arXiv:2608.08713v1 [cs.CV] 9 Aug 2026

Resolution Meets Reduction: Efficient Visual Context for 3D Radiology Report Generation

Jonathan Suprijadi[1,*], Raphael Stock[1,*], Moritz Langenberg[1,*], David Zimmerer[1], Kim-Celine Kahl[1], Stefan Denner[1], Yannick Kirchhoff[1], Karol Gotkowski[1], Maximilian Rokuss[1], Jeremias Traub[1], Tassilo Wald[1], Constantin Ulrich[1], and Klaus Maier-Hein[1]

Abstract

Vision-language models offer a promising path toward automating radiology report generation, but applying them to full 3D CT volumes poses substantial computational challenges. Modern foundation vision encoders (VEs) can produce tens of thousands of vision tokens per scan, making the visual sequence passed to the large language model (LLM) a primary computational bottleneck. Vision-to-language projectors can compress this sequence to reduce computation, but may discard clinically relevant detail; conversely, effective compression can accommodate higher-resolution inputs while keeping the downstream token count fixed. How this vision-token budget should be allocated across input field of view, spatial resolution, and vision-to-language projection therefore remains an open design question. We systematically evaluate four heterogeneous VEs (CNN- and ViT-based), five token-reducing projectors at up to 64x compression alongside a non-reducing MLP projector baseline, and five instruction-tuned LLMs (1.7B–4B) on two large-scale CT report datasets (CT-RATE and Merlin). At matched LLM token budgets, anatomy-guided region of interest cropping is the most consistent strategy, improving clinical macro F1 in 19 of 20 settings by +3.7 points on average for the 3D ViT Primus encoder and +1.1 for the slice-based 2D ViT Curia encoder. Increasing input resolution further is strongly projector-dependent: the PerceiverResampler, paired with higher-resolution Curia features, yields the strongest configuration in the resolution study on both datasets. Our best configurations achieve state-of-the-art clinical macro F1 on the test sets, reaching 49.5 on CT-RATE and 49.0 on Merlin. Code and models will be published upon publication.

Keywords: 3D radiology report generation, foundation model adaptation, token compression, vision-language models

*Contributed equally. Co-first authors may list as lead on CV.

[1]All authors are or were with the Division of Medical Image Computing, German Cancer Research Center (DKFZ), Heidelberg, Germany (e-mail: jonathan.suprijadi@dkfz-heidelberg.de, raphael.stock@dkfz-heidelberg.de, moritz.langenberg@dkfz-heidelberg.de).

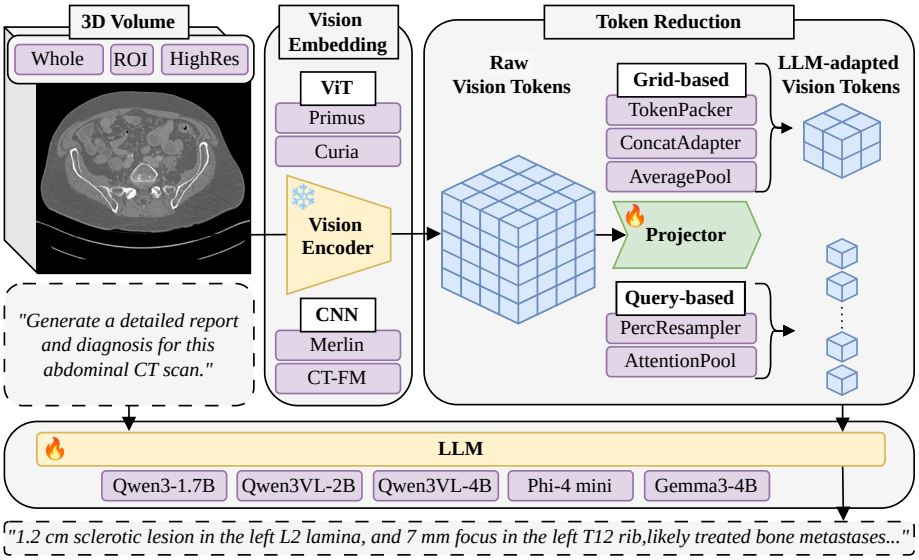

Figure 1: **Overview of our VLM pipeline for 3D radiology report generation**. Each 3D CT volume is first preprocessed according to one of three input regimes: whole-volume encoding, anatomy-guided ROI cropping, or higher-resolution encoding, before a frozen vision encoder extracts vision tokens. The projector compresses these tokens before they are passed to the LLM for report generation. This work systematically investigates the design choices along this pipeline, including the input regime, vision encoder, projector, and LLM.

1 Introduction

The diagnostic workload of radiologists has increased dramatically over the past two decades, and the interpretation of 3D Computed Tomography (CT) scans is among the most time-consuming tasks in clinical practice [1]. Vision-Language Models (VLMs) offer a promising solution by automating the drafting of diagnostic reports from imaging data [2, 3]. Recent VLM architectures consist of a pretrained Vision Encoder (VE) that produces vision tokens and a projector that maps them into the embedding space of a Large-Language Model (LLM), which then autoregressively generates the report [4] (Fig. 1). The projector is typically a compression-free Multilayer Perceptron (MLP) that forwards the entire token sequence to the LLM [5].

Applying this paradigm to full 3D CT volumes can produce extremely long visual contexts. Depending on the VE, spatial resolution, and field of view (FOV), a single scan may yield tens of thousands of vision tokens, with higher-resolution inputs increasing this number further. Common strategies for limiting the sequence length, such as downsampling to a fixed grid [6–8], region of interest (ROI) cropping [9, 10], or slice selection [11], reduce the amount of anatomy or spatial detail retained from the scan and may therefore suppress

diagnostically relevant findings [12]. Preserving finer spatial detail, however, produces increasingly long and potentially redundant token sequences at substantial computational cost. Recent work has addressed this challenge on the VE side by learning more compact or structured volumetric representations, for example through learned volumetric tokenization [13] or concept-specific visual aggregation [14]. For pretrained VEs, the vision-to-language projector provides a complementary control point, aggregating the encoder representation into a compact context for report generation while regulating the downstream sequence length. Yet despite this central role, projector choice in 3D VLMs is rarely studied as an independent design variable. Current approaches span a single linear layer in Merlin [8], an MLP in COLIPRI [5], 3D average pooling [6, 15], attention pooling [7, 16], and Perceiver-style resampling in RadFM [17], but their relative behavior across architectures and compression levels is not well characterized. The closest prior study [18] explores only part of this space, but is constrained to 1,287 training cases [19] and three projectors, one native to the evaluated VE and thus plausibly favored. We investigate these questions in two complementary experiments.

1.0.1 Input Field of View and Resolution

We present, to the best of our knowledge, the first controlled study of input FOV and effective spatial resolution for 3D Radiology Report Generation (RRG) at a matched LLM token budget. Holding the number of vision tokens forwarded to the LLM fixed separates the input regime from the sequence length the LLM attends to, and we compare whole-volume encoding, anatomy-guided ROI cropping, and two mechanisms for raising the effective resolution across two VEs and two datasets. ROI cropping is the most consistent intervention in the study, improving clinical macro F1 in 19 of 20 configurations, by 3.7 on average for Primus [20] and 1.1 for Curia [21], at the cost of a segmentation pass per volume. Further increases in resolution do not improve performance on average and benefit only projectors whose aggregation mechanisms can effectively accommodate the stronger compression induced by larger inputs.

1.0.2 Encoder, Projector, and Language Model

We report the broadest evaluation of encoder, projector, and language-model choices for 3D RRG to date. Our study spans four medical imaging foundation VEs with convolutional neural network (CNN)- and vision transformer (ViT)-based architectures, five token-reducing projectors evaluated at compression ratios up to 64× alongside a token-preserving MLP baseline, and five instruction-tuned LLMs ranging from 1.7B to 4B parameters. Experiments are conducted on two large-scale CT image-report datasets with more than 25k studies each, using three seeds per configuration and approximately 60,000 A100 and 6,000 H100 GPU-hours in total. The study varies one factor at a time around a common development configuration, shows that the choice of the VE has a larger impact than the choice of the LLM, and characterizes which projector families

remain stable under aggressive compression. To facilitate clinical evaluation on Merlin, we release a lightweight ModernBERT [22] classifier for 30 abnormality classes, trained on LLM extracted labels [23], achieving 95.7 macro F1 on the held-out test split.

Together, these studies yield state-of-the-art clinical performance on both benchmarks, with a macro F1 of 49.5 on CT-RATE against 48.0 [24] for the strongest comparable published system, and 49.0 on Merlin against 23.8 [14].

2 Methods

2.1 Overall Architecture

Our framework follows the standard three-stage LLaVA-style VLM architecture consisting of a pretrained VE, a vision-to-language projector, and a pretrained LLM. Given a 3D CT volume, the VE produces a sequence of N visual feature tokens. The projector maps these features into the embedding space of the LLM, yielding M projected vision tokens, where $M = N$ for token-preserving projectors and $M < N$ when token compression is applied. The projected tokens are prepended to the embeddings of a textual instruction and condition the autoregressive generation of the radiology report. To isolate projector design from visual representation learning, the VE remains frozen, while the projector and LLM are jointly trained. Unless otherwise specified, comparisons vary the projector architecture under a fixed LLM.

2.2 Vision-Language Projectors

For systematic comparison, we categorize the evaluated token-compressing projectors as grid-based or query-based and consider a token-preserving MLP as the baseline.

2.2.1 Grid-Based Projectors

Grid-based projectors group neighboring vision tokens into local blocks according to the per-axis reduction factors $\mathbf{r} = (r_x, r_y, r_z)$, where a factor of 1 indicates no reduction along the corresponding axis. Aggregating each block into a single representation reduces the token count from N to $M = N/(r_x r_y r_z)$ while retaining the spatial organization of the encoder feature grid. The resulting tokens are subsequently mapped to the LLM embedding dimension using an MLP.

The *AveragePool* projector [6] averages the tokens within each local block using pooling kernels and strides given by $\mathbf{r} = (r_x, r_y, r_z)$.

The *ConcatAdapter*, following Qwen3-VL [25], concatenates the tokens within each local block along the feature dimension. This increases the input dimension of the subsequent projection by a factor of $r_x r_y r_z$ while avoiding information aggregation before projection.

TokenPacker [26] constructs a low-resolution query grid by interpolating the encoder feature grid and uses the original high-resolution features as keys and

values. Each output token thereby aggregates fine-grained information from its corresponding spatial region. For comparability with other projectors, we use only the final-layer encoder features rather than the multi-layer features employed in the original implementation.

2.2.2 Query-Based Projectors

Query-based projectors introduce a fixed set of L learnable tokens that aggregate information from all N vision tokens through cross-attention, decoupling the output length from the spatial resolution of the encoder features.

AttentionPool [7] implements this aggregation using a single multi-head cross-attention layer.

The *PerceiverResampler* [27] extends this design by stacking multiple Perceiver layers, each composed of a cross-attention step followed by a 2-layer MLP with residual connections. Furthermore, the queries are included in the key-value sequence, enabling information exchange among queries. While query-based projectors provide flexible and globally informed compression, they sacrifice the structured spatial topology maintained by grid-based projectors.

2.2.3 Token-Preserving Baseline

Following the LLaVA paradigm [4], an MLP independently maps each of the N vision tokens into the LLM embedding space. The original sequence length is retained, such that $M = N$, providing a compression-free baseline.

2.2.4 Token-Budget Matching

The VEs considered in this work produce feature grids with substantially different token counts depending on the encoder and preprocessing configuration (Table 1). For each configuration, we evaluate one or more reduction vectors $\mathbf{r}$. Each reduction vector defines a token compression ratio of $N/M = r_x r_y r_z$ and a corresponding output token budget of $M = N/(r_x r_y r_z)$. Different choices of $\mathbf{r}$ therefore correspond to the evaluated compression levels.

To enable fair comparisons across projector families, the number of tokens passed to the LLM is matched for a given VE, preprocessing configuration, and compression ratio. Grid-based projectors use the corresponding reduction vector $\mathbf{r}$, whereas query-based projectors use $L = M$ learnable queries. Both projector families therefore expose the LLM to the same visual sequence length while employing different aggregation mechanisms.

2.3 Vision Encoders and Input Preprocessing

2.3.1 Region-of-Interest Cropping

Before encoder-specific resizing or resampling, each volume is cropped to a target anatomical ROI automatically extracted using TotalSegmentator [28]. The

Table 1: **Dataset- and vision-encoder-specific input preprocessing.** Token counts are reported before projection. "Full" denotes the full scan extent after whole-volume resizing. The VE input specifies the spatial input to a single encoder pass; Curia independently encodes 192 axial slices of 512^2 pixels, whereas Primus-4B independently encodes four 192^3-voxel blocks. For resampled inputs, Merlin and CT-FM use the voxel spacings of their original implementations, while Primus-4B uses isotropic 1-mm spacing. For resampled inputs: [a] $1 \times 1 \times 3$ mm; [b] $1.5 \times 1.5 \times 3$ mm; [c] $1 \times 1 \times 1$ mm.

Setting	VE	Dataset		Preproc.	VE input per pass	Block layout	Axial mean pooling	FOV [mm]	VE output grid	Vision tokens
		CT-RATE	Merlin							
Baseline	Primus-1B	✓	✓	Resize	192^3 voxels	–	–	Full	24^3	13,824
	Curia-8S	✓	✓	Resize	512^2 pixels	–	$192 \to 8$	Full	8×32^2	8,192
	CT-FM	✓	✓	Resample[a]	192^3 voxels	–	–	$192 \times 192 \times 576$	12^3	1,728
	Merlin	✓	✓	Resample[b]	$256^2 \times 160$ voxels	–	–	$384 \times 384 \times 480$	$8^2 \times 10$	640
Higher-resolution	Primus-4B	✓	✗	Resample[c]	192^3 voxels	$2 \times 1 \times 2$	–	$384 \times 192 \times 384$	4×24^3	55,296
		✗	✓	Resize		$2 \times 2 \times 1$	–	Full		
	Curia-64S	✓	✓	Resize	512^2 pixels	–	$192 \to 64$	Full	64×32^2	65,536

dataset-specific ROI definitions are provided in Section 3.1. ROI-cropping removes non-body background and reduces the spatial extent allocated to irrelevant regions, allowing the target anatomy to be represented at a higher effective spatial resolution within the fixed encoder input and vision-token budget.

2.3.2 Vision Encoders

To assess whether projector recommendations are robust to backbone choice, we benchmark four medical imaging foundation VEs: Primus [20], Curia [21], Merlin [8], and CT-FM [29]. Together, they span CNN- and transformer-based architectures, 2D and 3D inputs, and diverse pretraining objectives and scales. Their architectural and pretraining characteristics are summarized in Table 2, while the encoder-specific preprocessing and resulting feature-grid dimensions are reported in Table 1. Apart from the input-size adjustments described below, we retain the original preprocessing pipelines of Curia, CT-FM, and Merlin.

For the CNN-based VEs Merlin and CT-FM, we extract the bottleneck feature maps prior to the task-specific projection head to serve as volumetric token sequences. To ensure sufficient anatomical coverage, we increase the input sizes of CT-FM and Merlin from $128^2 \times 24$ to 192^3 voxels and from $224^2 \times 160$ to $256^2 \times 160$ voxels, respectively, thereby enlarging their effective FOVs. For CT-FM, we additionally explored resizing-based preprocessing and larger FOVs, but neither improved performance on CT-RATE. We therefore retain the 192^3 input throughout.

For Primus, we use the output tokens of the final transformer layer, which form a dense 3D token grid. The baseline configuration processes the volume as a single 192^3 block and is denoted *Primus-1B*, where "B" indicates the number of spatial blocks.

Curia is a 2D ViT that encodes axial slices independently. Each volume is resized to 192 axial slices with an in-plane resolution of 512^2 pixels, matching

Table 2: **Comparison of pretrained vision encoders.**

Model	Arch.	Param.	Dim.	Obj.	Data
Primus	Primus-M	150.0M	3D	MAE [30]	2.2M vol.
Curia	ViT-B	86.1M	2D	DINOv2 [31]	200M sl.
Merlin	ResNet152	121.9M	3D	CLIP [32]	25k CTs
CT-FM	SegResNet	77.8M	3D	SimCLR [33]	148k CTs

the input size of the original Curia implementation. Axial mean pooling then aggregates the slice-level representations into eight axial feature slices, defining the baseline configuration *Curia-8S*, where "S" denotes the number of retained axial feature slices after pooling. For Curia-8S, axial pooling already reduces the representation to only eight slices before projection, so projector compression is restricted to the transverse dimensions ($r_z = 1$).

2.4 Higher-Resolution Visual Representations

2.4.1 Primus-4B: Multi-Block Encoding

To increase the effective spatial resolution relative to the single-block Primus-1B configuration, the cropped ROI is mapped to a four-block input grid by either whole-volume resizing or resampling to an isotropic voxel spacing of $1\times1\times1$ mm. This grid is partitioned into four contiguous, non-overlapping 192^3-voxel blocks, each of which is encoded independently by the Primus VE. The blockwise feature grids are reassembled according to their positions in the original tiling, thereby preserving the global spatial arrangement of the four blocks. We refer to this four-block approach as *Primus-4B*.

2.4.2 Curia-64S: Denser Axial Features

While Primus-4B increases resolution through multi-block encoding, the 2D ViT architecture of Curia enables fine-grained resolution scaling along the vertical/longitudinal (z) axis. Specifically, by reducing the kernel and stride of the axial mean pooling operation, we aggregate the 192 resampled 2D slices into 64 axial feature maps rather than the baseline 8. We designate this configuration as *Curia-64S*, which preserves richer anatomical detail along the axial dimension and results in an eightfold increase in the total number of vision tokens.

2.5 Language Models

We evaluate five instruction-tuned language models spanning 1.7B–4B parameters: Qwen3-1.7B [34], Qwen3-VL-2B and Qwen3-VL-4B [25], Phi-4-mini [35], and Gemma-3-4B [36]. Qwen3-1.7B and Phi-4-mini are text-only models, whereas Qwen3-VL-2B, Qwen3-VL-4B, and Gemma-3-4B are initialized from multimodally pretrained vision-language models. For the multimodal models, we replace the

native vision pathway with our frozen vision encoder and vision-language projector. For Qwen3-VL, we leverage its multimodal rotary positional embeddings (M-RoPE) [25] by mapping the axial dimension to the temporal axis for grid-based adapters. For query-based adapters, whose outputs are spatially unstructured, we use the text formulation, under which M-RoPE reduces to standard one-dimensional RoPE.

3 Experiments

3.1 Datasets

As our development dataset, we use *CT-RATE* [7], comprising 50,188 non-contrast chest CT volumes (25,692 CT-report pairs from 21,304 patients), with multiple volumes per study reconstructed using different kernels. Reports contain a free-text *Findings* section and an optional *Impression* summary. We exclude 829 scans with a TotalSegmentator-estimated lung volume below 1 L [28] and two volumes whose associated report lacks a *Findings* section. Because no official test split is provided, we use the official validation set for testing and create a patient-disjoint validation split from the training data. The resulting splits contain 22,590, 1,530, and 1,564 reports for training, validation, and testing, respectively.

For CT-RATE, the ROI is the axis-aligned bounding box enclosing the union of the five TotalSegmentator lung-lobe masks, expanded by 4% of the side length along each axis. This region covers the pulmonary and cardiovascular abnormalities targeted by the downstream evaluation while substantially reducing the input dimensions.

For cross-dataset evaluation, we use *Merlin* [8], which contains 25,494 abdominal CT-report pairs. We retain the official test set, remove eight duplicate training entries, and reserve 1,000 volumes for validation, yielding 19,361, 1,000, and 5,125 training, validation, and test cases, respectively. Merlin differs substantially from CT-RATE in anatomical region, report structure, and clinical findings. Because its evaluation spans most abdominal organs, as described in Section 3.2, we define a whole-body ROI from the union of all TotalSegmentator foreground classes. After removing small disconnected components, we compute the foreground bounding box and expand it by 4% along each axis. The resulting shifts in image dimension distributions for both datasets are illustrated in Figure 2.

3.2 Evaluation Metrics

We use clinical macro F1 as the primary evaluation metric throughout all model-selection, hyperparameter-tuning, and ablation experiments. Lexical metrics are reported only for the final comparison of our best-performing configurations with existing methods and do not inform any development decisions. Commonly used lexical metrics primarily measure surface-level similarity to reference reports and

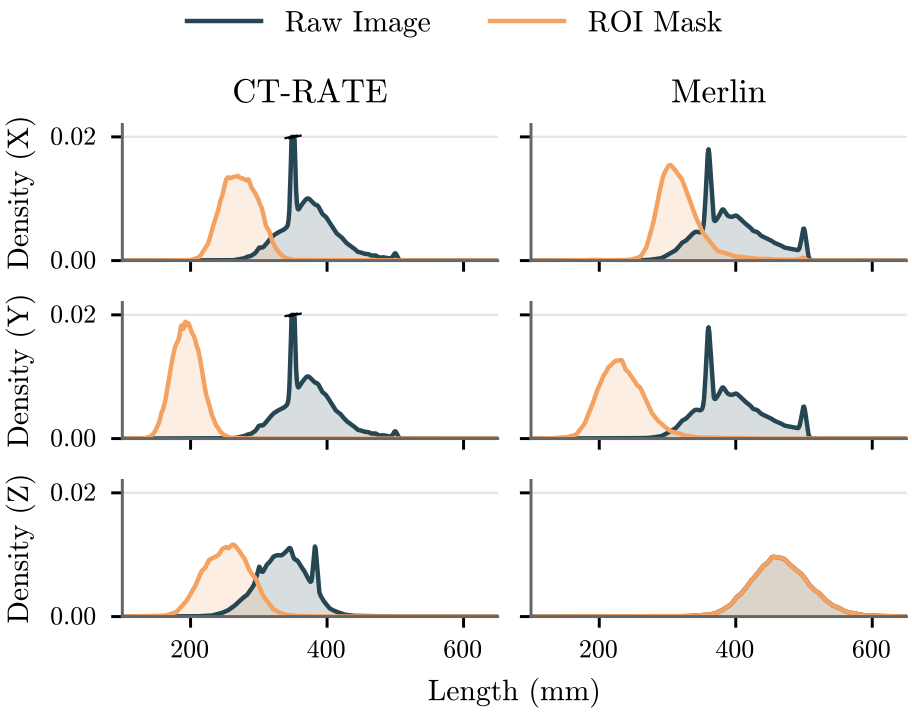

Figure 2: **Image dimension distributions before and after ROI cropping.** Distributions along the X, Y, and Z axes are shown for the raw scans (dark blue) and cropped ROIs (orange). CT-RATE uses a lung ROI, while Merlin uses a foreground ROI.

do not account for the clinical relevance or correctness of the generated content, making them unreliable indicators of medical report quality [37, 38].

3.2.1 Clinical Evaluation

For CT-RATE, we use the fine-tuned RadBERT classifier provided by Hamamci et al. [7, 39]. The classifier predicts 18 pulmonary and cardiovascular abnormalities from free-text radiology reports. We apply it to both the generated and reference reports and compute the macro F1 score across all 18 classes.

Because no established compute-efficient clinical evaluation model is available for Merlin, we fine-tune ModernBERT [22] (chosen over RadBERT [39] for its longer context, as ∼45% of Merlin reports exceed 512 tokens) as a multi-label classifier for the 30 abnormality classes of Blankemeier et al. [8], using labels generated by GPT-OSS-120B [23] instead of the original regular-expression-based extraction, inspired by [40]. Trained on 20,361 reports (official training and validation splits), the classifier achieves a macro F1 of 0.957 against the GPT-OSS labels on the held-out official test split (5,125 reports). At evaluation time we

apply it to the generated reports and compare against the GPT-OSS labels of the reference reports, reporting the macro F1 score across all 30 classes.

3.2.2 Lexical Evaluation

We additionally report BLEU-1 through BLEU-4 [41], METEOR [42], and ROUGE-L [43] using the implementations provided by the `pycocoevalcap` library [44, 45].

3.3 Training Setup

All models are trained using a next-token prediction objective with teacher forcing over the report text, conditioned on the vision tokens produced by the frozen VE and trainable projector. Unless stated otherwise, we use the instruction-tuned Qwen3-VL-2B [25] as the shared LLM backbone across all experiments. The projector is fully fine-tuned, while the LLM is adapted using LoRA [46] with r=32, α=64, and dropout = 0.1. Optimization uses AdamW [47] with β_1=0.9, β_2=0.999, and no weight decay [5,27]. Models are trained with a batch size of 24 for 16 epochs on CT-RATE and 32 epochs on Merlin, reflecting Merlin's approximately twofold smaller training set. We perform validation after every epoch on CT-RATE and every two epochs on Merlin. For each validation run, reports are generated and evaluated using the clinical macro F1 score. The checkpoint achieving the highest validation F1 is selected for subsequent evaluation.

Training was performed on four-GPU nodes equipped with NVIDIA A100 40 GB or NVIDIA H100 80 GB GPUs, with individual runs using between 4 and 12 GPUs. The single longest training run was the Primus MLP baseline, requiring roughly 240 H100 GPU-hours.

We adopt a two-stage linear warm-up schedule inspired by Wald et al. [48]. In the first phase, only the projector is optimized to establish initial vision–language alignment. In the second phase, the LoRA parameters are unfrozen and jointly optimized with the projector using a separate linear warm-up. Each warm-up phase spans 6.25% (1/16) of the total optimization steps. For the PerceiverResampler, the initial projector-only warm-up is extended to 25% (4/16), as preliminary experiments indicated that the standard duration was insufficient. After the two warm-up phases, the learning rate follows a cosine decay over the remaining steps.

3.4 Hyperparameter Selection

To limit computational cost and ensure a consistent search procedure across projectors, we perform the full hyperparameter search on the CT-RATE validation set using Primus-1B at an 8× token-compression ratio, corresponding to $\mathbf{r} = (2, 2, 2)$. For each projector, we evaluate the projector/LLM learning-rate pairs $(3\times10^{-3}, 3\times10^{-4})$, $(1\times10^{-3}, 3\times10^{-4})$, $(3\times10^{-4}, 3\times10^{-4})$, $(3\times10^{-4}, 1\times10^{-4})$, and $(1\times10^{-4}, 1\times10^{-4})$.

Table 3: **Resolution and ROI-cropping configurations under a fixed vision-token budget. r** denotes the grid-based reduction vector; query-based projectors use $L = M$ queries.

VE	Regime	Configuration	ROI	r	M
Primus	Whole	Primus-1B	✗	$(2, 2, 2)$	1,728
	ROI	Primus-1B	✓	$(2, 2, 2)$	
	High Res. (CT-RATE)	Primus-4B	✓	$(4, 2, 4)$	
	High Res. (Merlin)	Primus-4B	✓	$(4, 4, 2)$	
Curia	Whole	Curia-8S	✗	$(2, 2, 1)$	2,048
	ROI	Curia-8S	✓	$(2, 2, 1)$	
	High Res.	Curia-64S	✓	$(2, 2, 8)$	

We compare projector depth by evaluating 2- and 4-layer projection MLPs for all projectors except the PerceiverResampler. For the PerceiverResampler, depth instead denotes the number of Perceiver layers, and we evaluate depths of three and six.

For each token-reducing projector–depth combination, we retain the three learning-rate pairs achieving the highest validation macro F1 at 8× compression and evaluate them at the remaining Primus compression ratios of 27× and 64×. We then select the best-performing learning-rate and depth combination for each projector and compression ratio.

For each remaining VE, we transfer the learning rate and projector depth from the Primus configuration at the corresponding or nearest available compression ratio. No additional hyperparameter tuning is performed for other VEs or on Merlin, maintaining a consistent budget across comparisons.

3.5 Experiment Design

We conduct experiments on the CT-RATE and Merlin datasets. Unless otherwise stated, reported results are averaged over three seeds and presented as mean ± standard deviation.

3.5.1 Resolution and ROI Cropping

We evaluate three input regimes, denoted *Whole*, *ROI*, and *High Res.* *Whole* encodes the complete resized volume without cropping, whereas *ROI* applies the dataset-specific anatomical cropping procedure described in Section 2.3.1. *High Res.* combines ROI cropping with increased effective resolution through blockwise encoding with Primus-4B or reduced post-encoding axial mean pooling with Curia-64S. The corresponding configurations are summarized in Table 3.

For AveragePool, Curia-64S *High Res.* is equivalent to Curia-8S *ROI*, as projector pooling reproduces the preceding axial mean pooling.

Table 4: **Effect of input field-of-view (FOV) / resolution.** Validation-set, seed-averaged macro F1 scores [%] for increasing resolution (Whole → ROI → High Res.), with standard deviation across seeds (± std). The number of vision tokens M forwarded to the LLM is fixed across FOV / resolution settings for each VE (cf. Tables 1 and 3). Best per dataset and VE in **bold**; color coding is applied per dataset and VE. "= *ROI*": equivalent to the corresponding ROI setting.

		Primus			Curia		
Type	**Projector**	**Whole**	**ROI**	**High Res.**	**Whole**	**ROI**	**High Res.**
		CT-RATE					
Grid-based	AveragePool	44.6 ± 0.5	**47.3** ± 0.7	44.9 ± 1.5	47.8 ± 0.5	49.0 ± 0.1	= *ROI*
	ConcatAdapter	43.2 ± 0.7	**47.3** ± 0.8	43.8 ± 0.4	48.0 ± 0.4	48.0 ± 0.8	46.9 ± 1.5
	TokenPacker	43.3 ± 0.5	45.9 ± 0.2	44.6 ± 1.0	47.4 ± 1.2	48.3 ± 1.0	48.5 ± 0.7
Query-based	AttentionPool	40.6 ± 1.3	43.6 ± 1.3	41.0 ± 0.7	44.8 ± 0.3	46.8 ± 0.4	44.7 ± 0.2
	PercResampler	44.8 ± 0.6	46.4 ± 0.5	46.1 ± 0.9	48.1 ± 0.4	48.7 ± 0.4	**49.8** ± 1.0
		Merlin					
Grid-based	AveragePool	40.6 ± 0.8	45.8 ± 0.7	44.4 ± 1.6	44.8 ± 1.0	45.0 ± 0.3	= *ROI*
	ConcatAdapter	37.8 ± 1.3	42.6 ± 1.6	33.4 ± 0.9	43.7 ± 0.6	45.0 ± 2.0	43.3 ± 0.7
	TokenPacker	42.9 ± 0.5	46.0 ± 1.6	**48.3** ± 1.8	42.7 ± 1.3	44.5 ± 0.9	46.9 ± 0.9
Query-based	AttentionPool	32.8 ± 1.2	37.1 ± 1.2	31.6 ± 2.7	39.4 ± 1.8	41.6 ± 2.3	41.4 ± 1.5
	PercResampler	38.6 ± 0.6	44.5 ± 0.4	42.0 ± 2.3	43.3 ± 1.0	44.4 ± 0.7	**49.0** ± 1.7

3.5.2 Token Compression

To characterize how projector performance varies with compression strength, we evaluate token-reducing projectors at multiple encoder-specific reduction levels on ROI-cropped inputs. Spatial reduction is applied uniformly across all three axes for 3D encoders, but restricted to the two transverse axes for Curia-8S. Because native token counts differ, evaluated compression ratios vary across backbones: 8× to 64× for Primus-1B, 4× to 64× for Curia-8S, 8× and 27× for CT-FM, and 8× for Merlin. The token-preserving MLP is included for every encoder as an uncompressed reference with $M = N$.

3.5.3 LLM Choice

To quantify the influence of the LLM on downstream performance, we evaluate multiple LLMs while holding the visual configuration fixed. For each VE-dataset pair, we retain the best-performing configuration from the resolution and ROI-cropping study. In addition to Qwen3-VL-2B, which is used throughout model development, we evaluate Qwen3-1.7B, Qwen3-VL-4B, Phi-4-mini, and Gemma3-4B.

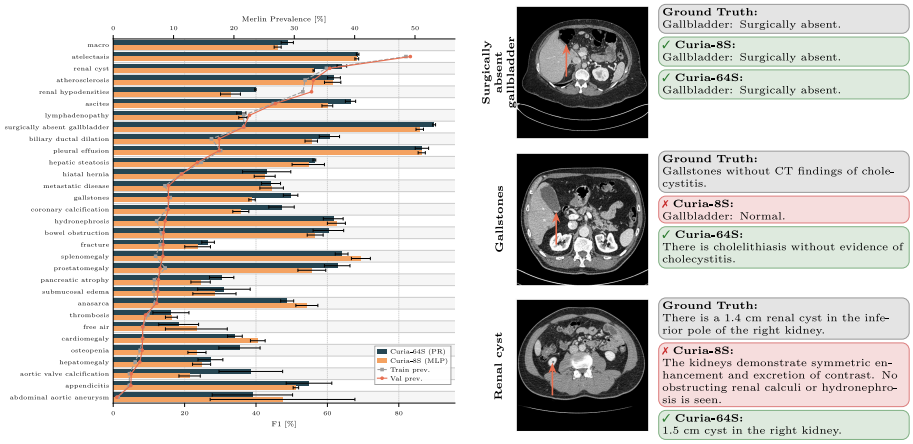

Figure 3: **Class-wise and qualitative results.** (left) Macro F1 (first row) and class-wise F1 scores [%] for Curia-64S PerceiverResampler in dark blue (cf. Table 4) and Curia-8S MLP in orange (cf. Table 5) on the 30 classes from the Merlin dataset. All results are seed averaged including ± std. Classes are sorted by validation set prevalence. (right) Qualitative example of Curia-64S catching the small gallstone and renal cyst indicated by the red arrows while the lower resolution Curia-8S misses those small pathologies, highlighting the importance of resolution.

4 Results

4.1 Impact of Resolution and ROI Cropping

As summarized in Table 4, cropping to the ROI improves F1 over the *Whole* baseline in 19 of 20 configurations. The magnitude of this gain, however, differs systematically between the two encoders. Averaged across all projectors and both datasets, the gain from *Whole* to *ROI* is smaller for Curia than for Primus (+1.14 vs. +3.72). This difference may reflect that Curia is already operating near native in-plane resolution, which limits the additional detail afforded by cropping, whereas Primus benefits more directly from the resulting increase in effective resolution. For Primus, the average gain from *Whole* to *ROI* is larger on Merlin than on CT-RATE (+4.63 vs. +2.82).

On CT-RATE, the *High Res.* regime does not yield a further improvement for Primus. Its best *High Res.* run, obtained with the PerceiverResampler (46.1), remains below the best *ROI* runs (47.3). We interpret this as a resolution ceiling effect: with a crop size of 192 mm, the lung ROI already fits within a single Primus block at close to the CT's original resolution. For Curia, *High Res.* is more beneficial, achieving the best overall CT-RATE result with the PerceiverResampler (49.8) and suggesting that reduced compression in z-slice aggregation preserves useful axial detail. This benefit is projector-dependent, though: TokenPacker is essentially flat (48.3 → 48.5), whereas ConcatAdapter

Table 5: **Effect of token compression.** Validation-set, seed-averaged macro F1 scores [%] in the *ROI* input regime, with standard deviation across runs (± std). Best per compression ratio and VE in **bold**; color coding per metric, dataset, and VE. *MLP* rows use no token compression; each is shown once per encoder in the central compression column, in italic.

				Primus			Curia			CT-FM		Merlin
Metric	**Dataset**	**Type**	**Projector**	8×	27×	64×	4×	16×	64×	8×	27×	8×
Macro F1	CT-RATE	MLP	*(no compression)*	*48.9* ± 1.1			*48.5* ± 0.8			*35.6* ± 0.6		*32.0* ± 0.9
		Grid-based	AveragePool	**47.3** ± 0.7	**46.4** ± 0.1	45.0 ± 0.6	**49.0** ± 0.1	47.1 ± 2.0	44.7 ± 0.7	**35.6** ± 0.7	34.4 ± 0.6	31.9 ± 0.3
			ConcatAdapter	**47.3** ± 0.8	44.7 ± 1.1	43.0 ± 1.2	48.0 ± 0.8	44.0 ± 0.5	42.2 ± 0.3	35.2 ± 0.6	33.8 ± 0.5	**33.0** ± 0.3
			TokenPacker	45.9 ± 0.2	46.0 ± 0.7	44.5 ± 1.1	48.3 ± 1.0	47.0 ± 0.6	44.1 ± 0.9	33.7 ± 2.1	**35.4** ± 0.9	31.5 ± 0.3
		Query-based	AttentionPool	43.6 ± 1.3	43.3 ± 1.6	43.8 ± 0.8	46.8 ± 0.4	44.4 ± 1.3	44.5 ± 0.6	34.5 ± 0.8	32.0 ± 0.8	31.1 ± 0.4
			PercResampler	46.4 ± 0.5	46.0 ± 0.3	**46.1** ± 0.3	48.7 ± 0.4	**48.8** ± 0.7	**49.2** ± 0.4	35.2 ± 0.3	35.1 ± 0.3	29.6 ± 0.4
	Merlin	MLP	*(no compression)*	*49.2* ± 1.0			*46.3* ± 1.0			*28.2* ± 1.5		*33.2* ± 0.9
		Grid-based	AveragePool	45.8 ± 0.7	41.4 ± 1.2	38.0 ± 1.5	**45.0** ± 0.3	41.3 ± 1.0	36.3 ± 1.5	**28.7** ± 0.9	26.0 ± 1.1	**33.4** ± 0.9
			ConcatAdapter	42.6 ± 1.6	35.3 ± 0.5	31.4 ± 1.5	45.0 ± 2.0	38.2 ± 0.3	34.0 ± 0.5	26.8 ± 2.1	22.6 ± 0.4	33.2 ± 0.7
			TokenPacker	**46.0** ± 1.6	**46.4** ± 0.9	**43.0** ± 0.6	44.5 ± 0.9	43.7 ± 1.4	39.6 ± 0.4	27.2 ± 1.9	25.8 ± 1.3	33.1 ± 0.4
		Query-based	AttentionPool	37.1 ± 1.2	39.7 ± 2.5	40.0 ± 1.3	41.6 ± 2.3	39.9 ± 1.6	40.4 ± 0.7	21.2 ± 1.2	22.0 ± 1.1	32.4 ± 0.5
			PercResampler	44.5 ± 0.4	43.1 ± 1.7	42.5 ± 0.6	44.4 ± 0.7	**45.1** ± 0.1	**42.9** ± 1.2	26.8 ± 1.2	**26.2** ± 0.6	32.7 ± 1.3

(48.0 → 46.9) and AttentionPool (46.8 → 44.7) degrade.

On Merlin, the *High Res.* regime is more consistently beneficial, although its effect remains projector-dependent. For Primus, TokenPacker achieves the best result overall (46.0 → 48.3), while the remaining projectors do not improve. For Curia, the PerceiverResampler gives the strongest result under *High Res.* over *ROI* (44.4 → 49.0), and TokenPacker likewise improves (44.5 → 46.9). By contrast, neither AveragePool, ConcatAdapter, nor AttentionPool benefits.

These results indicate that exploiting the additional input resolution depends on the projector's ability to aggregate a larger set of encoder tokens under stronger compression. Because the output token budget is held fixed, all *High Res.* configurations use a compression ratio of 32×. The favorable results are concentrated in TokenPacker and the PerceiverResampler, suggesting that their aggregation mechanisms are better suited to preserve fine-grained information. In contrast, ConcatAdapter is particularly negatively affected because concatenating all tokens within each reduction block causes the input dimension and parameter count of the subsequent projection layer to scale with the compression ratio, an effect examined further in Section 4.2.

Finally, a per-class analysis (Figure 3) indicates that the benefits of the *High Res.* regime are especially evident for spatially small pathologies, for which the finer resolution available to the projector is most beneficial. At the fixed output token budget described above, *High Res.* enables the projector to distill finer spatial detail into the same number of tokens. Specifically, renal cysts and gallstones on Merlin benefit significantly from the increased resolution.

4.2 Impact of Token Compression

Table 5 reports performance across the evaluated compression ratios. For the ViT-based encoders, token-reducing projectors exceed the uncompressed MLP reference only in a few cases. For Primus, the MLP achieves the best overall result on both datasets (48.9 on CT-RATE and 49.2 on Merlin), and no token-reducing projector matches its performance. On CT-RATE, AveragePool and

ConcatAdapter yield the strongest compressed results at 8× (47.3), whereas the PerceiverResampler performs best at 64× (46.1). On Merlin, TokenPacker is the strongest compressed projector at every evaluated ratio, achieving 46.0, 46.4, and 43.0 at 8×, 27×, and 64×, respectively.

Curia exhibits a different pattern. The PerceiverResampler is the strongest projector at higher compression ratios and surpasses the uncompressed MLP reference on CT-RATE at 64× (49.2 vs. 48.5). On Merlin, it likewise performs best among the token-reducing projectors at 16× and 64× (45.1 and 42.9, respectively), but remains below the MLP reference (46.3). These results are consistent with the analysis in Section 4.1, where TokenPacker and the PerceiverResampler likewise proved most effective at preserving useful information under strong compression in the *High Res.* regime.

The clearest failure mode under aggressive compression is observed with the ConcatAdapter, whose input dimension and parameter count grow with the compression ratio, as previously noted in Section 4.1. For Primus, the ConcatAdapter increases from 39.4 million parameters at 8× compression to 369.2 million at 64×, corresponding to a 9.4× increase despite capping the first hidden dimension. This substantial growth increases the projector's optimization burden and may contribute to its sensitivity to aggressive compression. In additional experiments, an extended projector-only warm-up did not improve performance, indicating that insufficient warm-up alone does not explain the degradation. On Merlin, ConcatAdapter performance drops by more than 10 F1 points over the same compression range for both Primus and Curia. AveragePool exhibits a similar decline, likely because averaging over large token groups removes fine-grained spatial information that cannot be recovered by the subsequent projection. At the lowest compression ratio, however, it remains highly competitive and achieves some of the strongest results overall.

The CNN-based encoders yield substantially lower absolute scores than Primus and Curia on both datasets. This may reflect the stronger spatial downsampling already performed by the CNN backbones, which produces shorter token sequences and may leave less fine-grained information for the projector to preserve. At 8× compression, AveragePool is strongest in three of the four encoder–dataset combinations and matches the corresponding uncompressed MLP reference in all four. The exception is the Merlin encoder on CT-RATE, where ConcatAdapter performs best and surpasses the MLP (33.0 vs. 32.0). At 27× with CT-FM, TokenPacker is strongest on CT-RATE (35.4), whereas the PerceiverResampler is strongest on Merlin (26.2), consistent with their favorable behavior under stronger compression observed above.

Across settings with multiple compression ratios, AttentionPool varies comparatively little as compression increases, a pattern also observed for the PerceiverResampler. This stability may represent an advantage of query-based projectors: learned queries aggregate information adaptively from the full encoder feature set rather than summarizing increasingly large fixed spatial blocks. AttentionPool nevertheless underperforms compared to the other token-reducing projectors, suggesting that a single cross-attention layer may be insufficient to extract and refine the relevant visual information. The stronger results of the

Table 6: **Effect of the language model.** For each dataset, the vision path is fixed to the best configuration from Table 4, indicated in parentheses after each vision encoder (HR: *High Res.*; AP: AveragePool; TP: TokenPacker; PR: PerceiverResampler). Values report validation-set seed-averaged macro F1 [%] with standard deviation across seeds (± std). The best LLM per row is shown in **bold**; cells are color-coded within rows.

VE, (FOV/Res., projector)	Language Model				
	Qwen3 1.7B	**Qwen3VL 2B**	**Phi-4 mini**	**Gemma3 4B**	**Qwen3VL 4B**
	CT-RATE				
Primus (*ROI*, AP)	46.5 ± 0.2	**47.3** ± 0.7	43.8 ± 0.9	45.1 ± 0.7	44.1 ± 0.7
Curia (*HR*, PR)	49.2 ± 0.5	49.8 ± 1.0	47.9 ± 0.4	**50.1** ± 0.6	49.2 ± 1.5
	Merlin				
Primus (*HR*, TP)	44.2 ± 4.1	48.3 ± 1.8	45.8 ± 1.8	20.5 ± 0.8	**49.5** ± 0.3
Curia (*HR*, PR)	46.5 ± 1.9	**49.0** ± 1.7	46.6 ± 1.1	46.3 ± 0.6	48.1 ± 1.6

PerceiverResampler indicate that deeper, iterative query-based aggregation is needed to combine robustness to compression with high predictive performance.

4.3 Impact of LLM Choice

Results are reported in Table 6, with the vision path fixed to its best configuration from Section 4.1. Excluding one failed run discussed below, performance varies by only 2.2–5.3 F1 across LLMs for a fixed dataset and VE, substantially less than the more than 10-point gap between the ViT- and CNN-based encoders in Table 5. Within the evaluated design space, the VE appears to be the more influential component.

Qwen3-1.7B and Qwen3-VL-2B provide a near-controlled comparison of language-only and vision–language pretraining, as their language backbones are parameter-identical and the original vision tower of Qwen3-VL-2B is replaced by our medical VE. Qwen3-VL-2B nevertheless performs better in all four settings, by 0.6–0.8 on CT-RATE and 2.5–4.1 on Merlin. This suggests that vision–language pretraining improves the language backbone's ability to consume visual embeddings even when the original VE is discarded. Parameter count alone is not predictive: the larger Phi-4-mini does not outperform Qwen3-VL-2B in any setting, while Gemma-3-4B and Qwen3-VL-4B each improve upon it in only one of four settings.

These comparisons remain subject to transferred hyperparameters. Qwen3-VL-2B served as the development LLM, and repeating the learning-rate search for every backbone was computationally infeasible. The remaining results may therefore underestimate the performance attainable with model-specific tuning. This issue is most apparent for Gemma-3-4B with Primus on Merlin, which reproducibly collapses to 20.5 F1 despite performing strongly elsewhere, high-

Table 7: **Comparison with the literature.** We report seed-averaged test-set results for two configurations: Curia-64S with the PerceiverResampler at 32× compression, using the best language model for each dataset, and Primus-1B with an MLP. For CT-RATE baselines, we report published metrics. For the Merlin and Jolia baselines, we apply our evaluation framework to their official test-set reports. Our Merlin configurations are evaluated on the *Findings* section and full report. Values are in [%]; the best per column and dataset is shown in **bold**. Missing entries indicate metrics not reported in the respective publication.

		Clinical					Lexical					
Dataset	**Model**	**F1**	**Bal-Acc**	**Acc**	**Recall**	**Prec.**	**BLEU-1**	**BLEU-2**	**BLEU-3**	**BLEU-4**	**METEOR**	**ROUGE-L**
CT-RATE	**Ours** (Curia-64S + PercResampler, Gemma-3-4B)	**49.5**	**68.4**	83.1	46.7	52.8	46.0	33.3	25.4	20.3	23.5	28.0
	Ours (Primus-1B + MLP, Qwen3-VL-2B)	49.0	68.0	**84.3**	43.8	**57.0**	42.9	31.6	24.7	20.2	22.8	31.0
	AdaRAG-CT (Llama-3.1) [24]	48.0	–	–	52.0	50.2	49.6	–	–	24.2	**24.6**	**35.4**
	COLIPRI-CRM [5]	44.9	–	–	–	–	–	–	–	–	–	–
	Ker-VLJEPA-3B (Phase 4) [49]	42.9	–	–	**52.4**	38.9	–	–	–	–	–	–
	CT-Agent [50]	42.0	–	–	47.7	42.3	**50.2**	**37.4**	29.0	23.1	–	–
	U-VLM [51]	41.4	–	–	42.9	49.1	47.4	36.7	**30.0**	**25.6**	–	–
	BTB3D [13]	25.8	–	–	26.0	26.0	43.9	32.0	24.8	21.3	22.3	–
Merlin (Findings)	**Ours** (Curia-64S + PercResampler, Qwen3-VL-2B)	46.8	68.4	**89.5**	42.1	54.7	35.1	**24.4**	**18.1**	**14.3**	**18.3**	**31.6**
	Ours (Primus-1B + MLP, Qwen3-VL-2B)	**49.0**	**69.5**	**89.5**	**44.4**	**56.3**	**35.3**	24.0	17.5	13.5	18.2	30.1
	Jolia [14]	23.8	58.4	87.5	21.4	39.5	13.7	10.0	7.8	6.4	14.1	29.3
	Merlin [8]	9.7	53.0	87.5	9.6	21.6	10.9	7.8	5.8	4.6	12.9	25.8
Merlin (Full report)	**Ours** (Curia-64S + PercResampler, Qwen3-VL-2B)	48.3	69.5	**89.6**	44.4	54.2	33.0	**22.3**	**16.4**	**12.8**	**17.0**	**29.0**
	Ours (Primus-1B + MLP, Qwen3-VL-2B)	**50.0**	**70.3**	**89.6**	**46.3**	**55.3**	**33.6**	22.2	15.9	12.1	16.9	27.6

lighting the need to retune the language-side optimization when exchanging the LLM.

4.4 Comparison to State-of-the-art

Table 7 compares our best configurations against published results on the held-out test splits. We report the strongest token-reducing configuration, Curia-64S with the PerceiverResampler at 32× compression, alongside the strongest uncompressed baseline, Primus-1B with an MLP. Gemma-3-4B is used for the token-reducing configuration on CT-RATE, while Qwen3-VL-2B is used otherwise.

On CT-RATE, Curia-64S with the PerceiverResampler achieves the highest F1 among the compared systems (49.5), exceeding AdaRAG-CT (48.0). The Primus MLP configuration attains the highest precision (57.0), while the recall of the Curia configuration (46.7) remains below that of Ker-VLJEPA-3B (52.4) and AdaRAG-CT (52.0). These results are obtained through end-to-end report generation, without the retrieval augmentation used by AdaRAG-CT [24] or the multi-stage agentic refinement used by CT-Agent [50]. Such mechanisms are orthogonal to the projector design studied here and could plausibly be combined with our recipe for potential further gains. CT-Agent and U-VLM obtain the highest BLEU scores, while AdaRAG-CT leads METEOR and ROUGE-L (24.6 and 35.4). Nevertheless, CT-Agent and U-VLM trail our Curia configuration by 7.5 and 8.1 F1, respectively. This divergence is consistent with the limitations discussed in Section 3.2: n-gram overlap rewards similarity to the reference report style, but does not directly measure whether clinically relevant findings are reported correctly.

On the Merlin findings-only evaluation, our models (46.8, 49.0) outperform

Jolia and Merlin (23.8, 9.7) across all reported metrics. Evaluating the full report further raises F1 for Curia and Primus (46.8 → 48.3 and 49.0 → 50.0) and improves recall, although precision decreases slightly, suggesting that clinically relevant labels are also conveyed outside the organ-wise *Findings* section. Primus with the MLP achieves stronger clinical performance, whereas Curia with the PerceiverResampler performs better on every lexical metric except BLEU-1.

5 Discussion, Limitations & Conclusion

This work asks where the vision-token budget is best spent. Cropping to the target anatomy raises the effective resolution at an unchanged token budget and improves clinical macro F1 in 19 of 20 VE-projector-dataset cells, by 3.7 F1 for Primus and 1.1 for Curia on average, which makes it the most consistent intervention in this study (Table 4). Resolution beyond that point is not free. When the emitted token budget is held fixed, a larger input must be compensated by a higher compression ratio and results in an average performance decrease of 1.2 F1 on CT-RATE and 1.1 F1 on Merlin. Only TokenPacker and the PerceiverResampler benefit from the higher-resolution regime, with their aggregation mechanisms better preserving useful information under stronger compression. Together, they yield the best results on 3 out of 4 VE-dataset configurations. We employ a lung-crop on CT-RATE specifically to ablate the effects of resolution in RRG: a lung-crop suits the closed-set clinical question we evaluate, but it also removes anatomy the reference reports describe, which can encourage statements about structures the input no longer contains.

Within the VLM design, the VE is the dominant axis. Averaged over all projectors and compression ratios, the spread between the best and the weakest VE is 15.0 F1 on CT-RATE and 16.1 on Merlin. The Merlin VE performs worse than CT-FM on CT-RATE but better on its in-domain dataset, underscoring the role of both architecture and pre-training data. Across our four VEs, architecture, pretraining objective, pretraining scale and native token count all covary: both ViT encoders are pretrained by masking or self-distillation on 2.2M volumes and 200M slices and emit 8k–14k tokens, whereas both CNN encoders are pretrained contrastively on 25k–148k volumes and emit 640–1,728. Separating these factors would require pretraining matched architectures on a common corpus, which was out of scope here on compute grounds; we consider it a worthwhile target for future investigations.

Projector choice has a smaller effect overall: averaged over VEs and compression ratios, the difference between the most and least effective projector is 2.35 F1 on CT-RATE and 4.47 on Merlin. One key finding is that most token-reducing projectors do not match the token-preserving MLP. We therefore recommend using it as the reference when evaluating projectors. Where compression is required, the choice is VE-dependent: The PerceiverResampler exceeds its MLP baseline (49.2 vs. 48.5 at 64× for Curia) while TokenPacker is the strongest compression projector for Primus on Merlin. When excluding

outliers, the LLM choice has an impact similar to that of the projector. Scores within the same VE configuration span 2.2–5.3 F1 points, but the models are not interchangeable: language-side hyperparameters must be retuned whenever it is exchanged, as one transferred configuration collapses reproducibly to 20.5 F1 ($\pm$0.8).

Several additional limitations should be considered when interpreting these results. First, the VE is frozen, so our findings concern adaptation rather than representation learning, and, given its substantial compute cost, the hyperparameter search was conducted only for Primus-1B on CT-RATE, so other configurations are lower bounds and comparisons against the development model are confounded with tuning effort. Second, because both the automated labeler and its underlying LLM-extracted training labels are prone to error, clinical macro F1 measures fidelity to a flawed teacher rather than true clinical ground truth. Nevertheless, as these errors are clinically plausible [40], an F1 score derived from extracted labels yields a substantially more clinically meaningful evaluation than rigid lexical metrics. Third, at around 50% precision and recall these systems compare design choices and are not deployable report generators.

Overall, our results show that substantial improvements can be obtained through careful choices of input preprocessing, vision encoder, and projector. Although hardware is expected to continue increasing computational capacity, trends toward higher-resolution imaging, higher-resolution vision encoders, and larger language models will continue to increase the demand for efficient vision-token utilization. We therefore expect vision-token allocation to remain a central design consideration for scalable 3D RRG.

6 Acknowledgments

This work is supported by the Helmholtz Association Initiative and Networking Fund on the HAICORE@JSC partition.

References

[1] D. Forsberg, B. Rosipko, and J. L. Sunshine, "Radiologists' variation of time to read across different procedure types," *Journal of digital imaging*, vol. 30, no. 1, pp. 86–94, 2017.

[2] S. Bannur, K. Bouzid, D. C. Castro, A. Schwaighofer, A. Thieme, S. Bond-Taylor, M. Ilse, F. Pérez-García, V. Salvatelli, H. Sharma *et al.*, "Maira-2: Grounded radiology report generation," *arXiv preprint arXiv:2406.04449*, 2024.

[3] Q. Liu, S. Zhang, G. Qin, Y. Gu, Y. Jin, S. Preston, Y. Xu, S. Kiblawi, W.-w. Yim, T. Ossowski *et al.*, "Scaling medical imaging report generation with multimodal reinforcement learning," *arXiv preprint arXiv:2601.17151*, 2026.

[4] H. Liu, C. Li, Q. Wu, and Y. J. Lee, "Visual instruction tuning," *Advances in neural information processing systems*, vol. 36, pp. 34 892–34 916, 2023.

[5] T. Wald, I. E. Hamamci, Y. Gao, S. Bond-Taylor, H. Sharma, M. Ilse, C. Lo, O. Melnichenko, A. Schwaighofer, N. C. Codella *et al.*, "Comprehensive language-image pre-training for 3d medical image understanding," *arXiv preprint arXiv:2510.15042*, 2025.

[6] F. Bai, Y. Du, T. Huang, M. Q.-H. Meng, and B. Zhao, "M3d: Advancing 3d medical image analysis with multi-modal large language models," *arXiv preprint arXiv:2404.00578*, 2024.

[7] I. E. Hamamci, S. Er, C. Wang, F. Almas, A. G. Simsek, S. N. Esirgun, I. Dogan, O. F. Durugol, B. Hou, S. Shit, W. Dai, M. Xu, H. Reynaud, M. F. Dasdelen, B. Wittmann, T. Amiranashvili, E. Simsar, M. Simsar, E. B. Erdemir, A. Alanbay, A. Sekuboyina, B. Lafci, A. Kaplan, Z. Lu, M. Polacin, B. Kainz, C. Bluethgen, K. Batmanghelich, M. K. Ozdemir, and B. Menze, "Generalist foundation models from a multimodal dataset for 3D computed tomography," *Nature Biomedical Engineering*, pp. 1–19, Feb 2026. [Online]. Available: https://doi.org/10.1038/s41551-025-01599-y

[8] L. Blankemeier, A. Kumar, J. P. Cohen, J. Liu, L. Liu, D. Van Veen *et al.*, "Merlin: A computed tomography vision-language foundation model and dataset," *Nature*, vol. 652, no. 8112, pp. 1318–1328, 2026.

[9] Z. Chen, Y. Bie, H. Jin, and H. Chen, "Large language model with region-guided referring and grounding for ct report generation," *IEEE transactions on Medical Imaging*, vol. 44, no. 8, pp. 3139–3150, 2025.

[10] H. Kalisch, F. Hörst, J. Kleesiek, K. Herrmann, and C. Seibold, "CT-GRAPH: Hierarchical graph attention network for anatomy-guided ct report generation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*, October 2025, pp. 6834–6843.

[11] A. Sellergren *et al.*, "MedGemma 1.5 technical report," *arXiv preprint arXiv:2604.05081*, 2026.

[12] C. F. Sabottke and B. M. Spieler, "The effect of image resolution on deep learning in radiography," *Radiology: Artificial Intelligence*, vol. 2, no. 1, p. e190015, 2020.

[13] I. E. Hamamci, S. Er, S. Shit, H. Reynaud, D. Yang, P. Guo, M. Edgar, D. Xu, B. Kainz, and B. Menze, "Better tokens for better 3d: Advancing vision-language modeling in 3d medical imaging," *Advances in Neural Information Processing Systems*, vol. 38, pp. 135 074–135 102, 2026.

[14] J. Khlaut, C. Corbière, B. Callard, A. Prat, L. Butsanets, A. Saporta, T. Danielou, L. Machado, K. L. Floch, T. Boeken *et al.*, "Jolia: Concept-level vision-language alignment for 3d ct contrastive learning," *arXiv preprint arXiv:2606.24570*, 2026.

[15] Y. Du, F. Bai, T. Huang, and B. Zhao, "SegVol: Universal and interactive volumetric medical image segmentation," in *Advances in Neural Information Processing Systems*, vol. 37, 2024, pp. 110 746–110 783.

[16] H. Liu, B. Georgescu, Y. Zhang, Y. Yoo, M. Baumgartner, R. Gao, J. Wang, G. Zhao, E. Gibson, D. Comaniciu, and S. Grbic, "Revisiting 2d foundation models for scalable 3d medical image classification," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2026, pp. 30 021–30 031.

[17] C. Wu, X. Zhang, Y. Zhang, H. Hui, Y. Wang, and W. Xie, "Towards generalist foundation model for radiology by leveraging web-scale 2d&3d medical data," *Nature Communications*, vol. 16, no. 1, p. 7866, 2025.

[18] M. Baharoon, J. Ma, C. Fang, A. Toma, and B. Wang, "Exploring the design space of 3d mllms for ct report generation," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 2025, pp. 237–246.

[19] Y. Ji, H. Bai, C. Ge, J. Yang, Y. Zhu, R. Zhang, Z. Li, L. Zhanng, W. Ma, X. Wan *et al.*, "Amos: A large-scale abdominal multi-organ benchmark for versatile medical image segmentation," *Advances in Neural Information Processing Systems*, vol. 35, pp. 36 722–36 732, 2022.

[20] T. Wald, S. Roy, F. Isensee, C. Ulrich, S. Ziegler, D. Trofimova, R. Stock, M. Baumgartner, G. Koehler, and K. Maier-Hein, "Primus: Enforcing attention usage for 3d medical image segmentation," *Transactions on Machine Learning Research*, 2026. [Online]. Available: https://openreview.net/forum?id=x4vZE4PDEu

[21] C. Dancette, J. Khlaut, A. Saporta, H. Philippe, E. Ferreres, B. Callard, T. Danielou, L. Alberge, L. Machado, D. Tordjman *et al.*, "Curia: A multimodal foundation model for radiology," *arXiv preprint arXiv:2509.06830*, 2025.

[22] B. Warner, A. Chaffin, B. Clavié, O. Weller, O. Hallström, S. Taghadouini, A. Gallagher, R. Biswas, F. Ladhak, T. Aarsen *et al.*, "Smarter, better, faster, longer: A modern bidirectional encoder for fast, memory efficient, and long context finetuning and inference," *arXiv preprint arXiv:2412.13663*, 2024.

[23] S. Agarwal, L. Ahmad, J. Ai, S. Altman, A. Applebaum, E. Arbus, R. K. Arora, Y. Bai, B. Baker, H. Bao *et al.*, "gpt-oss-120b & gpt-oss-20b model card," *arXiv preprint arXiv:2508.10925*, 2025.

[24] R. Liang, Y. Ma, Y. Xing, Z. Fan, J. Pan, C. Sun, L. Li, K. Gong, and J. Xu, "Beyond the embedding bottleneck: Adaptive retrieval-augmented 3d ct report generation," *arXiv preprint arXiv:2603.15822*, 2026.

[25] S. Bai, Y. Cai, R. Chen, K. Chen, X. Chen, Z. Cheng, L. Deng, W. Ding, C. Gao, C. Ge *et al.*, "Qwen3-vl technical report," *arXiv preprint arXiv:2511.21631*, 2025.

[26] W. Li, Y. Yuan, J. Liu, D. Tang, S. Wang, J. Qin, J. Zhu, and L. Zhang, "Tokenpacker: Efficient visual projector for multimodal llm," *International Journal of Computer Vision*, vol. 133, no. 10, pp. 6794–6812, 2025.

[27] J.-B. Alayrac, J. Donahue, P. Luc, A. Miech, I. Barr, Y. Hasson, K. Lenc, A. Mensch, K. Millican, M. Reynolds *et al.*, "Flamingo: a visual language model for few-shot learning," *Advances in neural information processing systems*, vol. 35, pp. 23 716–23 736, 2022.

[28] J. Wasserthal, H.-C. Breit, M. T. Meyer, M. Pradella, D. Hinck, A. W. Sauter, T. Heye, D. T. Boll, J. Cyriac, S. Yang *et al.*, "Totalsegmentator: robust segmentation of 104 anatomic structures in ct images," *Radiology: Artificial Intelligence*, vol. 5, no. 5, p. e230024, 2023.

[29] S. Pai, I. Hadzic, D. Bontempi, K. Bressem, B. H. Kann, A. Fedorov, R. H. Mak, and H. J. Aerts, "Vision foundation models for computed tomography," *arXiv preprint arXiv:2501.09001*, 2025.

[30] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick, "Masked autoencoders are scalable vision learners," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 16 000–16 009.

[31] M. Oquab, T. Darcet, T. Moutakanni, H. V. Vo, M. Szafraniec, V. Khalidov, P. Fernandez, D. HAZIZA, F. Massa, A. El-Nouby, M. Assran, N. Ballas, W. Galuba, R. Howes, P.-Y. Huang, S.-W. Li, I. Misra, M. Rabbat, V. Sharma, G. Synnaeve, H. Xu, H. Jegou, J. Mairal, P. Labatut, A. Joulin, and P. Bojanowski, "DINOv2: Learning robust visual features without supervision," *Transactions on Machine Learning Research*, 2024, featured Certification. [Online]. Available: https://openreview.net/forum?id=a68SUt6zFt

[32] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark *et al.*, "Learning transferable visual models from natural language supervision," in *International conference on machine learning*. PMLR, 2021, pp. 8748–8763.

[33] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *International conference on machine learning*. PMLR, 2020, pp. 1597–1607.

[34] A. Yang, A. Li, B. Yang, B. Zhang, B. Hui, B. Zheng, B. Yu, C. Gao, C. Huang, C. Lv *et al.*, "Qwen3 technical report," *arXiv preprint arXiv:2505.09388*, 2025.

[35] A. Abouelenin, A. Ashfaq, A. Atkinson, H. Awadalla, N. Bach, J. Bao, A. Benhaim, M. Cai, V. Chaudhary, C. Chen *et al.*, "Phi-4-mini technical report: Compact yet powerful multimodal language models via mixture-of-loras," *arXiv preprint arXiv:2503.01743*, 2025.

[36] G. Team, A. Kamath, J. Ferret, S. Pathak, N. Vieillard, R. Merhej, S. Perrin, T. Matejovicova, A. Ramé, M. Rivière *et al.*, "Gemma 3 technical report," *arXiv preprint arXiv:2503.19786*, 2025.

[37] R. Li, J. Li, B. Jian, K. Yuan, and Y. Zhu, "Reevalmed: Rethinking medical report evaluation by aligning metrics with real-world clinical judgment," in *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, 2025, pp. 11 823–11 837.

[38] S. Ostmeier, J. Xu, Z. Chen, M. Varma, L. Blankemeier, C. Bluethgen, A. E. M. Md, M. Moseley, C. Langlotz, A. S. Chaudhari *et al.*, "Green: Generative radiology report evaluation and error notation," in *Findings of the association for computational linguistics: EMNLP 2024*, 2024, pp. 374–390.

[39] A. Yan, J. McAuley, X. Lu, J. Du, E. Y. Chang, A. Gentili, and C.-N. Hsu, "Radbert: adapting transformer-based language models to radiology," *Radiology: Artificial Intelligence*, vol. 4, no. 4, p. e210258, 2022.

[40] R. Stock, M. Langenberg, D. Zimmerer, K. Dvornikovich, J. C. Holzschuh, J. Suprijadi, C. Ulrich, A. Rastogi, K. Schlamp, P. Vollmuth *et al.*, "Evaluating the evaluators: On the reliability of automated label extraction for radiology reports," in *Medical Imaging with Deep Learning-Short Papers*, 2026.

[41] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "Bleu: a method for automatic evaluation of machine translation," in *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, 2002, pp. 311–318.

[42] S. Banerjee and A. 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*, 2005, pp. 65–72.

[43] C.-Y. Lin, "Rouge: A package for automatic evaluation of summaries," in *Text summarization branches out*, 2004, pp. 74–81.

[44] X. Chen, H. Fang, T.-Y. Lin, R. Vedantam, S. Gupta, P. Dollár, and C. L. Zitnick, "Microsoft coco captions: Data collection and evaluation server," *arXiv preprint arXiv:1504.00325*, 2015.

[45] Salaniz, "pycocoevalcap: Microsoft coco caption evaluation," https://github.com/salaniz/pycocoevalcap, 2020.

[46] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, "LoRA: Low-rank adaptation of large language models," in *International Conference on Learning Representations*, 2022.

[47] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," *arXiv preprint arXiv:1711.05101*, 2017.

[48] T. Wald, C. Ulrich, J. Suprijadi, S. Ziegler, M. Nohel, R. Peretzke, G. Kohler, and K. Maier-Hein, "An openmind for 3d medical vision self-supervised learning," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2025, pp. 23 839–23 879.

[49] V. Bumgardner, M. A. Klusty, M. S. Gokmen, and E. W. Damron, "Curriculum-driven 3d ct report generation via language-free visual grafting and zone-constrained compression," *arXiv preprint arXiv:2603.23308*, 2026.

[50] Y. Mao, W. Xu, Y. Qin, and Y. Gao, "Ct-agent: A multimodal-llm agent for 3d ct radiology question answering," *Science China Information Sciences*, vol. 69, no. 5, p. 150107, 2026.

[51] P. Shi, M. Zhang, K. Song, J. Liu, Y. Gu, and X. Zhang, "U-vlm: Hierarchical vision language modeling for report generation," *arXiv preprint arXiv:2603.00479*, 2026.