E-Head: Efficient Head CT Report Generation via Multi-Resolution Series Integration and Two-Stage Parameter-Efficient Fine-Tuning

Satoshi Kondo[1*] and Satoshi Kasai[2]

[1] Muroran Institute of Technology, Hokkaido, Japan
[2] Fujita Health University, Aichi, Japan
* Corresponding author

Abstract. Automated report generation from non-contrast head CT scans is essential for reducing the workload on radiologists, but challenges remain in processing the spatial context of 3D data and performing many-to-many alignment between images and text. In this paper, we propose a highly efficient multimodal model based on the MICCAI 2026 HEADLINE Challenge. The method consists of a pre-trained HLIP visual encoder, a two-layer MLP projector, and a 4-bit quantized Qwen3.5-4B model. It achieves stable spatial alignment while minimizing computational resources with LoRA and a two-stage training strategy. Experiments confirm a BERTScore of 0.8632 on the validation set and an overall score of 0.2403 in the official validation phase.

Keywords: Head CT Report Generation, Vision-Language Models, Parameter-Efficient Fine-Tuning.

1 Introduction

Non-contrast head computed tomography is the imaging modality of first choice for acute neurological disorders, emergency medicine, and trauma diagnosis, and there is an extremely high clinical demand for it. Rapid assessment of cerebral hemorrhage, cerebral infarction, skull fractures, or space-occupying lesions such as tumors is directly linked to critical decisions that determine a patient's prognosis. However, the explosive increase in the number of examinations in clinical settings has placed a massive interpretation burden on radiologists and emergency physicians, raising concerns about delays in report generation and the risk of missed findings due to excessive fatigue [9]. Against this backdrop, there is a growing need to develop an artificial intelligence system capable of automatically generating accurate diagnostic reports from head CT images.

In recent years, systems for automatically generating medical reports, primarily based on 2D images such as chest X-rays, have advanced alongside developments in visual-language models [10]. However, applying these systems to head CT scans presents two major technical challenges.

The first is that, unlike 2D images, head CT scans consist of 3D volumetric data comprising numerous slices. To detect small lesions, it is necessary to robustly model the spatial context that spans across a vast number of slices. The second challenge is that many-to-many correspondence is harder to resolve in 3D than in the 2D or few-view settings of prior work: a single finding is often evidenced by only a variable subset of the tens to hundreds of contiguous slices along the depth axis, with most slices carrying no relevant evidence. This depth-wise correspondence problem has no direct analogue in 2D report generation.

To date, research on report generation in the field of head CT has been limited by a lack of publicly available, large-scale, high-quality benchmarks. The MICCAI HEADLINE Challenge held in 2026 [1] aims to bridge this gap by providing, for the first time, a real-world head CT dataset that has been rigorously anonymized and quality-assessed, along with reports written by specialist physicians. To address the challenges posed by the HEADLINE Challenge, this study proposes a new, computationally efficient multimodal learning model that generates high-precision and clinically valid reports from 3D head CT images. The model combines a pre-trained visual encoder, Hierarchical attention for Language-Image Pre-training (HLIP) [2], with a 4-bit quantized large-scale language model, Qwen3.5-4B [3, 4], and adapts the language model efficiently using LoRA-based parameter-efficient fine-tuning (PEFT) [11]. To stabilize the initial alignment between the image and language spaces, we further introduce a two-stage training strategy that optimizes the projector before jointly fine-tuning it with LoRA. Finally, the processing pipeline handles two CT series per case—bone-reconstructed images (1 mm slices) and soft-tissue-reconstructed images (4 mm slices)—simultaneously, enabling high-performance report generation under limited computational resources.

The main contributions of this study are as follows.

- Proposal of a highly efficient 3D multimodal architecture: By extracting image features from a pre-trained HLIP and connecting a lightweight projector to a quantized LLM (LoRA-adapted), we have established a generation pipeline capable of reflecting the spatial context of 3D volumes while keeping GPU memory consumption low.
- Cross-modal stabilization via a two-stage learning strategy: By including a phase that optimizes only the randomly initialized projector, we stabilize the initial alignment between the frozen image domain and the language domain, thereby achieving both efficient convergence and high generation performance.

2 Related Works

2.1 Medical Image Report Generation

With the rise of deep learning, particularly Transformers and large language models (LLMs), the task of generating diagnostic reports from medical images has been the subject of active research. Most previous studies have focused on chest X-rays, utilizing large-scale public datasets such as MIMIC-CXR [5] and OpenI [6]. While early attempts primarily relied on approaches using CNNs (Convolutional Neural Networks)

as encoders and RNNs (Recurrent Neural Networks) as decoders, recent research has shifted toward adapting pre-trained vision-language models to the medical domain. However, these methods were designed with 2D images or a small number of multi-view images in mind, and their direct application to head CT scans, which involve 3D volume data comprising tens to hundreds of slices, is limited.

2.2 Report Generation for Head CT Scan

Research on language and report generation using head CT scans has rapidly gained attention in recent years. Zhang et al. proposed a weakly guided attention model with hierarchical interaction (WGAM-HI) and attempted to solve the many-to-many alignment problem between multiple images (multi-slices) and multiple sentences [7]. Furthermore, Zheng et al.'s Pathological Clue-driven Representation Learning (PCRL) model improves the consistency between visual and linguistic features by incorporating "pathological clues", derived from segmentation regions and pathological entities, into cross-modal representation learning [8].

2.3 MICCAI 2026 HEADLINE Benchmark

Previous research on head CT report generation has relied on closed, in-house data sets collected independently by individual research groups, or on limited subsets of such data, making it extremely difficult to conduct a fair comparison of model performance. The MICCAI 2026 HEADLINE Challenge is the first standardized benchmark established to address this fragmentation in evaluation [1]. This challenge provides pairs of 3D non-contrast head CT images and structured and unstructured reports written by neuroradiologists. As evaluation metrics, a comprehensive evaluation framework is adopted that combines not only simple string similarity (NLP metrics) but also verification through clinical concept extraction and independent blind reviews by specialists.

The HEADLINE Challenge is a large-scale real-world clinical dataset combining 3D volumes of plain head CT scans with corresponding clinical reports written by radiologists. Each case includes soft tissue and bone reconstruction images that reflect a typical reading environment. The data consists of a wide range of cases in the field of emergency neuroradiology, ranging from normal cases to intracranial hemorrhage, ischemic changes, trauma, fractures, and tumors.

The dataset is divided into a training set of over 16,000 cases, a validation set of 500 cases, and a test set of 500 cases, for a total of over 17,000 cases.

3 Proposed Method

3.1 Overview

Figure 1 shows an overview of our proposed method. In the proposed method, we construct a multimodal model that automatically generates radiological reports using head CT data consisting of two series per case: bone-reconstructed images (1 mm thick

slices) and soft-tissue-reconstructed images (4 mm thick slices). The proposed method consists of HLIP [2], a pre-trained visual encoder for medical images; a projector that transforms image features into the embedding space of a language model; and the large-scale language model Qwen3.5-4B [3, 4]. To reduce computational load and GPU memory consumption, image features generated by HLIP are extracted and cached prior to training, and the language model itself is quantized to 4 bits and adapted using LoRA.

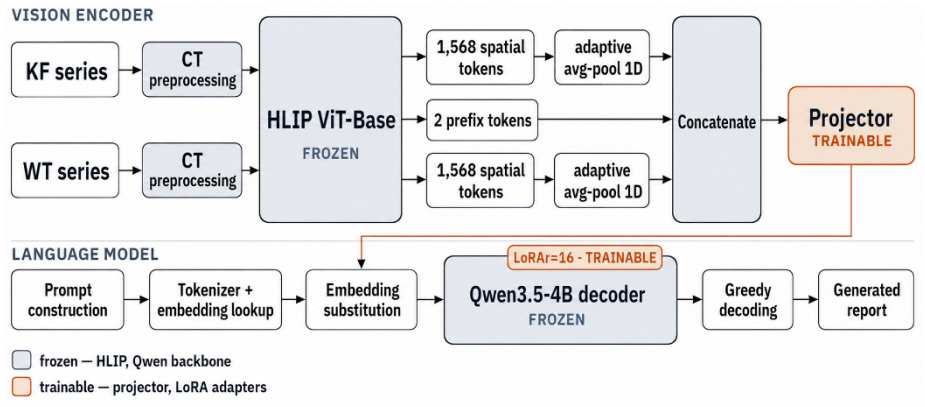

Fig. 1. Overview of our proposed method.

3.2 Image Feature Extraction

Following the official HLIP training pipeline [2], the pixel values of each CT series are clipped to the range [-1150, 350] HU and linearly transformed to [0, 1]. We adopt this range, rather than a bone-specific window, to keep the input distribution consistent with the one the frozen HLIP encoder was pretrained on; since HLIP's weights are not updated during our training, matching its pretraining input distribution takes priority over maximizing osseous contrast in the bone-reconstructed series. We note this as a limitation: values above 350 HU covering most osseous structures are saturated, and a bone-specific window for the KF series is a natural direction for future work.

Next, zero-padding is applied to the short side of each slice to create a square, and bilinear interpolation is performed to resize the in-plane dimensions to 256×256 pixels. The axial direction is standardized to 48 slices using nearest-exact interpolation. Subsequently, the central 224×224 pixels are extracted and normalized using scalar values obtained by averaging the RGB mean and standard deviation across channels from ImageNet. The same processing is applied to both of two series.

For the image encoder, we use a pre-trained HLIP ViT-Base model, whose weights are not updated during training. Each preprocessed 48×224×224 volume is divided into non-overlapping 3D patches of size 6×16×16 (6 slices along the depth axis, 16×16 pixels in-plane), projected to 768 dimensions via a 3D convolutional patch embedding; this yields an 8×14×14 grid, i.e., 1,568 spatial tokens per series. The encoder additionally maintains two prefix tokens per series, a class token and a register token, which a study-level attention stage averages across the two series, so that each case ultimately

carries a single shared pair of prefix tokens rather than one pair per series. The two preprocessed series are thus input jointly to extract, for each case, these two shared prefix tokens together with 1,568 spatial tokens per series, all of dimension 768. We apply one-dimensional adaptive average pooling independently to each series' spatial tokens, compressing the 1,568 tokens per series to 128. Consequently, the image representation for a single case consists of 2+2×128=258 tokens, and the feature matrix has shape 258×768.

3.3 Vision Language Model

Each image token from HLIP is transformed into Qwen embedding dimensions using a projector consisting of a two-layer MLP. The projector consists of a linear layer mapping 768 to 2,560 dimensions, a GELU nonlinearity, and a second linear layer mapping 2,560 to 2,560 dimensions.

Because the tokenizer and chat template operate on discrete text tokens rather than continuous vectors, we cannot insert image features directly into the prompt string. Instead, we reserve one placeholder token, <|image_pad|>, per image feature vector (258 per case) at the exact positions where the visual content should appear. After tokenization, we look up the embedding table as usual, locate the placeholder positions by their token id, and overwrite (not use) their embeddings with the projector's output; the placeholder's original embedding is never used. This placeholder-then-substitute step, common in vision-language models with image-in-text prompting, lets the image tokens occupy well-defined positions in the sequence (with correct attention masks and position indices) while carrying the projected visual content instead of the tokenizer's original embedding. The same placeholder id is also used to mask these positions out of the loss during training

We use Qwen3.5-4B [3, 4] as the language model. In the system prompt, we instruct the model to write a concise and conventional head CT report as a neuroradiologist. We also fix the output format to two sections, “**Findings:**” and “**Assessment:**”, and instruct the model to avoid speculations beyond the image findings and unnecessarily detailed descriptions. The goal is to maintain readability as a clinical report while generating a concise writing style similar to that of reference reports.

During training, we concatenate the correct report as an assistant response to the prompt and minimize the autoregressive cross-entropy loss for the ground truth report section.

4 Experiments and Results

4.1 Dataset

To prevent cases from the same patient from being included in both the training set and the validation set, we split the data by patient ID into groups. We split the data once, setting the training ratio to 0.8 and the validation ratio to 0.2, and conducted subsequent

experiments using the same split. The data used for training consisted of 9,106 cases in the training set and 2,362 cases in the validation set.

4.2 Experimental Conditions

We used PyTorch Lightning library for training and ran Distributed Data Parallel (DDP) on three GPUs. Since the minibatch size for each GPU was set to 1 and the gradient accumulation was set to 4 steps, the effective batch size used for a single optimizer update was 12 cases. Training was performed for 12 epochs using BF16 mixed precision and a maximum input sequence length of 1,024 tokens.

The Qwen backbone was quantized to 4 bits using the NF4 data type with double (nested) quantization, implemented via the bitsandbytes library. Quantized weights were dequantized to BF16 for computing, matching the BF16 mixed-precision training setting. The weights of the fully connected layers of Qwen were fixed, and LoRA was inserted into the q_proj, k_proj, v_proj, and o_proj weights of the full-attention layer, as well as the in_proj_qkv, in_proj_z, and out_proj weights of the linear-attention layer. We set the LoRA rank to 16 (with α=32, i.e., a scaling factor α/r=2) based on common practice for LoRA fine-tuning of billion-parameter LLMs, without performing a dedicated rank ablation due to the computational cost of full training runs. The dropout rate was set to 0.05; the bias was not trained. Only the LoRA parameters and projectors were trained, with approximately 12.4 M trainable parameters for LoRA and approximately 8.5 M for the projectors, for a total of approximately 20.9 M.

AdamW was used for optimization with $\beta_1 = 0.9$, $\beta_2 = 0.999$, and weight decay set to 1×10^{-4}. To stably align the projectors, which were randomly initialized at the start of training, with the image and language spaces, we performed the following two-stage training.

1. Stage 1 (epochs 1-2): The LoRA learning rate was set to 0, and only the projector was updated with a learning rate of 5×10^{-5}.
2. Stage 2 (epochs 3-12): LoRA and the projector were trained simultaneously, with learning rates of 5×10^{-6} and 1×10^{-5}, respectively.

In the two-stage training, we did not use a learning rate scheduler and kept the learning rate constant within each stage. We kept the LoRA parameters in a trainable state even after DDP initialization, and suppressed updates in Stage 1 by setting the learning rate to 0.

All experiments were run on a single machine with three NVIDIA GeForce RTX 4090 GPUs (24 GB each) under PyTorch Lightning DDP. Because the 4B-parameter backbone is loaded in 4-bit and adapted only through a rank-16 LoRA adapter and a lightweight projector, peak GPU memory usage reached approximately 25.7 GB, effectively saturating each 24 GB card without requiring model-parallel sharding across GPUs. The full 12-epoch training run took approximately 8.4 hours in total. Epochs with loss-only validation completed in 15-17 minutes each, whereas the two epochs (5 and 10) that additionally performed generation-based validation over the full 2,362-case validation set took approximately 2 hours 50 minutes each, since greedy autoregressive decoding used a per-GPU batch size of 1. This corresponds to roughly 11.6-

11.8 seconds of wall-clock inference time per case on a single RTX 4090, the same decoding configuration used to produce the results in Sec. 4.3.

4.3 Validation and Generation Conditions

The validation loss was calculated at the end of each epoch. However, since report generation is time-consuming, evaluations based on generation were conducted every 5 epochs, specifically, at the end of epochs 5 and 10. Generation was performed using greedy decoding without sampling, with a maximum output length of 192 tokens, an repetition penalty of 1.05, and a no-repeat n-gram setting of 4.

The evaluation metrics used were BLEU-4, ROUGE-L, METEOR, and BERTScore. Model selection was based on the validation BERTScore.

At epoch 10, when the generation evaluation was performed, BLEU-4 was 0.0502, ROUGE-L was 0.2356, METEOR was 0.1907, and BERTScore was 0.8632.

Furthermore, the evaluation results from the official validation phase on the Grand Challenge website were: Score = 0.2403 (SAuto = 0.2403, BERTScore = 0.2675, BLEU-4 = 0.1015, ROUGE-L = 0.2813, METEOR = 0.1478).

Since our checkpointing rule saves the LoRA adapter and projector whenever validation BERTScore F1 improves, and epoch 10 achieved the higher of the two generation-evaluated BERTScore values (0.8632 vs. 0.8631 at epoch 5), the epoch-10 checkpoint was retained as the final model and used to generate the official submission reported in the following results.

Because generation-based evaluation requires autoregressive decoding over the full validation set and is substantially more expensive than computing validation loss, we restricted it to every 5 epochs (epochs 5 and 10 of the 12 total epochs); validation loss, in contrast, was computed every epoch. Since 12 is not a multiple of 5, the final two epochs (11 and 12) were never evaluated with generation-based metrics, and epoch 10, being both the last generation-evaluated epoch and the one with the higher BERTScore F1 of the two candidates, was retained as the checkpoint used for the final model. We note this as a limitation: it remains possible that epochs 11–12, whose validation loss continued to decrease slightly, could have yielded a marginally different generation quality that our evaluation schedule did not capture.

4.4 Discussion

The high internal-validation BERTScore (0.8632) together with comparatively low BLEU-4/ROUGE-L reflects a systematic pattern rather than noise. The model reliably reproduces the two-section template and the phrasing of common negative findings (e.g., "no evidence of hemorrhage," "preserved gray-white differentiation"), which drives strong BERTScore agreement even for imperfect reports. However, case-specific findings—particularly rare or device-related findings such as shunt hardware, and the exact modality/timing of prior comparison studies—are frequently omitted or replaced with generic phrasing, which depresses n-gram-based metrics such as BLEU-4 and ROUGE-L that are sensitive to exact rare-entity matches. Table 1 illustrates this pattern with a representative validation case generated by the final (epoch-10)

checkpoint. We interpret this as evidence that the model has learned the stylistic and structural conventions of the target reports more reliably than fine-grained, low-frequency clinical content—an expected limitation of a frozen visual encoder combined with a heavily parameter-efficient (LoRA rank 16) adaptation of a 4-bit quantized backbone. We also note, without a confirmed explanation, a large gap between our internal-validation BERTScore (0.8632) and the BERTScore reported by the official validation phase (0.2675); we hypothesize this may stem from differences in the BERTScore implementation or reference-embedding backbone used by the official scorer, or from a distribution shift between our internal validation split and the official hidden validation set, and we leave a definitive explanation to future investigation.

Table 1. Representative generated vs. reference reports from the final (epoch-10) checkpoint, illustrating template/style fidelity alongside omission of case-specific findings.

	Text
Reference	Findings: Comparison with in-house native CCT from [DATE]. Status post VP-shunt implantation on the right frontal side with an intercurrent valve exchange. The shunt catheter descends retroauricularly to the right. The catheter tip is appropriately positioned in the anterior horn of the right lateral ventricle. Compared to the prior CT, the ventricular width is slightly reduced, with a partially only slit-like delineation. No hygroma detected. Regular gray-white differentiation throughout, without infarct demarcation. No hemorrhage-typical hyperdensities. Adequately ventilated paranasal sinuses and mastoid cell systems bilaterally, as visualized. Assessment: Slightly reduced ventricular width. With a partially slit-like delineated ventricular system, over-drainage is possible; however, this is a young patient and no hygroma is present.
Generated	Findings: For comparison, the CT examination from [DATE] is used. Unchanged width of the inner and outer CSF spaces. Midline interhemispheric fissure. No evidence of hemorrhagic hyperdensities. Preserved cortico-medullary differentiation. Known defect area in the left basal ganglia. Normal ventilation of the paranasal sinuses and mastoid cells. Assessment: No evidence of intracranial hemorrhage or infarct demarcation.

5 Conclusions

In this paper, we report on the model architecture and implementation results of a highly efficient multimodal language model that automatically generates structured diagnostic reports from 3D head CT volume images for the MICCAI 2026 HEADLINE Challenge benchmark.

References

1. https://headline.ccibonn.ai
2. Zhao, C., Lyu, Y., Chowdury, A., Harake, E., Kondepudi, A., Rao, A., Hou, X., Lee, H., Hollon, T.: Towards Scalable Language-Image Pre-training for 3D Medical Imaging. arXiv:2505.21862 (2026)
3. QwenTeam: Qwen3.5: Towards Native Multimodal Agents. https://qwen.ai/blog?id=qwen3.5 (2026)
4. https://huggingface.co/Qwen/Qwen3.5-4B
5. Johnson, A.E.W., Pollard, T.J., Berkowitz, S.J., Greenbaum, N.R., Lungren, M.P., Deng, C.Y., Peng, R.G., Horng, S., Mark, R.G.: MIMIC-CXR, a de-identified publicly available database of chest radiographs with free-text radiology reports. Scientific Data 6, 317 (2019)
6. Demner-Fushman, D., Kohli, M.D., Rosenman, M.B., Shooshan, S.E., Rodriguez, L., Antani, S., Thoma, G.R., McDonald, C.J.: Preparing a collection of radiology examinations for distribution and retrieval. Journal of the American Medical Informatics Association 23(2), 304–310 (2016)
7. Zhang, X., Yang, S., Shi, Y., Ji, J., Liu, Y., Wang, Z., Xu, H.: Weakly guided attention model with hierarchical interaction for brain CT report generation. Computers in Biology and Medicine 167, 107650 (2023)
8. Zheng, C., Ji, J., Shi, Y., Zhang, X., Qu, L.: See detail say clear: Towards brain CT report generation via pathological clue-driven representation learning. In: Findings of the Association for Computational Linguistics: EMNLP 2024, pp. 1650–1663. Association for Computational Linguistics (2024)
9. Kasalak, Ö., Alnahwi, H., Toxopeus, R., Pennings, J. P., Yakar, D., Kwee, T. C.: Work overload and diagnostic errors in radiology. European Journal of Radiology, 167, 111032 (2023)
10. Hartsock, I., Rasool, G.: Vision-language models for medical report generation and visual question answering: a review. Frontiers in Artificial Intelligence, 7, 1430984 (2024)
11. Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., Chen, W.: LoRA: Low-Rank Adaptation of Large Language Models. arXiv:2106.09685 (2021)