distillation preference loss with gradient conflict resolution to distill reward-improving revisions directly into the generation policy, enabling single-pass inference without an additional editing stage.
- Extensive experiments on MIMIC-CXR and CheXpert Plus demonstrate state-of-the-art performance across natural language generation and clinical efficacy metrics.

2 Related Work

Radiology report generation. Standard CXR-RG methods focus on cross-modal feature alignment (Xing et al., 2025; Yang et al., 2026; Deria et al., 2026; Wang et al., 2026a), memory mechanisms (Wang et al., 2025b; Wu et al., 2026), efficient backbones (Wang et al., 2025a), and longitudinal progression (Liu et al., 2026b;a; 2025a;b; Yang et al., 2025). Recent RL-based works align training with evaluation criteria, typically using report-level metric rewards (Guo et al., 2026; Ding et al., 2025; Xiao et al., 2025; Yi et al., 2025). However, a single report-level advantage does not identify which local findings should be revised. Even entity-based rewards are typically aggregated into a single scalar whose advantage is shared across all tokens (Wang et al., 2026b; Miura et al., 2021; Delbrouck et al., 2022). Our ED-SR augments report-level optimization with counterfactual rewards that assign credit to individual editing actions.

Self-correction and refinement. Inference-time self-correction does not consistently improve accuracy and can introduce new errors without reliable feedback (Zhang et al., 2025c; Huang et al., 2024; Xu et al., 2024; Kamoi et al., 2024). Existing approaches use stronger-model supervision (Qu et al., 2024), separately trained critics (Zhang et al., 2025a; Akyürek et al., 2023), or program-based verification (Song et al., 2025) to guide correction. SCoRe (Kumar et al., 2025) uses multi-turn RL to learn a sequential self-correction strategy, refining responses through additional attempts at inference. ED-SR instead uses self-editing during training to improve direct report generation without an inference-time correction stage. SCL (He et al., 2025) similarly targets direct generation and applies DPO to preference pairs derived from initial and revised responses. ED-SR further complements response-level preference learning with counterfactual rewards for individual edits, enabling finer-grained credit assignment.

Self-correction and refinement in CXR-RG. Several CXR-RG methods revise draft reports at inference time (Liu et al., 2024; Park et al., 2025; Dong et al., 2026). In contrast, ED-SR learns from revisions during training, enabling single-pass inference without an additional correction stage.

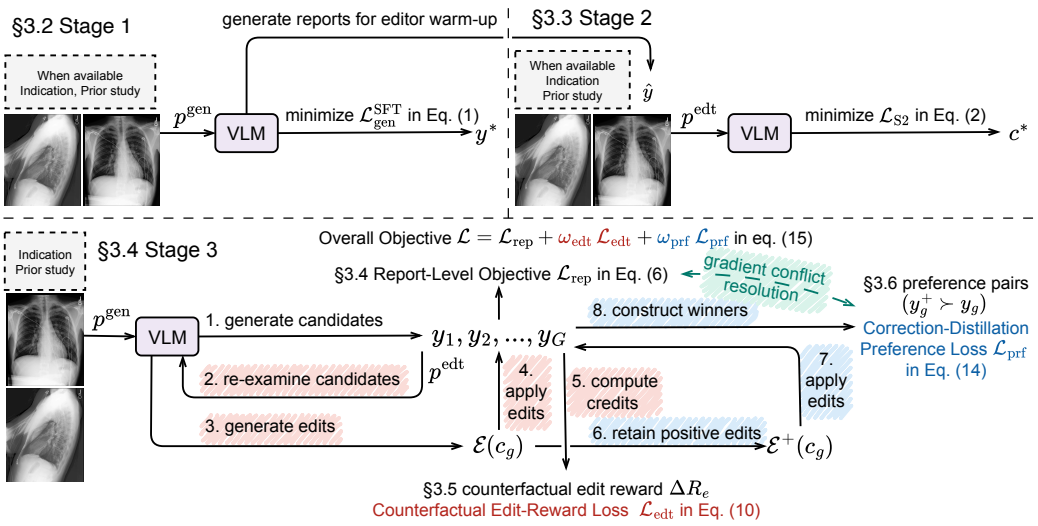


Figure 1: Overview of the proposed ED-SR framework. Stage 1 fine-tunes the VLM as a report generator, and Stage 2 warms up its editor role on a small corpus of edit scripts. In Stage 3, the generator samples candidate reports $y_1, \ldots, y_G$ for report-level optimization via $\mathcal{L}_{\text{rep}}$. In its editor role, the same VLM proposes edits $\mathcal{E}(c_g)$. Each edit is applied independently to the original draft and evaluated by its counterfactual reward ΔR_e, which supervises the corresponding edit block through $\mathcal{L}_{\text{edt}}$. Edits with positive rewards, $\mathcal{E}^+(c_g)$, are jointly applied to construct a revised report y_g^+. The preference pair $(y_g^+ \succ y_g)$ is used to optimize the generator via $\mathcal{L}_{\text{prf}}$ only if the combined revision improves the reward. Gradient conflict resolution manages opposing update directions on the candidate report between the report-level and preference objectives.

Among training-time methods, EditGRPO (Zhang et al., 2025b) combines on-policy rollouts with reference-edited reports as off-policy guidance, whereas ED-SR directly optimizes edit trajectories sampled from its own editing policy, making these on-policy revisions more naturally distillable into the generation policy (Tajwar et al., 2024). SAT-RRG (Liu et al., 2026c) uses a frozen LLM to compare reference and generated reports, identify correct and erroneous tokens, and pull up correct tokens while pushing down erroneous ones. Similarly, ESC-RL (Zhou et al., 2026) uses GPT-5 during training to annotate preferences by comparing generated reports with reference reports, and uses GPT-5 again at inference to revise generated drafts. However, neither method explicitly trains the generation model to perform self-refinement. ED-SR instead lets the VLM propose edits to refine its own draft, while counterfactual rewards and preference distillation reinforce reward-improving revisions into the generation policy. Moreover, ED-SR requires no external LLM calls during inference, reducing potential patient privacy risks.

3 Method

Figure 1 illustrates the proposed ED-SR, a 3-stage framework for training-time self-refinement where a single VLM serves as both generator and editor. Stages 2 and 3 are initialized from the best checkpoint of the preceding stage. Stage 1 adapts a pretrained VLM into a report generator via supervised fine-tuning. Stage 2 warms up the editor role to produce structured edit scripts. Stage 3, the core RL stage, jointly optimizes three losses using reports and edit scripts sampled by the same VLM: (i) report-level group-relative policy optimization; (ii) the *counterfactual edit-reward* loss, which assigns fine-grained credit to individual editing actions; and (iii) the *correction-distillation preference* loss, which distills reward-improving revised reports into the generation policy. The latter two losses are our main contributions and are detailed in §3.5 and §3.6.

3.1 Problem Formulation

Let $\mathcal{D} = \{(x, y^*)\}$ denote the training corpus. A study x comprises up to two current CXRs with view tags, optional clinical context, and, when available, a prior study. The target y^* is the reference

Findings section. The model is an autoregressive VLM $\pi_\theta(y \mid x, p)$ over token sequences $y = (y_1, \ldots, y_{|y|})$, conditioned on the study x and a task prompt p. Two prompts specify the two roles: the generation prompt p^{gen} requests a report y for the study x; the editing prompt $p^{\text{edt}}(y)$ asks the same VLM to re-examine x and y and produce a structured edit script c. The reference report is not included in the editor's input.

3.2 Stage 1: Supervised Report Generation

Stage 1 fine-tunes the generator to maximize the likelihood of y^* given (x, p^{gen}) via

$$\mathcal{L}_{\text{gen}}^{\text{SFT}} = -\frac{1}{|y^*|} \sum_{t=1}^{|y^*|} \log \pi_\theta\big(y_t^* \mid y_{<t}^*, x, p^{\text{gen}}\big). \tag{1}$$

3.3 Stage 2: Editor Warm-Up

Stage 2 initializes the editor role to produce structured scripts for self-editing before RL.

Structured edit scripts. Given a generated report whose sentences are enumerated $S_1, \ldots, S_n$, an edit script $c = (e_1, \ldots, e_K)$ is a sequence of edit blocks, serialized in an XML-like format. Each block performs exactly one of three actions:

- REVISE: $e = (\text{REVISE}, s_e, f_e, r_e)$ replaces the sentence with identifier s_e by a revised sentence r_e, where f_e is a one-sentence rationale for the proposed revision;
- DELETE: $e = (\text{DELETE}, s_e, f_e)$ removes the targeted sentence(s); s_e may reference several sentences when they share the same rationale f_e (e.g., duplicated content);
- ADD: $e = (\text{ADD}, f_e, r_e)$ appends a new sentence r_e to the end of the report, where f_e specifies a missing finding.

Because REVISE and DELETE use explicit sentence identifiers and ADD appends to the report, each script can be applied deterministically and each edit scored independently. Stage 3 uses both properties. An example edit script is shown in Appendix A.1.

Teacher-annotated warm-up corpus. Offline, for each dataset, the frozen Stage 1 model generates a report $\hat{y}$ for each of 100 sampled training studies. GPT-5.6 Sol (OpenAI, 2026) then compares each $\hat{y}$ with the reference y^* to produce a minimal reference-guided edit script c^*, forming $\mathcal{D}_2 = \{(x, y^*, \hat{y}, c^*)\}$. This small teacher-annotated corpus initializes the editor's structured output format and editing behavior before RL. The external LLM is used only to construct this offline warm-up corpus.

Joint warm-up objective. Stage 2 jointly optimizes two cross-entropy losses:

$$\mathcal{L}_{\text{S2}} = \tfrac{1}{2}\,\mathcal{L}_{\text{gen}}^{\text{SFT}} + \tfrac{1}{2}\,\mathcal{L}_{\text{edt}}^{\text{SFT}}. \tag{2}$$

Both losses are computed on $\mathcal{D}_2$. $\mathcal{L}_{\text{gen}}^{\text{SFT}}$ (Eq. 1) retains report-generation supervision, while $\mathcal{L}_{\text{edt}}^{\text{SFT}}$ maximizes the likelihood of c^* given $(x, p^{\text{edt}}(\hat{y}))$ to initialize the editor role.

3.4 Stage 3: RL Settings and Report-Level Optimization

Stage 3 maintains three configurations of the same model: the current policy π_θ, which receives gradients; the behavior policy π_{old}, a frozen snapshot of π_θ that generates rollouts and is refreshed every K_{old} steps; and the reference policy π_{ref}, fixed at the Stage 2 checkpoint, which serves as the anchor for both the Kullback–Leibler (KL) regularizer and the preference loss.

Composite reward. A report y is scored against the reference y^* with a composite reward consisting of two natural language generation metrics and two clinical efficacy metrics,

$$R(y, y^*) = \lambda \left[\text{BLEU-2} + \text{ROUGE-L}\right] + (1 - \lambda) \left[F_1^{\text{RadGraph}} + F_1^{\text{CheXbert}}\right], \tag{3}$$

where $\lambda \in [0, 1]$ balances the two metric families.

Report-level group-relative optimization. For each study x_i in a batch of B studies, following GRPO (Shao et al., 2024), the behavior policy samples a group of G candidate reports,

$y_{i,1}, \ldots, y_{i,G} \sim \pi_{\text{old}}(\cdot \mid x_i, p^{\text{gen}})$, with rewards $r_{i,g} = R(y_{i,g}, y_i^*)$. Advantages are group-mean-centered but normalized by a global scale rather than the per-group standard deviation:

$$\hat{A}_{i,g} = \frac{r_{i,g} - \bar{r}_i}{\bar{\sigma}_r}, \qquad \bar{r}_i = \frac{1}{G}\sum_{g=1}^{G} r_{i,g}. \tag{4}$$

Here, $\bar{\sigma}_r$ is an exponential moving average (EMA) of the group-wise reward standard deviations averaged across the B groups. This shared scale avoids independently amplifying low-variance groups (Liu et al., 2025c) and adapts the normalization to the observed reward variability.

Smooth-gated surrogate. PPO-style hard clipping can abruptly suppress updates beyond the clipping region. We use SAPO's smooth sigmoid gate (Gao et al., 2025), which smoothly attenuates policy-gradient updates as token-level importance ratios deviate from one,

$$\phi_\tau(\rho) = \frac{4}{\tau}\,\sigma\big(\tau(\rho - 1)\big), \qquad \tau = \begin{cases} \tau_+, & \hat{A} > 0, \\ \tau_-, & \hat{A} \le 0, \end{cases} \tag{5}$$

where $\sigma(\cdot)$ is the logistic function. For any sampled sequence z, we denote its per-token importance ratio by $\rho_t(z) = \exp\big(\log \pi_\theta(z_t \mid \cdot) - \log \pi_{\text{old}}(z_t \mid \cdot)\big)$.

The report-level policy-gradient loss $\mathcal{L}_{\text{PG}}^{\text{rep}}$ and its KL-regularized objective $\mathcal{L}_{\text{rep}}$ are

$$\mathcal{L}_{\text{PG}}^{\text{rep}} = -\frac{1}{\sum_{i,g} |y_{i,g}|} \sum_{i,g} \sum_{t=1}^{|y_{i,g}|} \phi_\tau\big(\rho_t(y_{i,g})\big)\, \hat{A}_{i,g}, \qquad \mathcal{L}_{\text{rep}} = \mathcal{L}_{\text{PG}}^{\text{rep}} + \beta^{\text{rep}}\, \mathbb{D}_{\text{KL}}^{\text{rep}}, \tag{6}$$

where $\mathbb{D}_{\text{KL}}^{\text{rep}}$ is the per-token KL regularizer with coefficient β^{rep}. For any sampled sequence z, we use the nonnegative KL penalty (Schulman, 2020) $\hat{D}_t(z) = e^{\delta_t(z)} - \delta_t(z) - 1$, where $\delta_t(z) = \log \pi_{\text{ref}}(z_t \mid \cdot) - \log \pi_\theta(z_t \mid \cdot)$. Accordingly, $\mathbb{D}_{\text{KL}}^{\text{rep}}$ averages $\hat{D}_t(y_{i,g})$ over the same report tokens used in $\mathcal{L}_{\text{PG}}^{\text{rep}}$.

3.5 Counterfactual Edit-Reward Loss

The report-level objective in Eq. 6 assigns the same advantage $\hat{A}_{i,g}$ to all tokens in a report and provides no edit-specific credit. We complement this objective by evaluating the reward change from applying each proposed edit alone to the original draft.

Self-editing rollout. For each candidate report $y_{i,g}$, the behavior policy is prompted again with the editing prompt to generate an edit script

$$c_{i,g} \sim \pi_{\text{old}}\big(\cdot \mid x_i,\, p^{\text{edt}}(y_{i,g})\big), \tag{7}$$

from which we parse the set of edit blocks $\mathcal{E}(c_{i,g})$. Blocks that cannot be parsed or that reference nonexistent identifiers are dropped. If multiple valid edit blocks target the same sentence, we retain only the first block in generation order.

Counterfactual edit reward. Let $y \oplus e$ denote the report obtained by applying edit e alone to y. Each edit is assigned the counterfactual reward difference

$$\Delta R_e = R\big(y_{i,g} \oplus e,\; y_i^*\big) - R\big(y_{i,g},\; y_i^*\big), \quad e \in \mathcal{E}(c_{i,g}). \tag{8}$$

Positive (negative) ΔR_e indicates an increase (decrease) in the composite reward relative to the original draft. Applying each edit independently isolates its effect from other edits in the script. Because each edit typically affects only a small portion of the report, ΔR_e is small in magnitude; we therefore normalize it using a global scale and clip extreme values:

$$\hat{A}_e = \text{clip}\left(\frac{\Delta R_e}{\bar{\sigma}_\Delta},\, -\kappa,\, \kappa\right), \tag{9}$$

where $\bar{\sigma}_\Delta$ is an EMA of the batch-wise standard deviation computed over all ΔR_e values, and κ bounds the influence of outlier edits.

Quality gate and loss. We localize credit to the tokens of each edit block. Writing $t \in e$ for the token positions within edit block e, every such token receives the block's advantage $\hat{A}_e$, assigning

credit at the edit level rather than to the entire script. We also discard scripts containing no beneficial edits, as uniformly penalizing all edits in such scripts may provide noisy supervision and encourage editor abstention. Let $\mathcal{E}^{\text{gate}}$ denote the edit blocks pooled from all retained scripts in the batch. Within each retained script, all valid edit blocks contribute to the loss, including those with negative counterfactual rewards. Reusing the smooth gate from Eq. 5, the edit-level policy-gradient loss $\mathcal{L}_{\text{PG}}^{\text{edt}}$ and the counterfactual edit-reward loss $\mathcal{L}_{\text{edt}}$ are

$$\mathcal{L}_{\text{PG}}^{\text{edt}} = -\frac{1}{N_{\text{edt}}} \sum_{e \in \mathcal{E}^{\text{gate}}} \sum_{t \in e} \phi_\tau\big(\rho_t(e)\big)\, \hat{A}_e, \qquad \mathcal{L}_{\text{edt}} = \mathcal{L}_{\text{PG}}^{\text{edt}} + \beta^{\text{edt}}\, \mathbb{D}_{\text{KL}}^{\text{edt}}, \tag{10}$$

where $N_{\text{edt}} = \sum_{e \in \mathcal{E}^{\text{gate}}} |e|$ is the total number of tokens in the retained edit blocks, $\mathbb{D}_{\text{KL}}^{\text{edt}}$ averages $\hat{D}_t(e)$ over the same tokens, and β^{edt} is the corresponding KL coefficient.

3.6 Correction-Distillation Preference Loss

The counterfactual edit-reward loss directly supervises editing actions and can improve first-pass generation through shared parameters. The correction-distillation preference loss strengthens this transfer by training the generator to prefer reward-improving revisions over drafts.

Revised reports and preference pairs. For each candidate, we jointly apply all reward-positive edits $\mathcal{E}_{i,g}^{+} = \{e \in \mathcal{E}(c_{i,g}) : \Delta R_e > 0\}$ to construct a revised report

$$y_{i,g}^{+} = y_{i,g} \oplus \mathcal{E}_{i,g}^{+}, \;\; \Delta_{i,g} = R\big(y_{i,g}^{+}, y_i^{*}\big) - R\big(y_{i,g}, y_i^{*}\big), \tag{11}$$

where $\oplus$ denotes jointly applying all edits in $\mathcal{E}_{i,g}^{+}$ according to their specified operations.

Because individually beneficial edits may interact when applied jointly, we recheck the combined gain: if $\Delta_{i,g} > 0$, we add the preference pair $(y_{i,g}^{+} \succ y_{i,g})$ to the preference set $\mathcal{P}$. Because the two reports differ only by the applied edits, each pair provides a direct comparison between a draft and a reward-improving revision constructed from the model's own edits.

Reward-weighted preference loss with adaptive target. Let $\ell_\pi(y) = \frac{1}{|y|} \sum_{t=1}^{|y|} \log \pi(y_t \mid y_{<t}, x, p^{\text{gen}})$ be the length-normalized log-likelihood of y under π. We define the reference-adjusted score $h_\theta(y) = \ell_\theta(y) - \ell_{\text{ref}}(y)$ and pairwise margin $\mu_{i,g} = h_\theta(y_{i,g}^{+}) - h_\theta(y_{i,g})$. Inspired by IPO (Azar et al., 2024), we use a squared loss that drives the margin toward a finite target. However, a fixed positive target m may still leave the revised report with a lower length-normalized log-likelihood than the draft when the reference-model score gap exceeds m. We therefore define $d_{i,g} = \ell_{\text{ref}}(y_{i,g}) - \ell_{\text{ref}}(y_{i,g}^{+})$ and use the capped adaptive target

$$\tilde{m}_{i,g} = \max\big(m,\ \min(d_{i,g},\ 3m)\big). \tag{12}$$

When the pairwise margin reaches its target, $\mu_{i,g} = \tilde{m}_{i,g}$. For $d_{i,g} \le 3m$, this gives

$$\ell_\theta(y_{i,g}^{+}) - \ell_\theta(y_{i,g}) = \tilde{m}_{i,g} - d_{i,g} \ge 0. \tag{13}$$

Thus, when $d_{i,g} \le 3m$ and the target margin is reached, the revised report $y_{i,g}^{+}$ has a length-normalized log-likelihood no lower than that of the candidate $y_{i,g}$. The $3m$ cap limits the target margin for pairs with large reference-model score gaps.

We weight each pair by its verified reward gain, so that corrections with larger gains receive proportionally larger weights, yielding the correction-distillation preference loss:

$$\mathcal{L}_{\text{prf}} = \frac{1}{|\mathcal{P}|} \sum_{(i,g) \in \mathcal{P}} \Delta_{i,g} \left(\mu_{i,g} - \tilde{m}_{i,g}\right)^2. \tag{14}$$

Gradient conflict resolution. The candidate report $y_{i,g}$ contributes to both $\mathcal{L}_{\text{PG}}^{\text{rep}}$ and $\mathcal{L}_{\text{prf}}$, which can request opposing updates to its log-likelihood terms. We manage these conflicts using a sign-consistency rule. With respect to the candidate's own score $\ell_\theta(y_{i,g})$, the preference objective favors a decrease when $\mu_{i,g} < \tilde{m}_{i,g}$ and an increase when $\mu_{i,g} > \tilde{m}_{i,g}$. Comparing this direction with the sign of $\hat{A}_{i,g}$ yields two conflict cases:

1. $\hat{A}_{i,g} > 0$ and $\mu_{i,g} < \tilde{m}_{i,g}$: $\mathcal{L}_{\mathrm{prf}}$ would suppress the above-average candidate $y_{i,g}$, whereas $\mathcal{L}_{\mathrm{PG}}^{\mathrm{rep}}$ promotes it. We stop the gradient through $\ell_\theta(y_{i,g})$ in $\mathcal{L}_{\mathrm{prf}}$, so this pair's preference gradient is computed only through the revised-report term.
2. $\hat{A}_{i,g} < 0$ and $\mu_{i,g} > \tilde{m}_{i,g}$: $\mathcal{L}_{\mathrm{prf}}$ would promote the below-average candidate $y_{i,g}$, whereas $\mathcal{L}_{\mathrm{PG}}^{\mathrm{rep}}$ suppresses it. We exclude this pair from $\mathcal{P}$ to remove its potentially large squared-loss contribution and the conflicting preference update. We also mask the candidate's contribution to $\mathcal{L}_{\mathrm{PG}}^{\mathrm{rep}}$ to remove its direct pressure toward further margin growth.

In all other cases, the two objectives do not request opposing direct updates to the candidate's log-likelihood, so no additional masking is applied. This rule manages conflicts over the candidate report during joint optimization of the report-level and preference objectives. The detailed derivation is given in Appendix A.8.

3.7 Overall Objective

The overall objective is

$$\mathcal{L} = \mathcal{L}_{\mathrm{rep}} + \omega_{\mathrm{edt}}\mathcal{L}_{\mathrm{edt}} + \omega_{\mathrm{prf}}\mathcal{L}_{\mathrm{prf}}, \tag{15}$$

where ω_{edt} and ω_{prf} are weighting coefficients. ED-SR uses self-editing only during training. At inference, the trained VLM generates each report in a single pass without invoking the editor.

4 Experiments

4.1 Experimental Settings

Datasets. MIMIC-CXR (Johnson et al., 2019) is a public benchmark of chest X-rays paired with free-text radiology reports, partitioned into 240,197 training, 2,113 validation, and 3,852 test samples. Each sample contains the current study (image(s), report, indication, and clinical history) alongside an optional prior study. **CheXpert Plus** (Chambon et al., 2024) comprises 223,228 radiographs and 187,711 reports. Following the established CXPMRG-Bench (Wang et al., 2025a) split (7:1:2), we use 40,463 training, 5,780 validation, and 11,562 test samples. Each instance includes the current image(s), report, and clinical history, without a longitudinal prior study.

Implementation details. We fine-tune Qwen3-VL-4B-Instruct (Bai et al., 2025) with LoRA (Hu et al., 2022). Stages 1 and 3 are trained on the full training set of each dataset, while Stage 2 uses a 100-sample subset of the corresponding training set. Experiments run on up to 8 H100 (80 GB) GPUs. See Appendix A.9 for further details.

Evaluation metrics. We evaluate generated reports against references using natural language generation (NLG) metrics for linguistic similarity and clinical efficacy (CE) metrics for clinical relevance. NLG metrics include BLEU-1–4 (Papineni et al., 2002), METEOR (Banerjee & Lavie, 2005), and ROUGE-L (Lin, 2004). CE metrics include CheXbert (Smit et al., 2020) micro-averaged F1 and macro-averaged F1 over 14 observations; RadGraph F1 (Delbrouck et al., 2024) evaluates agreement in extracted clinical entities and their relations, whereas RaTEScore (Zhao et al., 2024) uses embedding-based entity matching to assess semantic similarity, accounting for synonymous medical expressions; and GREEN (Ostmeier et al., 2024) measures clinically significant errors using an LLM-based evaluator.

4.2 Results

Baselines. All baseline results are taken from their respective publications. **On MIMIC-CXR**, Table 1 includes 16 recent methods from three main groups: *longitudinal methods* for modeling disease progression (HC-LLM (Liu et al., 2025b), STREAM (Yang et al., 2025), MLRG (Liu et al., 2025a), BiOT (Liu et al., 2026b), PriorRG (Liu et al., 2026a)), *RL-based methods* aligning training with evaluation criteria (ITAdaptor (Ding et al., 2025), LHR-RFL (Yi et al., 2025), MPO (Xiao et al., 2025), HiMed (Wang et al., 2026b), REVA (Guo et al., 2026)), and *self-correction and refinement methods* refining draft reports (B-LLM (Liu et al., 2024), DART (Park et al., 2025), EditGRPO (Zhang et al., 2025b), TIM (Dong et al., 2026), SAT-RRG (Liu et al., 2026c), ESC-RL (Zhou et al., 2026)). These groups reflect primary contributions and may overlap. **On CheXpert Plus**, following the CXPMRG-Bench (Wang et al., 2025a) protocol, we compare against 7 baselines in Table 2:

Table 1: Comparison with state-of-the-art methods on the MIMIC-CXR dataset. B-n: BLEU-n; MTR: METEOR; R-L: ROUGE-L; Mi-F1: the micro-averaged CheXbert F1; Ma-F1: the macro-averaged CheXbert F1; RG-F1: RadGraph F1; RaTE: RaTEScore. Higher is better for all metrics, and the best results are shown in bold.

Method	Year	Natural Language Generation						Clinical Efficacy				
		B-1	B-2	B-3	B-4	MTR	R-L	Mi-F1	Ma-F1	RG-F1	RaTE	GREEN
Longitudinal data												
HC-LLM	2025	0.416	0.276	0.193	0.142	0.162	0.284	0.498	0.357	–	–	–
STREAM	2025	0.437	0.278	0.192	0.139	0.172	0.297	0.478	–	0.223	0.520	–
MLRG	2025	0.411	0.277	0.204	0.158	0.176	0.320	0.505	–	0.291	–	–
BiOT	2026	0.397	0.253	0.174	0.126	0.155	0.285	0.417	–	–	–	–
PriorRG	2026	0.412	0.290	0.220	0.175	0.189	0.324	0.511	–	–	–	–
Reinforcement learning												
ITAdaptor	2025	0.411	0.260	0.187	0.141	0.152	0.314	0.434	–	–	–	–
LHR-RFL	2025	0.400	0.253	0.171	0.120	0.154	0.296	–	0.342	–	–	–
MPO	2025	0.416	0.269	0.191	0.139	0.162	0.309	–	0.353	–	–	–
HiMed	2026	0.292	0.156	0.088	0.052	**0.220**	0.271	–	–	–	0.544	–
REVA	2026	0.424	0.280	0.195	0.143	0.170	0.293	0.506	–	0.246	–	–
Self-correction and refinement												
B-LLM	2024	0.402	0.262	0.180	0.128	0.175	0.291	0.473	–	–	–	–
DART	2025	0.437	0.279	0.191	0.137	0.175	0.310	0.546	–	–	–	–
EditGRPO	2025	–	–	–	–	–	–	0.559	0.367	0.284	0.571	0.383
TIM	2026	0.430	0.265	0.179	0.124	0.185	0.287	0.511	–	–	–	–
SAT-RRG	2026	0.428	0.280	0.196	0.143	0.167	0.303	–	–	0.205	0.540	0.310
ESC-RL	2026	0.487	–	–	0.199	–	0.352	**0.608**	–	0.304	–	0.394
ED-SR		**0.494**	**0.361**	**0.277**	**0.220**	0.213	**0.373**	0.597	**0.388**	**0.362**	**0.629**	**0.414**

MCA-RG (Xing et al., 2025), MambaXray (Wang et al., 2025a), AM-MRG (Wang et al., 2025b), LEAD (Yang et al., 2026), R2GenCSR (Wang et al., 2026a), MedMO (Deria et al., 2026), and DADS (Wu et al., 2026). Appendix A.6 extends the MIMIC-CXR comparison to all 23 baselines.

Comparison with state-of-the-art methods. As shown in Table 1, ED-SR achieves the best scores on 9 of 11 metrics on MIMIC-CXR. Compared with ESC-RL, ED-SR improves BLEU-4 from 0.199 to 0.220 and ROUGE-L from 0.352 to 0.373. It also achieves higher scores on the fine-grained RadGraph F1 metric (0.362 vs. 0.304) and the LLM-based GREEN metric (0.414 vs. 0.394), although its CheXbert micro-F1 is slightly lower (0.597 vs. 0.608). ESC-RL uses GPT-5 at inference, whereas ED-SR supports local inference without external LLM calls, allowing patient data to remain within the deployment environment. ED-SR also achieves the best results on RaTEScore and GREEN, neither of which is included in the reward function, suggesting that its improvements extend beyond the directly optimized metrics. Table 2 further demonstrates ED-SR's performance on CheXpert Plus, where it outperforms recent methods across all metrics, achieving BLEU-4 of 0.149, CheXbert macro-F1 of 0.376 (+0.015 over DADS), RaTEScore of 0.586 (+0.113 over MedMO), and GREEN of 0.310 (+0.064 over R2GenCSR).

4.3 Ablation Study

We evaluate the effects of editor warm-up (Stage 2), report-level optimization, the two proposed losses, and gradient conflict resolution (GCR).

Impact of Stage 2 and $\mathcal{L}_{\text{rep}}$. Table 3 shows comparable performance between Stage 1 and Stage 2 (rows 1–2), suggesting that editor warm-up largely preserves report-generation ability. Report-level optimization with $\mathcal{L}_{\text{rep}}$ (row 3) improves CheXbert micro-F1 from 0.513 to 0.546 and RadGraph F1 from 0.300 to 0.311, while slightly reducing BLEU-4 and ROUGE-L.

Table 2: Comparison with state-of-the-art methods on the CheXpert Plus dataset. B-n: BLEU-n; MTR: METEOR; R-L: ROUGE-L; Ma-F1: the macro-averaged CheXbert F1; RaTE: RaTEScore. Best results are shown in bold.

Method	Year	Natural Language Generation						Clinical Efficacy		
		B-1	B-2	B-3	B-4	MTR	R-L	Ma-F1	RaTE	GREEN
MCA-RG	2025	0.367	0.218	0.149	0.102	0.147	0.266	–	–	–
MambaXray	2025	–	–	–	0.112	0.157	0.276	0.335	–	–
AM-MRG	2025	0.381	0.238	0.159	0.109	0.173	0.282	0.336	–	–
LEAD	2026	–	–	–	–	0.125	0.197	0.275	–	–
R2GenCSR	2026	–	–	–	0.103	0.163	0.272	–	–	0.246
MedMO	2026	–	–	–	–	–	0.236	–	0.473	–
DADS	2026	–	–	–	0.133	–	0.291	0.361	–	–
ED-SR		**0.427**	**0.290**	**0.206**	**0.149**	**0.178**	**0.320**	**0.376**	**0.586**	**0.310**

Table 3: Ablation study on MIMIC-CXR.

Stage	$\mathcal{L}_{\text{rep}}$	$\mathcal{L}_{\text{edt}}$	$\mathcal{L}_{\text{prf}}$	GCR	B-4	R-L	Mi-F1	RG
1					0.196	0.341	0.521	0.300
2					0.196	0.343	0.513	0.300
3	✓				0.194	0.341	0.546	0.311
3	✓	✓			0.211	0.363	0.584	0.348
3	✓	✓	✓		0.030	0.124	0.350	0.043
3	✓	✓	✓	✓	**0.220**	**0.373**	**0.597**	**0.362**



Figure 2: Training reward dynamics with and without GCR; omitting GCR causes collapse.

Effect of $\mathcal{L}_{\text{edt}}$. Adding $\mathcal{L}_{\text{edt}}$ to the report-level baseline (rows 3–4) improves CheXbert micro-F1 from 0.546 to 0.584, RadGraph F1 from 0.311 to 0.348, and BLEU-4 from 0.194 to 0.211. These gains occur without preference distillation or inference-time editing, supporting the transfer of editing supervision to first-pass generation through the parameters shared by the editor and generator.

Effect of $\mathcal{L}_{\text{prf}}$. Adding $\mathcal{L}_{\text{prf}}$ with GCR (row 6) further improves all reported metrics over $\mathcal{L}_{\text{rep}} + \mathcal{L}_{\text{edt}}$ (row 4), including ROUGE-L (0.363 $\rightarrow$ 0.373), CheXbert micro-F1 (0.584 $\rightarrow$ 0.597), and RadGraph F1 (0.348 $\rightarrow$ 0.362). These results show that preference distillation, when combined with GCR, provides additional gains in first-pass report quality. Qualitative comparisons across variants are presented in Appendix A.2.

Effect of GCR. Without GCR, adding $\mathcal{L}_{\text{prf}}$ leads to performance collapse across all four metrics (row 5). Figure 2 shows a sharp reward drop around step 8,000. With GCR, training remains stable and the model achieves the best results (row 6). GCR manages opposing direct updates from $\mathcal{L}_{\text{PG}}^{\text{rep}}$ and $\mathcal{L}_{\text{prf}}$ on shared candidate reports using sign-based masking. Further collapse analysis and training dynamics are provided in Appendix A.3 and Appendix A.4, respectively.

Computational overhead. ED-SR adds training cost for editor rollouts, edit reward computation, and preference optimization, while retaining standard single-pass inference (see Appendix A.5).

5 Conclusion

We presented ED-SR, a training-time self-refinement framework for chest X-ray report generation using a single VLM as both generator and editor. Counterfactual edit rewards assign fine-grained credit to self-generated edits, while preference distillation with gradient conflict resolution transfers reward-improving revisions into the generation policy. ED-SR achieves state-of-the-art results across NLG and CE metrics on MIMIC-CXR and CheXpert Plus while retaining single-pass inference.

AI USE STATEMENT

In this work, we used generative AI tools to generate synthetic data sets. We have not used generative AI tools to help develop theoretical models or conceptual frameworks, formulate mathematical claims, provide critical ingredients for proving mathematical claims, assist in the writing of proofs, propose or refine hypotheses, design or provide feedback on research methodology or experiments, implement methods, assist with translation, clean and reformat dataset, support qualitative and thematic data analysis, or interpret results.

Additionally, we used generative AI tools to edit software code and polish the paper to improve readability.

We have reviewed all AI-assisted work. (i) We used AI tools to generate 100 editor warm-up samples for each of the two datasets. The generated edit scripts were reviewed by a radiologist. (ii) We used AI tools to help write specific functional code blocks for this project. The LLM-generated code was verified and tested for correctness by the authors. The overall architectural design, system structure, and integration of these components were entirely planned and executed by the authors. (iii) We wrote the manuscript and used AI tools strictly to check grammar and suggest improvements to sentences. All suggested changes were manually reviewed, edited, and rewritten by the authors. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This study utilizes real-world patient data from the MIMIC-CXR and CheXpert Plus datasets. These datasets have been de-identified and released under controlled access for research purposes; consequently, the risk of privacy leakage is minimal. We adhere to the respective data use agreements and employ the data exclusively for the development and evaluation of automated CXR-RG models. Regarding the generation of the editor warm-up corpus using GPT-5.6 Sol, the inputs provided to the model consisted solely of reports generated by the Stage 1 model and reference reports; no images or clinical history data were included.

REPRODUCIBILITY STATEMENT

Implementation details for ED-SR, including image preprocessing, training configurations, and inference settings, are provided in Appendix A.9. Hyperparameters are summarized in Table 7. The appendix also provides the assumptions and proofs for our theoretical analysis. Our code and the editor warm-up annotation prompt will be released upon acceptance.

REFERENCES

Afra Feyza Akyürek, Ekin Akyürek, Ashwin Kalyan, Peter Clark, Derry Tanti Wijaya, and Niket Tandon. Rl4f: Generating natural language feedback with reinforcement learning for repairing model outputs. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 7716–7733, 2023.

Mohammad Gheshlaghi Azar, Zhaohan Daniel Guo, Bilal Piot, Remi Munos, Mark Rowland, Michal Valko, and Daniele Calandriello. A general theoretical paradigm to understand learning from human preferences. In *International Conference on Artificial Intelligence and Statistics*, pp. 4447–4455. PMLR, 2024.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*, 2025.

Satanjeev Banerjee and Alon Lavie. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, pp. 65–72, 2005.

Pierre Chambon, Jean-Benoit Delbrouck, Thomas Sounack, Shih-Cheng Huang, Zhihong Chen, Maya Varma, Steven QH Truong, Chu The Chuong, and Curtis P Langlotz. Chexpert plus:

Augmenting a large chest x-ray dataset with text radiology reports, patient demographics and additional image formats. *arXiv preprint arXiv:2405.19538*, 2024.

Jean-Benoit Delbrouck, Pierre Chambon, Christian Bluethgen, Emily Tsai, Omar Almusa, and Curtis Langlotz. Improving the factual correctness of radiology report generation with semantic rewards. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pp. 4348–4360, 2022.

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.

Ankan Deria, Komal Kumar, Adinath Madhavrao Dukre, Eran Segal, Salman Khan, and Imran Razzak. Medmo: Grounding and understanding multimodal large language model for medical images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5092–5103, June 2026.

Shuaipeng Ding, Mengnan Fan, Mingyong Li, and Chao Wang. Itadaptor: Image-tag adapter framework with knowledge enhancement for radiology report generation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 353–362. Springer, 2025.

Yiheng Dong, Yi Lin, Shilong Huang, Xiyan Yang, and Xin Yang. Tim: Temporal decoupling with iterative mutual-refinement model for longitudinal radiology report generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6951–6961, 2026.

Stanislav Fort, PaweĹ Nowak, Stanislaw Jastrzebski, Srini Narayanan, et al. Stiffness: A new perspective on generalization in neural networks. *arXiv preprint arXiv:1901.09491*, 2019.

Chang Gao, Chujie Zheng, Xiong-Hui Chen, Kai Dang, Shixuan Liu, Bowen Yu, An Yang, Shuai Bai, Jingren Zhou, and Junyang Lin. Soft adaptive policy optimization. *arXiv preprint arXiv:2511.20347*, 2025.

Li Guo, Anas M. Tahir, and Z. Jane Wang. Reva-po: Stabilizing reinforcement learning for chest x-ray report generation. In *Computer Vision – ECCV 2026*, pp. 565–583. Springer Nature Switzerland, 2026.

Jiayi He, Hehai Lin, Qingyun Wang, Yi R Fung, and Heng Ji. Self-correction is more than refinement: A learning framework for visual and language reasoning tasks. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 6405–6421, 2025.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. URL `https://openreview.net/forum?id=nZeVKeeFYf9`.

Jie Huang, Xinyun Chen, Swaroop Mishra, Huaixiu Steven Zheng, Adams Yu, Xinying Song, and Denny Zhou. Large language models cannot self-correct reasoning yet. In *International conference on learning representations*, volume 2024, pp. 32808–32824, 2024.

Alistair EW Johnson, Tom J Pollard, Nathaniel R Greenbaum, Matthew P Lungren, Chih-ying Deng, Yifan Peng, Zhiyong Lu, Roger G Mark, Seth J Berkowitz, and Steven Horng. Mimic-cxr-jpg, a large publicly available database of labeled chest radiographs. *arXiv preprint arXiv:1901.07042*, 2019.

Ryo Kamoi, Yusen Zhang, Nan Zhang, Jiawei Han, and Rui Zhang. When can llms actually correct their own mistakes? a critical survey of self-correction of llms. *Transactions of the Association for Computational Linguistics*, 12:1417–1440, 2024.

Aviral Kumar, Vincent Zhuang, Rishabh Agarwal, Yi Su, John D Co-Reyes, Avi Singh, Kate Baumli, Shariq Iqbal, Colton Bishop, Rebecca Roelofs, Lei M Zhang, Kay McKinney, Disha Shrivastava, Cosmin Paduraru, George Tucker, Doina Precup, Feryal Behbahani, and Aleksandra Faust. Training language models to self-correct via reinforcement learning. In *The Thirteenth International*

Conference on Learning Representations, 2025. URL https://openreview.net/forum?id=CjwERcAU7w.

Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pp. 74–81, 2004.

Chang Liu, Yuanhe Tian, Weidong Chen, Yan Song, and Yongdong Zhang. Bootstrapping large language models for radiology report generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 18635–18643, 2024.

Kang Liu, Zhuoqi Ma, Xiaolu Kang, Yunan Li, Kun Xie, Zhicheng Jiao, and Qiguang Miao. Enhanced contrastive learning with multi-view longitudinal data for chest x-ray report generation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 10348–10359, 2025a.

Kang Liu, Zhuoqi Ma, Zikang Fang, Yunan Li, Kun Xie, and Qiguang Miao. Priorrg: Prior-guided contrastive pre-training and coarse-to-fine decoding for chest x-ray report generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 7206–7214, 2026a.

Tengfei Liu, Jiapu Wang, Yongli Hu, Mingjie Li, Junfei Yi, Xiaojun Chang, Junbin Gao, and Baocai Yin. Hc-llm: Historical-constrained large language models for radiology report generation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 39, pp. 5595–5603, 2025b.

Tengfei Liu, Yijian Fan, Boyue Wang, Yongli Hu, Mingjie Li, Jinghua Li, Junbin Gao, Xiaojun Chang, Zhihui Li, and Baocai Yin. Biotprompt: Bidirectional optimal transport guided prompting for disease evolution-aware radiology report generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13755–13765, 2026b.

Yunyi Liu, Yingshu Li, Tong Chen, Lingqiao Liu, Lei Wang, and Luping Zhou. Sat-rrg: Llm-guided self-adaptive training for radiology report generation with token-level push-pull optimization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 35363–35372, 2026c.

Zichen Liu, Changyu Chen, Wenjun Li, Penghui Qi, Tianyu Pang, Chao Du, Wee Sun Lee, and Min Lin. Understanding r1-zero-like training: A critical perspective. *arXiv preprint arXiv:2503.20783*, 2025c.

Ilya Loshchilov, Frank Hutter, et al. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

Yasuhide Miura, Yuhao Zhang, Emily Tsai, Curtis Langlotz, and Dan Jurafsky. Improving factual completeness and consistency of image-to-text radiology report generation. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 5288–5304, 2021.

OpenAI. GPT-5.6: Frontier Intelligence That Scales with Your Ambition. https://openai.com/index/gpt-5-6/, July 2026.

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. GREEN: Generative radiology report evaluation and error notation. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 374–390, November 2024.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pp. 311–318, 2002.

Sang-Jun Park, Keun-Soo Heo, Dong-Hee Shin, Young-Han Son, Ji-Hye Oh, and Tae-Eui Kam. Dart: Disease-aware image-text alignment and self-correcting re-alignment for trustworthy radiology report generation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 15580–15589, 2025.

Yuxiao Qu, Tianjun Zhang, Naman Garg, and Aviral Kumar. Recursive introspection: Teaching language model agents how to self-improve. *Advances in Neural Information Processing Systems*, 37:55249–55285, 2024.

Karthik Abinav Sankararaman, Soham De, Zheng Xu, W Ronny Huang, and Tom Goldstein. The impact of neural network overparameterization on gradient confusion and stochastic gradient descent. In *International conference on machine learning*, pp. 8469–8479. PMLR, 2020.

John Schulman. Approximating KL divergence. `http://joschu.net/blog/kl-approx.html`, 2020. Blog post.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Akshay Smit, Saahil Jain, Pranav Rajpurkar, Anuj Pareek, Andrew Y Ng, and Matthew 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 (EMNLP)*, pp. 1500–1519, 2020.

Xiaoshuai Song, Yanan Wu, Weixun Wang, Jiaheng Liu, Wenbo Su, and Bo Zheng. Progco: Program helps self-correction of large language models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 944–959, 2025.

Fahim Tajwar, Anikait Singh, Archit Sharma, Rafael Rafailov, Jeff Schneider, Tengyang Xie, Stefano Ermon, Chelsea Finn, and Aviral Kumar. Preference fine-tuning of llms should leverage suboptimal, on-policy data. *arXiv preprint arXiv:2404.14367*, 2024.

Xiao Wang, Fuling Wang, Yuehang Li, Qingchuan Ma, Shiao Wang, Bo Jiang, and Jin Tang. Cxpmrg-bench: Pre-training and benchmarking for x-ray medical report generation on chexpert plus dataset. In *Proceedings of the computer vision and pattern recognition conference*, pp. 5123–5133, 2025a.

Xiao Wang, Fuling Wang, Haowen Wang, Bo Jiang, Chuanfu Li, Yaowei Wang, Yonghong Tian, and Jin Tang. Activating associative disease-aware vision token memory for llm-based x-ray report generation. *IEEE Transactions on Medical Imaging*, 2025b.

Xiao Wang, Yuehang Li, Fuling Wang, Shiao Wang, Chuanfu Li, and Bo Jiang. R2gencsr: Mining contextual and residual information for llms-based radiology report generation. *IEEE Journal of Biomedical and Health Informatics*, 2026a.

Yuan Wang, Shujian Gao, Jiaxiang Liu, Songtao Jiang, Xia Haoxiang, Xiaotian Zhang, Zhaolu Kang, Yemin Wang, and Zuozhu Liu. Beyond n-grams: A hierarchical reward learning framework for clinically-aware medical report generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 33719–33727, 2026b.

Puzhen Wu, Hexin Dong, Yi Lin, Yihao Ding, and Yifan Peng. A disease-aware dual-stage framework for chest x-ray report generation. In *Proceedings of the AAAI conference on artificial intelligence*, pp. 33953–33961, 2026.

Ting Xiao, Lei Shi, Peng Liu, Zhe Wang, and Chenjia Bai. Radiology report generation via multi-objective preference optimization. In *Proceedings of the AAAI conference on artificial intelligence*, volume 39, pp. 8664–8672, 2025.

Qilong Xing, Zikai Song, Youjia Zhang, Na Feng, Junqing Yu, and Wei Yang. Mca-rg: Enhancing llms with medical concept alignment for radiology report generation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 380–390. Springer, 2025.

Wenda Xu, Guanglei Zhu, Xuandong Zhao, Liangming Pan, Lei Li, and William Wang. Pride and prejudice: Llm amplifies self-bias in self-refinement. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 15474–15492, 2024.

Ruixiao Yang, Yuanhe Tian, Xu Yang, Huiqi Li, and Yan Song. Lead: Layer-wise expert-aligned decoding for faithful radiology report generation. *arXiv preprint arXiv:2602.04617*, 2026.

Yan Yang, Xiaoxing You, Ke Zhang, Zhenqi Fu, Xianyun Wang, Jiajun Ding, Jiamei Sun, Zhou Yu, Qingming Huang, Weidong Han, et al. Spatio-temporal and retrieval-augmented modelling for chest x-ray report generation. *IEEE Transactions on Medical Imaging*, 2025.

Xiulong Yi, You Fu, Jianzhi Yu, Ruiqing Liu, Hao Zhang, and Rong Hua. Lhr-rfl: Linear hybrid-reward-based reinforced focal learning for automatic radiology report generation. *IEEE Transactions on Medical Imaging*, 44(3):1494–1504, 2025.

Di Zhang, Jingdi Lei, Junxian Li, Xunzhi Wang, Yujie Liu, Zonglin Yang, Jiatong Li, Weida Wang, Suorong Yang, Jianbo Wu, et al. Critic-v: Vlm critics help catch vlm errors in multimodal reasoning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9050–9061, 2025a.

Kai Zhang, Christopher Malon, Lichao Sun, and Martin Renqiang Min. Editgrpo: Reinforcement learning with post-rollout edits for clinically accurate chest x-ray report generation. In *Proceedings of the 14th International Joint Conference on Natural Language Processing and the 4th Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics*, pp. 304–316, 2025b.

Qingjie Zhang, Di Wang, Haoting Qian, Yiming Li, Tianwei Zhang, Minlie Huang, Ke Xu, Hewu Li, Liu Yan, and Han Qiu. Understanding the dark side of llms' intrinsic self-correction. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 27066–27101, 2025c.

Weike Zhao, Chaoyi Wu, Xiaoman Zhang, Ya Zhang, Yanfeng Wang, and Weidi Xie. Ratescore: A metric for radiology report generation. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 15004–15019, 2024.

Qin Zhou, Guoyan Liang, Qianyi Yang, Jingyuan Chen, Sai Wu, Chang Yao, and Zhe Wang. Enhancing reinforcement learning for radiology report generation with evidence-aware rewards and self-correcting preference learning. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 37044–37056, 2026.

Reference: Frontal and lateral views of the chest. Somewhat linear opacity is seen at the right lung base. Retrocardiac mixed lucency and density is most suggestive of a hiatal hernia. The lungs are otherwise clear. There is no effusion. Cardiomediastinal silhouette is within normal limits. Accentuated thoracic kyphosis is noted. Osseous and soft tissue structures are otherwise unremarkable.

Generated: **S1:** AP and lateral views of the chest are compared to previous exam from. **S2:** ~~Lower lung volumes seen on the current exam.~~ **S3:** ~~There is secondary crowding of the bronchovascular markings.~~ **S4:** Blunting of posterior costophrenic angle could be due to small effusions. **S5:** Cardiac silhouette is enlarged but stable in configuration given lower inspiratory effort. **S6:** Osseous and soft tissue structures are unchanged, noting significant degenerative changes at the glenohumeral joints bilaterally.

Structured Edit Script:

...

<edit sid="S2,S3">
Action: delete
Rationale: Low lung volumes and bronchovascular crowding are not supported.
</edit>
<edit sid="S6">
Action: revise
Rationale: Thoracic kyphosis is present, but significant shoulder degeneration is not supported.
Revised: Accentuated thoracic kyphosis is noted; osseous and soft tissue structures are otherwise unremarkable.
</edit>
<edit sid="ADD_1">
Action: add
Rationale: A linear right basilar opacity is missing.
Revised: Somewhat linear opacity is seen at the right lung base.
</edit>

...

Figure 3: Example of a structured edit script. Given the reference and generated reports, the script identifies and corrects different types of generation errors through three edit operations: DELETE removes unsupported findings (red, struck through), REVISE corrects partially supported or inaccurate statements (orange), and ADD inserts missing findings (teal). Each edit block specifies its target, rationale, and, when applicable, the revised or added sentence.

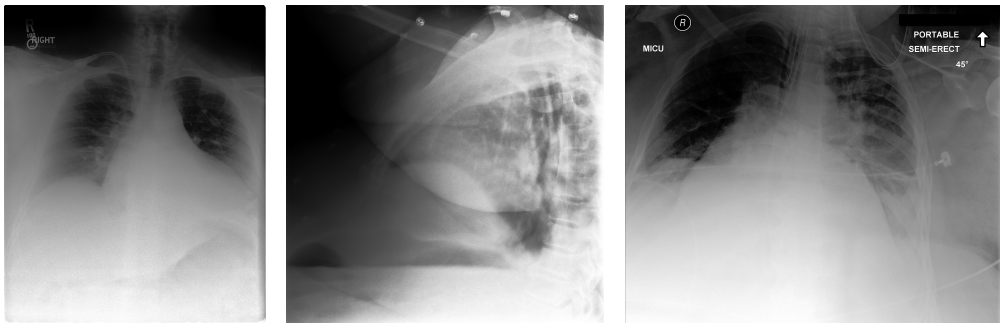
A Appendix

Appendix Overview

A.1 Edit Script Example 15
A.2 Qualitative Analysis 16
A.3 Analysis of Policy Collapse Under Gradient Conflict 16
A.4 Training Statistics During Stage 3 17
A.5 Computational Overhead 18
A.6 Extended Comparisons on MIMIC-CXR 18
A.7 Comparison of Inference Strategies 19
A.8 Derivations for the Correction-Distillation Preference Loss (§3.6) 20
A.9 Implementation Details 25

A.1 Edit Script Example

Figure 3 shows a concrete example of the structured edit script used in Stage 2. In this case, the generated report contains three error types: unsupported findings, partially correct but inaccurate statements, and missing findings. The edit script converts these errors into a sequence of edit blocks.



current – frontal current – lateral prior – frontal

Reference: As compared to the previous radiograph, there is no relevant change. Low lung volumes and moderate cardiomegaly without evidence of pulmonary edema or pleural effusions. Moderate retrocardiac atelectasis. No evidence of pneumonia.

Baseline ($\mathcal{L}_{\text{rep}}$): AP and lateral views of the chest were reviewed. The cardiomediastinal and hilar contours are stable. There is no pleural effusion or pneumothorax. Lungs are well expanded and clear$^{\times}$. There is no focal consolidation concerning for pneumonia. Pulmonary vasculature is within normal limits. The right internal jugular central venous catheter ends in the low SVC. Left PICC ends in the upper SVC.
$\times$ omits: prior comparison; cardiomegaly.

ED-SR ($\mathcal{L}_{\text{rep}}+\mathcal{L}_{\text{edt}}+\mathcal{L}_{\text{prf}}$): PA and lateral views of the chest were reviewed and compared to the prior studies. Lung volumes are low and there is bibasilar atelectasis. Moderate cardiomegaly is unchanged. The mediastinal contour is normal. There is no pulmonary edema, pleural effusion or pneumothorax. A right hemodialysis catheter ends in the right atrium. Left PICC ends in the mid SVC.

error (baseline) revise (ours) add (ours) GT evidence

Figure 4: A qualitative comparison example. The baseline hallucinates clear lungs (red) and omits key findings; ED-SR revises the error (orange) and adds the omitted findings (teal).

More specifically, sentences $S2$ and $S3$ are removed with a single DELETE block because both claims, namely low lung volumes and secondary bronchovascular crowding, are not supported by the reference report. Sentence $S6$ is handled by a REVISE block. Although it attempts to describe osseous and soft tissue findings, it introduces an unsupported statement about significant bilateral glenohumeral degenerative changes. The revised sentence includes the supported finding of thoracic kyphosis and restores the correct normal statement for the remaining osseous and soft tissue structures. Finally, an ADD block appends the missing finding that a somewhat linear opacity is seen at the right lung base.

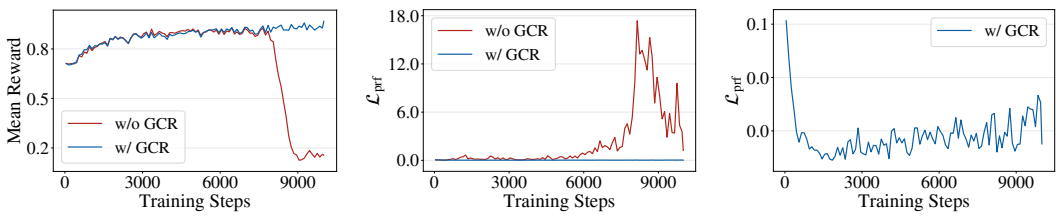
This example illustrates two useful properties of our edit interface. First, each edit block has a clear local target together with a short rationale, making the correction process explicit. Second, because DELETE and REVISE refer to sentence identifiers and ADD always appends a new sentence to the end of the report, the entire script can be deterministically applied to obtain a corrected report. These properties are later used in Stage 3, where edit blocks can be scored in isolation.

A.2 Qualitative Analysis

Figure 4 illustrates how training-time self-editing and preference distillation improve first-pass report generation. The $\mathcal{L}_{\text{rep}}$ baseline produces a fluent report containing an incorrect statement and clinically relevant omissions. It describes the lungs as well expanded and clear, inconsistent with the low lung volumes and retrocardiac atelectasis in the reference. It also omits cardiomegaly and a comparison with the prior examination, despite receiving the prior image as input. With the additional $\mathcal{L}_{\text{edt}}$ and $\mathcal{L}_{\text{prf}}$ objectives, ED-SR generates a report describing low lung volumes and atelectasis, while also reporting cardiomegaly and comparing the current and prior studies. This example complements the quantitative results in Table 3, illustrating improved consistency with the reference and coverage of clinically relevant findings without inference-time editing.

A.3 Analysis of Policy Collapse Under Gradient Conflict

As illustrated in Figure 5a, training without gradient conflict resolution (GCR) triggers a catastrophic policy collapse around 8,000 steps on MIMIC-CXR. This failure is accompanied by a sharp explo-



(a) Mean reward dynamics. (b) Preference loss $\mathcal{L}_{\text{prf}}$. (c) $\mathcal{L}_{\text{prf}}$ (zoomed view with GCR).

Figure 5: Training dynamics with and without gradient conflict resolution (GCR): (a) Mean reward curves showing policy collapse around step 8,000 when GCR is omitted; (b) Catastrophic inflation of the preference loss $\mathcal{L}_{\text{prf}}$ without GCR; (c) Stable convergence of $\mathcal{L}_{\text{prf}}$ maintained under GCR.

sion in the correction-distillation preference loss $\mathcal{L}_{\text{prf}}$ (Figure 5b). A closer look at the gradient interactions between the report-level policy gradient loss $\mathcal{L}_{\text{PG}}^{\text{rep}}$ and $\mathcal{L}_{\text{prf}}$ reveals two distinct conflict regimes:

- Case 1 ($\hat{A}_{i,g} > 0$ and $\mu_{i,g} < \tilde{m}_{i,g}$): The candidate report $y_{i,g}$ has an above-average reward within its group, but the condition $h_\theta(y_{i,g}^+) < h_\theta(y_{i,g}) + \tilde{m}_{i,g}$ indicates that the reference-adjusted margin has not reached its target. Here, $\mathcal{L}_{\text{PG}}^{\text{rep}}$ seeks to reinforce $y_{i,g}$ due to its positive advantage, whereas $\mathcal{L}_{\text{prf}}$ pushes to suppress $y_{i,g}$ in favor of $y_{i,g}^+$. This conflict may weaken the reinforcement of high-reward candidates and slow convergence.
- Case 2 ($\hat{A}_{i,g} < 0$ and $\mu_{i,g} > \tilde{m}_{i,g}$): The sampled report $y_{i,g}$ has a below-average reward within its group, while the reference-adjusted margin $\mu_{i,g}$ exceeds its target $\tilde{m}_{i,g}$. Because $\hat{A}_{i,g} < 0$, $\mathcal{L}_{\text{PG}}^{\text{rep}}$ favors decreasing the candidate's log-likelihood, which can push $\mu_{i,g}$ further above its target $\tilde{m}_{i,g}$. As $\mu_{i,g} - \tilde{m}_{i,g}$ grows, the squared preference penalty increases quadratically, potentially destabilizing joint optimization. Moreover, because $y_{i,g}^+$ retains most of the content in $y_{i,g}$, continued suppression of the candidate also penalizes content shared with the revised report.

Although Case 2 accounts for only around 5% of preference pairs throughout training (Figure 6b), these conflicts may contribute to the observed collapse. Implementing GCR suppresses these destabilizing updates and restores stable training.

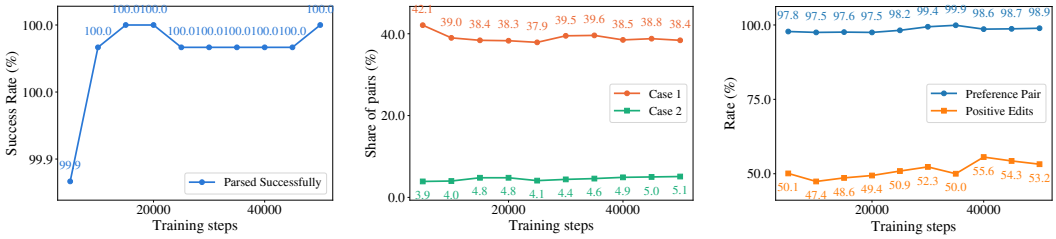
A.4 Training Statistics During Stage 3

To verify that all components functioned as intended, we monitored several statistics throughout Stage 3 RL training on the MIMIC-CXR dataset. Each data point in Figures 6a–6c is averaged over the preceding 5,000 training steps.

Figure 6a shows the rate at which model-generated edit blocks were successfully parsed. Parsing failures include malformed or incomplete edit XML, as well as references to sentence identifiers that do not exist in the candidate report (e.g., an edit block specifying $\text{sid} = 9$ for a report containing only 8 sentences). The parse success rate remains above 99.9% throughout training, indicating that the Stage 2 warm-up on only 100 samples was sufficient for the model to learn the structured edit-script format. The few remaining failures were mainly DELETE actions referencing nonexistent sentence identifiers, which we attribute to hallucination.

Figure 6b reports the prevalence of the two gradient-conflict cases defined in §3.6 across constructed preference pairs. Case 1 accounts for approximately 40% of pairs, whereas Case 2 remains rare at roughly 5%. Case 1 occurs when a candidate has an above-average group reward but the reference-adjusted margin remains below its target. Case 2 occurs when a candidate has a below-average group reward and the reference-adjusted margin exceeds its target. Although Case 2 occurs infrequently, neither rate is negligible. Both cases involve opposing direct updates to the candidate's log-likelihood, which can hinder joint optimization and motivate the use of GCR.

Figure 6c presents two key statistics: (1) Positive edits: the proportion of edit blocks e whose counterfactual edit-reward is positive ($\Delta R_e > 0$, Eq. 8); and (2) Preference pairs: among candidate



(a) Parse success rate. (b) Rates of gradient-conflict cases. (c) Edit and preference rates.

Figure 6: Training statistics and self-correction dynamics: (a) Parse success rate of model-generated edit blocks; (b) Frequency of the two gradient-conflict cases (§3.6) over constructed pairs; (c) Positive-edit rate and valid preference-pair construction rate.

Table 4: Training overhead and performance ablation of ED-SR on the MIMIC-CXR dataset. Baseline is trained with $\mathcal{L}_{\text{rep}}$ alone; ED-SR uses the full objective $\mathcal{L}_{\text{rep}} + \mathcal{L}_{\text{edt}} + \mathcal{L}_{\text{prf}}$.

Method	**Training Cost**		**Performance Metrics**			
	Memory/GPU (MiB)	Time/Step (s)	BLEU-4	ROUGE-L	Micro-F1	RadGraph
Baseline ($\mathcal{L}_{\text{rep}}$)	47,881	3.3	0.194	0.341	0.546	0.311
ED-SR ($\mathcal{L}_{\text{rep}}+\mathcal{L}_{\text{edt}}+\mathcal{L}_{\text{prf}}$)	55,638	11.2	0.220	0.373	0.597	0.362
Overhead / Gain (%)	+16.2	+239.4	+13.4	+9.4	+9.3	+16.4

reports with at least one positive edit, the proportion for which jointly applying all positive edits increases the report reward ($\Delta_{i,g} > 0$, Eq. 11). During training-time self-editing, approximately 50% of edit blocks have positive counterfactual rewards, and nearly 100% of candidate reports containing positive edits yield reward-improving preference pairs. Negative-reward edits in retained scripts also supervise the model through $\mathcal{L}_{\text{edt}}$, discouraging harmful modifications. Although ED-SR increases per-step training time to approximately 3.4 times that of the baseline (Table 4), retained edits and reward-improving revised reports provide supervision for $\mathcal{L}_{\text{edt}}$ and $\mathcal{L}_{\text{prf}}$, respectively.

A.5 Computational Overhead

Table 4 summarizes the training cost and performance trade-offs of ED-SR relative to the baseline ($\mathcal{L}_{\text{rep}}$ alone). Optimizing the full objective ($\mathcal{L}_{\text{rep}}+\mathcal{L}_{\text{edt}}+\mathcal{L}_{\text{prf}}$) increases peak memory from 47,881 to 55,638 MiB (+16.2%) and per-step training time from 3.3 to 11.2 s (+239.4%), primarily due to the additional rollouts and reward computations required by the reinforcement learning objectives.

This additional cost is confined to training. ED-SR improves first-pass generation through shared parameters and preference distillation. At deployment, it uses the same model architecture and single-pass generation procedure as the baseline, without additional editing passes or extra components.

As shown in Table 4, this training investment yields substantial downstream improvements across both NLG (BLEU-4 +13.4%, ROUGE-L +9.4%) and CE metrics (CheXbert micro-F1 +9.3%, RadGraph F1 +16.4%). Since training costs can be amortized over many inference requests, these results suggest a favorable trade-off between additional training computation and report-generation performance.

A.6 Extended Comparisons on MIMIC-CXR

Table 5 extends our main MIMIC-CXR comparison (Table 1) with 7 additional baselines under *Cross-Modal Alignment & Representation*. ED-SR achieves the best scores on 9 of 11 metrics on MIMIC-CXR.

Table 5: Comprehensive comparison with state-of-the-art methods on the MIMIC-CXR dataset. B-n: BLEU-n; MTR: METEOR; R-L: ROUGE-L; Mi-F1: the micro-averaged CheXbert F1; Ma-F1: the macro-averaged CheXbert F1; RG-F1: RadGraph F1; RaTE: RaTEScore. Higher is better for all metrics, and the best results are shown in bold.

Method	Year	Natural Language Generation						Clinical Efficacy				
		B-1	B-2	B-3	B-4	MTR	R-L	Mi-F1	Ma-F1	RG-F1	RaTE	GREEN
Longitudinal data												
HC-LLM	2025	0.416	0.276	0.193	0.142	0.162	0.284	0.498	0.357	–	–	–
STREAM	2025	0.437	0.278	0.192	0.139	0.172	0.297	0.478	–	0.223	0.520	–
MLRG	2025	0.411	0.277	0.204	0.158	0.176	0.320	0.505	–	0.291	–	–
BiOT	2026	0.397	0.253	0.174	0.126	0.155	0.285	0.417	–	–	–	–
PriorRG	2026	0.412	0.290	0.220	0.175	0.189	0.324	0.511	–	–	–	–
Reinforcement learning												
ITAdaptor	2025	0.411	0.260	0.187	0.141	0.152	0.314	0.434	–	–	–	–
LHR-RFL	2025	0.400	0.253	0.171	0.120	0.154	0.296	–	0.342	–	–	–
MPO	2025	0.416	0.269	0.191	0.139	0.162	0.309	–	0.353	–	–	–
HiMed	2026	0.292	0.156	0.088	0.052	**0.220**	0.271	–	–	–	0.544	–
REVA	2026	0.424	0.280	0.195	0.143	0.170	0.293	0.506	–	0.246	–	–
Self-correction and refinement												
B-LLM	2024	0.402	0.262	0.180	0.128	0.175	0.291	0.473	–	–	–	–
DART	2025	0.437	0.279	0.191	0.137	0.175	0.310	0.546	–	–	–	–
EditGRPO	2025	–	–	–	–	–	–	0.559	0.367	0.284	0.571	0.383
TIM	2026	0.430	0.265	0.179	0.124	0.185	0.287	0.511	–	–	–	–
SAT-RRG	2026	0.428	0.280	0.196	0.143	0.167	0.303	–	–	0.205	0.540	0.310
ESC-RL	2026	0.487	–	–	0.199	–	0.352	**0.608**	–	0.304	–	0.394
Cross-modal alignment & representation												
MCA-RG	2025	0.411	0.256	0.175	0.128	0.163	0.300	0.408	0.335	–	–	–
MambaXray	2025	0.422	0.268	0.184	0.133	0.167	0.289	0.505	–	–	–	–
AM-MRG	2025	0.426	0.271	0.187	0.136	0.174	0.291	0.484	–	–	–	0.330
LEAD	2026	–	–	–	–	0.120	0.208	–	0.247	–	–	–
R2GenCSR	2026	0.420	0.268	0.186	0.136	0.167	0.291	0.484	–	–	–	0.329
MedMO	2026	–	–	–	–	–	0.317	–	–	–	0.571	–
DADS	2026	0.428	0.272	0.187	0.131	–	0.291	–	–	–	–	–
ED-SR		**0.494**	**0.361**	**0.277**	**0.220**	0.213	**0.373**	0.597	**0.388**	**0.362**	**0.629**	**0.414**

A.7 Comparison of Inference Strategies

We further investigate whether self-correction should be applied at inference time. Table 6 compares three strategies: (1) *First-pass*, which directly outputs the model's initial generation; (2) *Self-corrected*, which unconditionally applies all edits produced in the second pass; and (3) *Oracle-filtered*, which evaluates each proposed edit using the reference-based composite reward and applies edits with positive reward changes. This strategy requires reference reports and is included only for analysis (shown in gray).

Unconditionally applying all edits degrades performance across all metrics (row 1 vs. row 2). The drop is sharpest in micro-F1 ($0.597 \rightarrow 0.563$). This is consistent with the training statistics in Figure 6c, where approximately half of generated edits have positive counterfactual rewards. One possible explanation is an asymmetry in edit impact: an erroneous DELETE action can remove a clinically important finding and outweigh the gains from other edits.

Conversely, *Oracle-filtered* outperforms *First-pass* on 5 of the 6 metrics (row 1 vs. row 3), yielding substantial gains in clinical efficacy (Micro-F1: $0.597 \rightarrow 0.641$) and RadGraph F1 ($0.362 \rightarrow 0.372$), while traditional NLG metrics remain largely unchanged. These results show that self-generated edits can improve report quality when filtered using reference-based rewards. Because this reward-guided filtering requires reference reports, we adopt *First-pass* for our main results and leave reliable reference-free edit selection to future work.

Table 6: Comparison of inference strategies on the MIMIC-CXR dataset. Micro-F1: CheXbert micro-averaged F1. The Oracle-filtered row (gray) uses reference reports to select reward-improving edits and is included only for analysis. The best results are bolded, and the second-best are underlined.

Inference	BLEU-2	BLEU-4	METEOR	ROUGE-L	Micro-F1	RadGraph
First-pass	<u>0.361</u>	<u>0.220</u>	**0.213**	<u>0.373</u>	<u>0.597</u>	<u>0.362</u>
Self-corrected	0.345	0.211	0.203	0.372	0.563	0.352
Oracle-filtered	**0.363**	**0.222**	<u>0.212</u>	**0.380**	**0.641**	**0.372**

A.8 Derivations for the Correction-Distillation Preference Loss (§3.6)

A.8.1 Setup and notation

We drop the (study, candidate) indices (i, g) whenever they are clear, writing $y = y_{i,g}$ for the candidate report (loser), $y^+ = y^+_{i,g}$ for the revised report (winner), and $\mu = \mu_{i,g}$, $\tilde{m} = \tilde{m}_{i,g}$, $d = d_{i,g}$, $\Delta = \Delta_{i,g}$, $\hat{A} = \hat{A}_{i,g}$. For a report y generated under the prompt p^{gen}, denote its token conditionals and their parameter gradients by

$$z_t(y) = \log \pi_\theta\big(y_t \mid y_{<t}, x, p^{\mathrm{gen}}\big), \quad g_t(y) = \nabla_\theta z_t(y), \tag{16}$$

so that $\ell_\theta(y) = \frac{1}{|y|}\sum_t z_t(y)$ and $\nabla_\theta \ell_\theta(y) = \frac{1}{|y|}\sum_t g_t(y)$. All reference quantities $\ell_{\mathrm{ref}}(\cdot)$, d, and $\tilde{m}$ are constants with respect to θ, and every pair in $\mathcal{P}$ satisfies $\Delta > 0$ by construction. The following analysis isolates the contributions of $\mathcal{L}^{\mathrm{rep}}_{\mathrm{PG}}$ and $\mathcal{L}_{\mathrm{prf}}$. The KL and editing losses are omitted, although their gradients can also affect both reports through the shared parameters.

A.8.2 Preference Pairs Concentrate the Preference Signal on Residual Errors

This subsection decomposes the preference gradient into contributions from edited and shared sentences and bounds the shared contribution under Assumption 1.

Token partition. The operator $\oplus$ acts on whole sentences, so the tokens of the two reports split into three groups: (i) $\mathcal{T}^{\mathrm{err}}(y)$, the tokens of sentences of y that are REVISEd or DELETEd by $\mathcal{E}^+$; (ii) $\mathcal{T}^{\mathrm{cor}}(y^+)$, the tokens of sentences of y^+ produced by REVISE or ADD; and (iii) the shared sentences, which appear verbatim in both reports, whose tokens we write $\mathcal{T}^{\mathrm{sh}}(y)$ and $\mathcal{T}^{\mathrm{sh}}(y^+)$ with the induced bijection $t \mapsto t'$.

Assumption 1 (Weak cross-sentence coupling)**.** *There exists a small $\varepsilon \geq 0$ such that $\big\|g_{t'}(y^+) - g_t(y)\big\| \leq \varepsilon$ for every shared token pair (t, t').*

A shared token has the same identity in both reports; its conditional differs only through edited sentences appearing earlier in the context. In *Findings*-style reports, sentences state largely independent per-structure observations, so each conditional is dominated by the study (x, p^{gen}) and the sentence-local prefix rather than by the surrounding sentences, which motivates Assumption 1.

Two refinements are worth recording. First, for shared tokens that precede the first edit the two contexts are literally identical, so $g_{t'}(y^+) = g_t(y)$ exactly; ε only measures the perturbation of shared tokens downstream of an edit. Second, what the later analysis requires is relative smallness: we assume $\varepsilon \ll \|g_t(y)\|$ for typical shared tokens, i.e., the context perturbation is small compared to the token gradient itself (this is the form used in Assumption 2).

Proposition 1 (Gradient support of a preference pair)**.** *Since $\tilde{m}$ and d depend only on π_{ref}, $\nabla_\theta \mu = \nabla_\theta \ell_\theta(y^+) - \nabla_\theta \ell_\theta(y)$, and the pair term of Eq. 14 satisfies*

$$\begin{aligned} \nabla_\theta\Big[\Delta\big(\mu - \tilde{m}\big)^2\Big] &= 2\Delta\big(\mu - \tilde{m}\big)\big[\Gamma^{\mathrm{cor}} - \Gamma^{\mathrm{err}}\big] + r, \\ \Gamma^{\mathrm{cor}} &:= \frac{1}{|y^+|}\sum_{t \in \mathcal{T}^{\mathrm{cor}}} g_t(y^+), \quad \Gamma^{\mathrm{err}} := \frac{1}{|y|}\sum_{t \in \mathcal{T}^{\mathrm{err}}} g_t(y), \end{aligned} \tag{17}$$

where Γ^{cor} and Γ^{err} are correction/error tokens' contributions to the length-normalized likelihood gradient, and the remainder r collects the shared-sentence contributions and is bounded by

$$\|r\| \le 2\Delta|\mu - \tilde{m}|\left(\frac{|\mathcal{T}^{\mathrm{sh}}|}{|y^+|}\,\varepsilon + \frac{\big|\,|y^+| - |y|\,\big|}{|y|\,|y^+|}\sum_{t\in\mathcal{T}^{\mathrm{sh}}(y)}\big\|g_t(y)\big\|\right). \tag{18}$$

Proof. Split $\nabla_\theta \ell_\theta(y^+) - \nabla_\theta \ell_\theta(y)$ into edited and shared tokens. The edited tokens give $\Gamma^{\mathrm{cor}} - \Gamma^{\mathrm{err}}$. Each shared pair contributes

$$\frac{g_{t'}(y^+)}{|y^+|} - \frac{g_t(y)}{|y|} = \frac{g_{t'}(y^+) - g_t(y)}{|y^+|} + g_t(y)\Big(\frac{1}{|y^+|} - \frac{1}{|y|}\Big);$$

bound the first term by Assumption 1, the second by $\big|\frac{1}{|y^+|} - \frac{1}{|y|}\big| = \frac{|\,|y|-|y^+|\,|}{|y|\,|y^+|}$, sum over $\mathcal{T}^{\mathrm{sh}}$, and apply the triangle inequality. □

The two retained terms $(\Gamma^{\mathrm{cor}}, \Gamma^{\mathrm{err}})$ in Eq. 17 are supported exclusively on the edited tokens. The remainder collects the two distinct mechanisms by which a shared sentence can fail to cancel: a *context perturbation* (an upstream edit slightly shifts the conditional of a downstream shared token; the ε term) and a *renormalization mismatch* (an identical gradient enters the two length-normalized averages with weights $\frac{1}{|y^+|}$ versus $\frac{1}{|y|}$; the second term, which vanishes identically when $|y^+| = |y|$). These terms are small when cross-sentence coupling and the relative length change are small. Concentration on the edited contributions additionally requires the remainder to be small relative to $2\Delta|\mu - \tilde{m}|\|\Gamma^{\mathrm{cor}} - \Gamma^{\mathrm{err}}\|$.

Corollary 1 (Direction of the margin update)**.** *For a gradient-descent step on the pair term $\Delta(\mu - \tilde{m})^2$ alone, with both gradient paths active and $\mu < \tilde{m}$,*

$$\delta\mu = 2\eta\Delta(\tilde{m} - \mu)\|\nabla_\theta \mu\|^2 + O(\eta^2).$$

Thus, the pair increases its margin to first order whenever $\nabla_\theta \mu \neq 0$.

Interpretation. The candidate y is sampled from the recent policy, and y^+ is constructed through reward-improving local edits. When the shared remainder is small relative to the edited contributions, the preference gradient is dominated by the edited sentences.

Remark 1 (Contrast with reference-based pairs)**.** *A pair $(y^* \succ y)$ built directly from the reference report may differ in wording, ordering, and coverage even when the two reports are clinically close. These differences can limit the shared-token cancellation analyzed in Proposition 1, introducing gradient contributions from otherwise clinically equivalent content. The self-edited winner y^+ instead differs from a policy sample y by only $|\mathcal{E}^+_{i,g}|$ sentence-level edits, preserving the unedited sentences of the sampled draft. Under Assumption 1, this shared structure supports cancellation of shared-token contributions, up to the remainder in Eq. 18. When this remainder is small relative to the edited contributions, the preference gradient is concentrated on the edited sentences.*

Remark 2. *Proposition 1 describes the generic two-sided pair. The Case-1 detachment of Appendix A.8.4 deliberately removes the loser's gradient path from selected pairs, trading part of this cancellation for sign consistency with the report-level objective.*

A.8.3 Equilibrium of the Adaptive-Target Objective

We derive the claim following Eq. 12: at equilibrium, $\ell_\theta(y^+) - \ell_\theta(y) = \tilde{m} - d \ge 0$ whenever $d \le 3m$ (Eq. 13).

Proposition 2 (Equilibrium gap)**.** *Fix a pair $(y^+ \succ y) \in \mathcal{P}$ and let $m > 0$. The pairwise loss in Eq. 14 is uniquely minimized in μ at $\mu = \tilde{m}$, at which point*

$$\ell_\theta(y^+) - \ell_\theta(y) = \tilde{m} - d \;\ge\; 0, \quad \textit{whenever } d \le 3m. \tag{19}$$

Proof. Step 1 (equilibrium condition). For a fixed pair, $\Delta(\mu - \tilde{m})^2$ with $\Delta > 0$ is a strictly convex quadratic in μ, and $\tilde{m}$ is θ-independent. Its unique minimizer solves

$$\frac{\partial}{\partial\mu}\,\Delta\big(\mu - \tilde{m}\big)^2 = 2\Delta\big(\mu - \tilde{m}\big) = 0 \iff \mu = \tilde{m}. \tag{20}$$

Step 2 (margin identity). Expanding $h_\theta = \ell_\theta - \ell_{\text{ref}}$,

$$\begin{aligned}\mu &= \big[\ell_\theta(y^+) - \ell_{\text{ref}}(y^+)\big] - \big[\ell_\theta(y) - \ell_{\text{ref}}(y)\big] \\ &= \ell_\theta(y^+) - \ell_\theta(y) + \underbrace{\big[\ell_{\text{ref}}(y) - \ell_{\text{ref}}(y^+)\big]}_{=\,d}.\end{aligned} \tag{21}$$

Substituting $\mu = \tilde{m}$ from Eq. 20 yields $\ell_\theta(y^+) - \ell_\theta(y) = \tilde{m} - d$.

Step 3 (sign of the gap). With $\tilde{m} = \max\big(m, \min(d, 3m)\big)$, three cases arise:

1. $d \le m$: $\min(d, 3m) = d$ and $\max(m, d) = m$, so $\tilde{m} = m$ and the gap is $m - d \ge 0$, strictly positive whenever $d < m$; in particular, when the reference is indifferent ($d = 0$) or already prefers the winner ($d < 0$), the floor still enforces a margin of at least m.
2. $m < d \le 3m$: $\min(d, 3m) = d$ and $\max(m, d) = d$, so $\tilde{m} = d$ and the gap is exactly 0: the revised report y^+ has the same length-normalized log-likelihood as the candidate y.
3. $d > 3m$: $\min(d, 3m) = 3m$ and $\max(m, 3m) = 3m$, so $\tilde{m} = 3m$ and the gap is $3m - d < 0$; the guarantee is deliberately forgone (see Remark 5).

Cases 1–2 cover all $d \le 3m$ and give a nonnegative gap. □

Remark 3 (Reading of the equilibrium)**.** *Proposition 2 is a per-pair stationarity statement: $\mu = \tilde{m}$ is the point at which this pair's pull vanishes (Lemma 1(b) below, $u \propto \mu - \tilde{m}$), holding everything else fixed. In training, μ is a function of the shared θ and the pair's term is balanced against all other losses, so $\tilde{m} - d$ is the gap the objective* targets *for this pair, not a guarantee about the realized optimum of the full objective.*

Remark 4 (Why the target must adapt)**.** *With a fixed target $\tilde{m} \equiv m$, Eq. 19 gives a gap of $m - d < 0$ for every pair with $d > m$: the loss then sits at its optimum while the revised report y^+ still has a lower length-normalized log-likelihood than the candidate y, i.e., the objective is satisfied without achieving the intended reversal. Raising the target to $\tilde{m} = d$ on $(m, 3m]$ is the smallest choice that closes the gap.*

Remark 5 (Why the target is capped)**.** *By Lemma 1, a pair's per-token preference pull scales with $|\mu - \tilde{m}|$. For pairs with $d > 3m$, an uncapped target $\tilde{m} = d$ demands a large reference-adjusted margin. Capping the target at $3m$ relaxes this demand: when $\mu < 3m$, it reduces the margin deficit from $d - \mu$ to $3m - \mu$. For these pairs, the resulting equilibrium gap is $3m - d < 0$, trading the nonnegative-gap guarantee for a smaller target margin.*

Remark 6 (Length-normalized reading)**.** *ℓ is length-normalized, so Eq. 19 compares average per-token log-likelihoods. Local edits typically cause only small relative changes in report length, so $|y^+| \approx |y|$. The normalization factors $1/|y^+|$ and $1/|y|$ are therefore close, keeping the length-normalization mismatch between the two reports small.*

A.8.4 Sign Consistency Between the Report-Level and Preference Gradients

Both $\mathcal{L}_{\text{PG}}^{\text{rep}}$ (Eq. 6) and $\mathcal{L}_{\text{prf}}$ (Eq. 14) act on the token conditionals of the same candidate $y = y_{i,g}$. We make their per-token update directions explicit, show that the two conflict cases are exactly the sign-inconsistent configurations, and derive why each is resolved as described. Write the relevant part of the Stage-3 objective (Eq. 15) as $\mathcal{L} = \mathcal{L}_{\text{PG}}^{\text{rep}} + \omega_{\text{prf}}\mathcal{L}_{\text{prf}}$ with loss weight $\omega_{\text{prf}} > 0$ (cf. the Setup paragraph (A.8.1) for the omitted terms).

Lemma 1 (Per-token pulls)**.** *Define the pull on a conditional as $u_t := -\partial\mathcal{L}/\partial z_t$, the first-order pressure of the loss to raise ($u_t > 0$) or lower ($u_t < 0$) z_t. For a token t of the candidate y:*

(a) Report level. *With $\rho_t = \exp\big(z_t - z_t^{\text{old}}\big)$ and $Z = \sum_{i,g} |y_{i,g}|$, Eq. 5 and Eq. 6 give*

$$u_t^{\text{rep}} = a_t\,\hat{A}, \quad a_t := \frac{1}{Z}\,\underbrace{4\,\sigma_t\,(1 - \sigma_t)\,\rho_t}_{=\,\phi'_\tau(\rho_t)\,\rho_t} \;>\; 0, \tag{22}$$

where $\sigma_t := \sigma\big(\tau(\rho_t - 1)\big)$. Because $a_t > 0$ strictly for every ρ_t (the smooth gate attenuates but never zeroes the gradient), $\operatorname{sign} u_t^{\text{rep}} = \operatorname{sign} \hat{A}$ *at* every *token of y. This is where the smooth gate matters: under PPO clipping, the surrogate gradient can vanish on the clipped branch, depending*

on the sign of $\hat{A}$; with ϕ_τ every token carries a (possibly attenuated) reward-signed pull, which is precisely what makes the conflict analysis below exhaustive.

(b) Preference level. *In Eq. 14, μ depends on z_t only through $\ell_\theta(y)$, with $\partial\mu/\partial z_t = -1/|y|$, so*

$$u_t^{\mathrm{prf}} = b\,(\mu - \tilde{m}), \qquad b := \frac{2\,\omega_{\mathrm{prf}}\,\Delta}{|\mathcal{P}|\,|y|} > 0; \tag{23}$$

symmetrically, a winner token $t' \in y^+$ ($\partial\mu/\partial z_{t'} = +1/|y^+|$) receives $u_{t'}^{\mathrm{prf},+} = -\frac{2\omega_{\mathrm{prf}}\Delta}{|\mathcal{P}|\,|y^+|}(\mu - \tilde{m})$.

Proof. (a) The candidate contributes $-\frac{1}{Z}\sum_t \phi_\tau(\rho_t)\hat{A}$ to $\mathcal{L}$, with $\hat{A}$ computed from the frozen π_{old} rollouts; differentiate using $\partial\rho_t/\partial z_t = \rho_t$ and $\phi'_\tau(\rho) = 4\sigma(\tau(\rho-1))\big[1-\sigma(\tau(\rho-1))\big]$, then negate. (b) Differentiate $\frac{\omega_{\mathrm{prf}}}{|\mathcal{P}|}\Delta(\mu-\tilde{m})^2$ using $\partial\mu/\partial z_t(y) = -1/|y|$ for loser tokens and $\partial\mu/\partial z_{t'}(y^+) = 1/|y^+|$ for winner tokens, then negate. □

Lemma 1(b) characterizes the direct preference pull on candidate tokens: it is negative when $\mu < \tilde{m}$ and positive when $\mu > \tilde{m}$. By part (a), the direct report-level pull has the sign of $\hat{A}$. Two structural features are worth noting. First, the squared objective encourages each pair's reference-adjusted margin to match the adaptive target $\tilde{m}$. Second, the preference pull $b(\mu - \tilde{m})$ is uniform across candidate tokens, whereas the report-level pull $a_t\hat{A}$ depends on token-specific coefficients a_t. When both pulls are nonzero and a_t varies across tokens, they cannot be related by a single global scaling factor.

From pulls to realized updates. The pulls of Lemma 1 describe the pressure each loss places on a conditional, but the optimizer does not act on the conditionals directly: it updates the shared parameters θ, from which every conditional in the batch is computed, so a step taken on behalf of one token can in principle move any other. To first order, this coupling is governed by the Gram matrix of per-token gradients. For one descent step $\theta \leftarrow \theta - \eta\nabla_\theta\mathcal{L}$, a Taylor expansion of z_t in θ gives $\delta z_t = \langle g_t, -\eta\nabla_\theta\mathcal{L}\rangle + O(\eta^2)$, and since $\mathcal{L}$ depends on θ only through the conditionals, the chain rule gives $\nabla_\theta\mathcal{L} = \sum_s(\partial\mathcal{L}/\partial z_s)\,g_s = -\sum_s u_s\,g_s$, where s ranges over all tokens of all sequences in the batch. Combining the two,

$$\delta z_t = \eta\sum_s \langle g_t, g_s\rangle\, u_s + O(\eta^2), \tag{24}$$

i.e., $\delta z = \eta\,G\,u$ with $G_{ts} = \langle g_t, g_s\rangle$: a conditional moves not with its own pull but with every pull u_s, weighted by the overlap between gradient directions. The diagonal entry $s = t$ contributes $\eta\|g_t\|^2 u_t$; every off-diagonal entry is cross-talk between tokens through the shared parameters. Whether the sign analysis of Lemma 1 survives this cross-talk is a question about the shape of G, which the following assumption answers.

Assumption 2 (Local gradient geometry)**.** *For distinct, unmatched conditionals $s \neq t$ in the batch, we assume $|\langle g_s, g_t\rangle| \ll \|g_s\|\,\|g_t\|$. Identical conditionals, including those within partially revised sentences, are treated as matched.*

We further assume that the aggregate cross-token contribution is small relative to the retained contribution. Specifically, for an unpaired conditional with $u_t \neq 0$ and $\|g_t\| > 0$,

$$\left|\sum_{s\neq t}\langle g_t, g_s\rangle u_s\right| \leq \kappa_{\mathrm{geom}}\|g_t\|^2|u_t|,$$

for a small constant $0 \leq \kappa_{\mathrm{geom}} < 1$. For shared conditionals, the aggregate first-order contribution omitted from Eq. 26 is assumed to be bounded in magnitude by κ_{geom} times the magnitude of the nonzero combined contribution retained there.

We use this geometry as a local analytical approximation, motivated by studies of gradient interactions (Sankararaman et al., 2020; Fort et al., 2019). It retains each token's direct contribution and the coupling between matched conditionals, while controlling the aggregate effect of the remaining terms.

For a shared pair (t, t') between y and y^+, the strong gradient alignment follows from Assumption 1. By Cauchy–Schwarz,

$$\begin{aligned}\big|\langle g_{t'}, g_t\rangle - \|g_t\|^2\big| &= \big|\langle g_{t'} - g_t, g_t\rangle\big| \\ &\le \|g_{t'} - g_t\|\,\|g_t\| \le \varepsilon\|g_t\|.\end{aligned}$$

Thus, provided that $\varepsilon \ll \|g_t\|$, $\langle g_{t'}, g_t\rangle \approx \|g_t\|^2$: the interaction between matched conditionals is comparable in magnitude to the diagonal contribution.

For an unpaired conditional, retaining the diagonal term of Eq. 24 gives

$$\delta z_t \approx \eta\|g_t\|^2 u_t. \tag{25}$$

The aggregate bound in Assumption 2 ensures $\mathrm{sign}(\delta z_t) = \mathrm{sign}(u_t)$ for a sufficiently small gradient-descent step. Under these conditions, the direct pull therefore determines the direction of the conditional's update.

For a shared conditional, the matched contribution must also be retained. Using the shared-pair approximation above gives

$$\delta z_{t'} \approx \eta\big(\|g_{t'}\|^2 u_{t'} + \|g_t\|^2 u_t\big). \tag{26}$$

The second term captures how a pull on the candidate reaches the corresponding conditional in the revised report. The shared conditional's update therefore depends on the combined contributions of both pulls. Proposition 3 below quantifies how candidate-side suppression can transfer to the revised report through these shared conditionals.

Net pull and the two conflict cases. Combining Lemma 1, each loser token receives

$$u_t = a_t\,\hat{A} + b\,\big(\mu - \tilde{m}\big). \tag{27}$$

When $\hat{A}(\mu - \tilde{m}) \ge 0$, the direct pulls do not oppose each other, and both losses are applied unchanged. Strict sign conflicts arise in the following two cases:

Case 1: $\hat{A} > 0$ and $\mu < \tilde{m}$. Then $u_t = a_t\hat{A} - b(\tilde{m} - \mu)$ mixes terms of opposite sign. The two objectives exert opposing direct pulls on the candidate. Because a_t can vary across tokens while the preference pull is uniform, their balance can differ across the report, weakening either contribution and making the net pull sensitive to loss scaling.

Resolution (detach). Replacing $h_\theta(y)$ by $\mathrm{sg}\big[h_\theta(y)\big]$ in μ (sg: stop-gradient) leaves the value of μ unchanged and removes only the gradient path through $\ell_\theta(y)$. On loser tokens, $u_t = a_t\hat{A} > 0$, while the winner's pull is unaffected, $u_{t'}^{\mathrm{prf},+} = \frac{2\omega_{\mathrm{prf}}\Delta}{|\mathcal{P}|\,|y^+|}(\tilde{m} - \mu) > 0$. The preference gradient is then computed only through $\ell_\theta(y^+)$, while the report-level objective retains its direct reinforcement of the positive-advantage candidate. Moreover, by Assumption 1, the gradients of shared conditionals in y and y^+ are close. After detachment, the report-level term provides a positive direct pull on each shared candidate token, while the preference term provides a positive direct pull on its matched winner token. Thus, the two direct contributions associated with this pair no longer oppose each other on the shared conditionals.

Case 2: $\hat{A} < 0$ and $\mu > \tilde{m}$. Here $u_t^{\mathrm{prf}} = b(\mu - \tilde{m}) > 0$ would promote the below-average candidate y, opposing its negative report-level pull. The preference loss would also exert a negative pull on the higher-reward revised report y^+, $u_{t'}^{\mathrm{prf},+} = -\frac{2\omega_{\mathrm{prf}}\Delta}{|\mathcal{P}|\,|y^+|}(\mu - \tilde{m}) < 0$, to reduce the excess margin $\mu - \tilde{m}$. We remove this pair from $\mathcal{P}$ to eliminate its conflicting preference update and potentially large squared-loss contribution. Masking the candidate's report-level term additionally removes its direct pressure toward further margin growth. The following proposition gives an additional reason for masking: if retained, this term can also suppress shared content in the revised report.

Proposition 3 (Suppression leakage onto the revised report)**.** *Consider the update contributed by the candidate's report-level term alone after its preference pair is removed. Assume a nonempty shared-token set with $\bar{c} > 0$, and that the aggregate omitted first-order contribution is small relative to the retained shared-token contribution. Under Assumptions 1 and 2, with $\hat{A} < 0$ and a sufficiently*

small gradient-descent step, the shared-token approximation gives

$$\delta\ell_\theta(y^+) \;\approx\; \frac{\eta}{|y^+|}\sum_{t\in\mathcal{T}^{\mathrm{sh}}(y)} \|g_t\|^2\, a_t\, \hat{A} \;=\; -\,\frac{|\mathcal{T}^{\mathrm{sh}}|}{|y^+|}\,\eta\,\bar{c}\,|\hat{A}| \;<\; 0,$$
$$\textit{where}\;\; \bar{c} := \frac{1}{|\mathcal{T}^{\mathrm{sh}}|}\sum_{t\in\mathcal{T}^{\mathrm{sh}}(y)} \|g_t\|^2\, a_t \;>\; 0. \tag{28}$$

Proof. By Eq. 24, a conditional of y^+ carries no pull of its own and moves only through its gradient overlaps with the pulled tokens, i.e., the tokens of y. For a shared winner conditional t' with matched candidate token t, the shared-gradient approximation derived from Assumption 1 gives $\langle g_{t'}(y^+), g_t(y)\rangle \approx \|g_t(y)\|^2$. Retaining this contribution and applying the aggregate bound in Assumption 2 yields $\delta z_{t'} \approx \eta\|g_t\|^2 u_t = \eta\|g_t\|^2 a_t\hat{A}$, with u_t given by Eq. 22. Contributions from the remaining overlaps, including those involving revised sentences, are treated as an aggregate remainder under the stated dominance condition. Therefore

$$\begin{aligned}
\delta\ell_\theta(y^+) &= \frac{1}{|y^+|}\sum_{t'\in y^+}\delta z_{t'} && \text{(definition of } \ell_\theta\text{)}\\
&\approx \frac{1}{|y^+|}\sum_{t\in\mathcal{T}^{\mathrm{sh}}(y)}\delta z_{t'} && \text{(retaining the shared-token contribution)}\\
&\approx \frac{\eta}{|y^+|}\sum_{t\in\mathcal{T}^{\mathrm{sh}}(y)}\|g_t\|^2 a_t\hat{A} && \text{(Eq. 26 with } u_{t'}=0\text{)}\\
&= -\frac{|\mathcal{T}^{\mathrm{sh}}|}{|y^+|}\,\eta\,\bar{c}\,|\hat{A}| \;<\; 0 && (\hat{A}<0\text{, def. of }\bar{c}\text{),}
\end{aligned}$$

where the second line sums $\delta z_{t'}$ over shared pairs via the bijection $t \mapsto t'$. □

Remark 7 (Partially revised sentences)**.** *Equation 28 explicitly retains matched tokens from unchanged sentences. Overlaps involving added or revised sentences are collected in the aggregate remainder. In particular, positively aligned conditionals within partially revised sentences can receive additional suppression from the candidate's negative pulls.*

When local edits preserve most report tokens, $\frac{|\mathcal{T}^{\mathrm{sh}}|}{|y^+|} \approx 1$, and shared-token coupling can transfer substantial suppression to the revised report. Masking the candidate's contribution to $\mathcal{L}_{\mathrm{PG}}^{\mathrm{rep}}$ sets its direct report-level pulls to zero, removing this source of suppression. Its KL term is retained for reference regularization.

Remark 8 (Why the two cases are treated asymmetrically)**.** *In Case 1, the retained report-level term promotes shared conditionals in y^+, in the same direction as the remaining winner-side preference pull. We therefore retain the report-level term and detach only the candidate's path through the preference loss. In Case 2, the candidate's report-level term continues to suppress shared conditionals in the higher-reward revised report after the preference pair is removed. We therefore also mask the candidate's report-level policy-gradient term.*

Summary of the sign-consistency rule.

sign $\hat{A}$	regime	pulls on y	action
$+$	$\mu < \tilde{m}$	conflict	detach $h_\theta(y)$ in $\mathcal{L}_{\mathrm{prf}}$
$-$	$\mu > \tilde{m}$	conflict	drop pair; mask y's PG, keep KL
$+$	$\mu \geq \tilde{m}$	no conflict	apply both unchanged
$-$	$\mu \leq \tilde{m}$	no conflict	apply both unchanged
0	any	no conflict	apply both unchanged

A.9 Implementation Details

Backbone and adaptation. All three stages fine-tune Qwen3-VL-4B-Instruct (Bai et al., 2025) with LoRA (Hu et al., 2022) applied to all linear layers, keeping the remaining parameters frozen.

We use LoRA rank $r_{\mathrm{LoRA}} = 32$, scaling factor $\alpha_{\mathrm{LoRA}} = 32$, and dropout 0.1 (0 during Stage-3 RL). All training runs use BF16 precision on up to 8 NVIDIA H100 80 GB GPUs.

Input construction. For each study, up to two current CXRs are provided with their view-position tags, followed by the indication and clinical history when available and, if present, the prior image and its report and indication. Images are resized while preserving the aspect ratio and then center-cropped to 512×512.

Stage 1: supervised report generation. Stage 1 trains on the full training set for 6 epochs with a per-GPU batch size of 8 (effective batch size 32), using the AdamW (Loshchilov et al., 2017) optimizer with learning rate 1×10^{-4}, a cosine annealing learning-rate schedule with a warm-up ratio of 0.05, and weight decay 0.001.

Stage 2: editor warm-up. The warm-up corpus $\mathcal{D}_2$ contains 100 training studies sampled uniformly at random from the target dataset (MIMIC-CXR or CheXpert Plus); each dataset thus uses a warm-up corpus drawn from itself. The evaluation is conducted on the entire validation and test sets of the corresponding datasets. For each study, the frozen Stage-1 model generates $\hat{y}$, and GPT-5.6 Sol (OpenAI, 2026) compares it with the reference report y^* and produces the minimal edit script c^* (the annotation prompt will be released in our code). Stage 2 trains for 2 epochs on MIMIC-CXR and 3 epochs on CheXpert Plus with a batch size of 3, a gradient accumulation of 2, a constant learning rate of 1×10^{-5} with a warm-up ratio of 0.05, and weight decay 0.01.

Stage 3: rollout configuration. Each RL step samples $B = 8$ studies; for each study, the behavior policy π_{old} generates $G = 8$ candidate reports with temperature 0.7 and top-p 0.8. Conditioned on the editing prompt, one edit script per candidate is generated with the same temperature and top-p. Edit blocks that fail to parse or reference nonexistent sentence identifiers are dropped; on average 99.9% of blocks are retained. The main cause of parsing failures is that DELETE actions reference nonexistent sentence identifiers. The behavior policy is refreshed every $K_{\mathrm{old}} = 16$ steps, and the reference policy π_{ref} is fixed at the Stage-2 checkpoint throughout.

Reward computation. The composite reward in Eq. 3 uses $\lambda = 0.4$. BLEU-2 and ROUGE-L are computed with pycocoevalcap[1]. F_1^{RadGraph} is the partial variant computed with the RadGraph[2] model, and F_1^{CheXbert} is the sample-level F1 computed over the 14 CheXbert[3] observation labels.

Advantage normalization and gating. We denote by γ the decay of the exponential moving averages used for both the report-level scale $\bar{\sigma}_r$ in Eq. 4 and the edit-level scale $\bar{\sigma}_\Delta$ in Eq. 9, and set $\gamma = 0.99$ with the EMAs initialized to 0.1. The edit-level advantage is clipped at $\kappa = 3$. The smooth gate in Eq. 5 uses $\tau_+ = 1.0$ for positive advantages and $\tau_- = 1.05$ for non-positive advantages, following SAPO (Gao et al., 2025), in both the report-level and edit-level losses. The KL coefficients are $\beta^{\mathrm{rep}} = 0.01$ and $\beta^{\mathrm{edt}} = 0.001$. We set β^{edt} an order of magnitude below β^{rep} to allow greater flexibility in learning editing actions, while retaining KL regularization to limit drift from the reference policy.

Correction-distillation preference loss. The base margin target in Eq. 12 is $m = 0.5$.

Overall objective and optimization. The loss weights in Eq. 15 are $\omega_{\mathrm{edt}} = 0.1$ and $\omega_{\mathrm{prf}} = 0.1$. Stage 3 trains for 50,000 steps on MIMIC-CXR and 45,522 steps (9 epochs) on CheXpert Plus with the AdamW optimizer and a constant learning rate of 3×10^{-6} with a warm-up ratio of 0.05.

Inference. At test time, ED-SR performs a single generation pass conditioned on (x, p^{gen}) with beam search decoding (beam size 3) for reproducibility. No editing pass is performed, and no edit script is generated at inference.

The complete set of hyperparameters is provided in Table 7.

[1] https://github.com/salaniz/pycocoevalcap
[2] https://github.com/Stanford-AIMI/radgraph
[3] https://github.com/stanfordmlgroup/CheXbert

Table 7: Hyperparameters of ED-SR. The EMA decay γ is shared by $\bar{\sigma}_r$ and $\bar{\sigma}_\Delta$. Dashes indicate hyperparameters without a dedicated symbol in the main text. Unless stated otherwise, CheXpert Plus uses the same hyperparameter settings as MIMIC-CXR.

Hyperparameter	Symbol	Value
Shared		
LoRA rank	r_{LoRA}	32
LoRA scaling factor	α_{LoRA}	32
LoRA dropout (Stages 1–2 / Stage 3)	–	0.1 / 0
Precision	–	BF16
Image resolution	–	512×512
Max report token length (MIMIC / CheXpert)	–	160 / 300
Max edit-script token length	–	400
Stage 1: supervised generation		
Epochs	–	6
Batch size (per GPU / effective)	–	8 / 32
Optimizer	–	AdamW
Learning rate	–	1×10^{-4}
LR schedule (warm-up ratio)	–	cosine (0.05)
Weight decay	–	0.001
Stage 2: editor warm-up		
Warm-up corpus size	$\lvert\mathcal{D}_2\rvert$	100
Generation / editing loss weights	–	$1/2$ / $1/2$
Epochs (MIMIC / CheXpert)	–	2 / 3
Batch size	–	3
Gradient accumulation	–	2
Learning rate	–	1×10^{-5}
LR schedule (warm-up ratio)	–	constant (0.05)
Weight decay	–	0.01
Stage 3: rollouts		
Studies per step	B	8
Group size	G	8
Behavior-policy refresh interval	K_{old}	16
Sampling temperature	–	0.7
Sampling top-p	–	0.8
Stage 3: reward and advantages		
Reward balance	λ	0.4
EMA decay	γ	0.99
EMA initialization	–	0.1
Edit-advantage clip	κ	3
Gate temperature (positive adv.)	τ_+	1.0
Gate temperature (non-positive adv.)	τ_-	1.05
Stage 3: losses and optimization		
KL coefficient (report-level)	β^{rep}	0.01
KL coefficient (edit-level)	β^{edt}	0.001
Preference margin	m	0.5
Adaptive-target cap	–	$3m$
Edit-loss weight	ω_{edt}	0.1
Preference-loss weight	ω_{prf}	0.1
Optimizer	–	AdamW
Learning rate	–	3×10^{-6}
LR schedule (warm-up ratio)	–	constant (0.05)
Training steps (MIMIC / CheXpert)	–	50,000 / 45,522 (9 epochs)
Weight decay	–	0.0
Inference		
Decoding strategy	–	beam search
Beam size	–	3