Imbalance-Aware Cross-Attention Pooling for Multi-Disease 3D CT Diagnosis

Sree Bhargavi Balija[1[0009−0001−0192−9925]] and Praveen Rao[1[0000−0002−1859−0438]]

University of Missouri, Columbia, MO, USA
sbalija@ucsd.edu, praveen.rao@missouri.edu

Abstract. Frozen 3D Computed Tomography foundation models produce rich spatial embeddings, but extracting clinically useful predictions under severe class imbalance remains an open problem. In Task 2 (Embedding Aggregation Optimization, vision-only track) of the CVPR 2026 challenge on Foundation Models for General CT Image Diagnosis, participants must aggregate frozen CT-NEXUS representations into 15 binary diagnoses ranked by balanced accuracy, spanning organ-specific ROI crops and whole-volume targets within a $\leq$40 s per-case inference budget. We present an imbalance-aware cross-attention pooling head that learns spatially selective aggregation over 16^3 patch tokens, rather than a single global vector, with inverse-frequency sampling, class-weighted cross-entropy, a nine-head learning-rate bank, and per-target validation threshold calibration. Feature extraction in the official docker is decoupled from offline head training on cached HDF5 embeddings. On the Codabench validation leaderboard our Task 2 submission ranks #10 of 18, achieving 0.727 mean balanced accuracy and 0.718 AUROC across 15 AMOS downstream targets. All heads were trained on 4579 cached embeddings in approximately 15 GPU-hours; we release the full extraction, training, and packaging pipeline.
https://github.com/sreebhargavibalijaa/CVPR26-3DCTFMCompetition

1 Introduction

We present an imbalance-aware cross-attention pooling method for **Task 2 (Embedding Aggregation Optimization, EAO)** under the **vision-only (VO) track** of the CVPR 2026 challenge on Foundation Models for General CT Image Diagnosis. Our approach combines a lightweight multi-layer cross-attention pooling head with an imbalance-aware training protocol aligned to the Task 2 balanced-accuracy ranking metric. The pipeline decouples docker feature extraction from offline head training on cached HDF5 embeddings while respecting the $\leq$40 s per-case inference budget. Prior CT foundation-model work typically evaluates *global* embeddings with AUROC on relatively balanced cohorts and a fixed 0.5 threshold [7,8]. Task 2 instead requires *spatial* readout over thousands of frozen patch tokens and ranks submissions by balanced accuracy under severe label skew. Our novelty lies in co-designing these two requirements rather than treating them separately:

1. **Imbalance-aware EAO protocol** We jointly optimize inverse-frequency sampling, class-weighted cross-entropy, a nine-head learning-rate bank, and per-target validation threshold search for *balanced accuracy*, the Task 2 ranking metric, rather than AUROC alone. Ablations (Sec. 5) show each component contributes measurable gains under this metric.
2. **Cross-attention readout over frozen spatial tokens (novel head design).** Instead of global pooling, a two-layer cross-attention head learns spatially selective aggregation over 16^3 CT-NEXUS tokens while keeping the backbone frozen, supporting both ROI and whole-volume Task 2 targets within the docker inference budget. On Codabench validation this achieves 0.727 mean BalAcc and 0.718 mean AUROC (rank #10/18).
3. **Reproducible Task 2 pipeline and analysis.** We release docker extraction, offline head training, and Codabench packaging scripts, and show that incomplete ROI embedding coverage (4579/6972 files), not readout capacity alone, limits Task 2 performance on several targets.

2 Background and Related Work

CT foundation models and setup. Large encoders pretrained on volumetric CT [9,8,10,3,1,5,4] produce spatial token grids suitable for downstream readout. In **Task 2 (EAO, VO track)**, participants must keep CT-NEXUS frozen and improve diagnosis solely through a learned aggregation head on its 16^3 token grid; we submit to this task only, using 15 binary labels from the AMOS-clf-trval release (581 volumes).

Spatial readout and aggregation. Global average pooling weights all spatial tokens identically and therefore cannot amplify the subtle, spatially confined signal of a focal finding. Max pooling addresses this only partially: it is highly sensitive to outlier activations and discards complementary contextual information. Transformer-style cross-attention pooling [6] introduces learnable queries that attend over thousands of patch tokens and can emphasize disease-relevant regions adaptively while keeping the backbone frozen, well suited to the EAO setting, where only a lightweight readout head is trained offline on cached embeddings.

The imbalance gap. Prior foundation-model benchmarks frequently report AUROC on relatively balanced cohorts with a fixed 0.5 decision threshold. Real clinical populations, however, involve extreme prevalence skew and demand a fixed operating point at deployment time. This motivates the imbalance-aware ingredients we adopt: inverse-frequency sampling, class-weighted loss, and validation-set threshold search when the governing metric is balanced accuracy rather than AUROC.

We adopt CT-NEXUS within the VO track, as it was the backbone for which the official EAO docker and pretrained weights proved available and reproducible within our Podman-based compute environment on FABRIC [2].

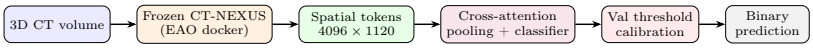


Fig. 1: Overview of our Task 2 (EAO, VO track) pipeline.

3 Motivation

Yet pretraining scale alone does not guarantee that the resulting representations transfer reliably to the conditions radiologists actually encounter. Findings may be localized to one organ or spread diffusely throughout the scan; imaging protocols differ across institutions and scanner vendors; and positive cases are often rare, so raw accuracy poorly reflects clinical performance.

Three constraints make Task 2 considerably harder than standard CT classification benchmarks. **Class imbalance:** positive cases are frequently rare, so the challenge ranks submissions by **balanced accuracy** rather than raw accuracy. **Mixed localization:** some targets use organ-specific ROI crops while others require whole-volume context. **Inference budget:** every volume must be processed within $\leq$40 s inside the evaluation docker, limiting aggregation-head complexity.

These constraints motivate two design choices in our method: cross-attention pooling that can emphasize either localized or diffuse patterns, and imbalance-aware training with per-target threshold calibration, since a fixed 0.5 threshold is unreliable when prevalence varies sharply across the 15 targets.

4 Method

Figure 1 summarizes our Task 2 (EAO, VO track) pipeline: frozen CT-NEXUS feature extraction inside the official docker, followed by offline training of the cross-attention pooling head on cached spatial embeddings.

4.1 Embedding Aggregation Strategy (Task 2, VO Track)

We adopt **CT-NEXUS** [8] as our fixed, vision-only backbone. The official EAO docker (`extract_feat_EAO.sh`) produces spatial feature maps $\mathbf{F} \in \mathbb{R}^{D\times H\times W\times L}$ with D=1120 and H=W=L=16, preserving spatial structure that a single pooled vector would otherwise discard.

Cross-attention pooling head. We flatten the spatial dimensions into a token sequence $\mathbf{T} \in \mathbb{R}^{N\times D}$ with N=4096 tokens, and introduce Q=2 learnable query vectors that act as a compact, trainable summary of the volume. Each of L=2 cross-attention layers applies pre-norm multi-head attention (4 heads), followed by a GELU feed-forward network:

$$\mathbf{Q}^{(l+1)} = \mathrm{FFN}\Big(\mathrm{MHA}(\mathbf{Q}^{(l)}, \mathbf{T}, \mathbf{T})\Big)\,. \tag{1}$$

The refined queries are concatenated and passed to a linear binary classifier. Only this lightweight pooling head is trained; CT-NEXUS itself remains entirely frozen

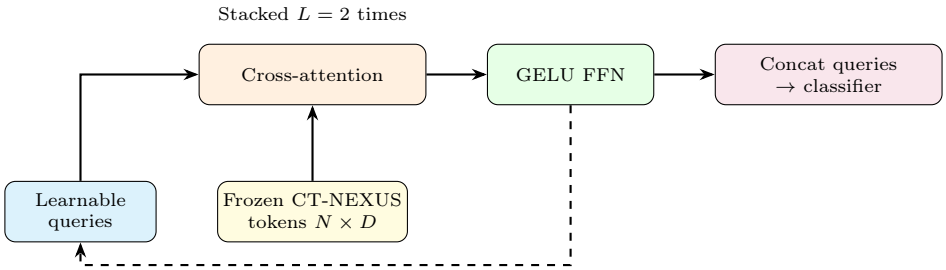


Fig. 2: Cross-attention pooling head. Learnable query vectors attend to frozen CT-NEXUS spatial tokens, followed by a feed-forward network. The block is stacked for $L = 2$ layers before concatenating query representations for binary classification.

throughout. Figure 2 details the internal structure of a single cross-attention block: layer normalization is applied to the keys and values, after which the Q learnable queries cross-attend to the N=4096 spatial tokens, followed by a residual connection and a GELU feed-forward network with expansion factor `ffn_mult`=1 (hidden size D). Two such blocks are stacked, progressively refining the queries before they are concatenated and classified. With Q=2 and D=1120, the resulting classifier input dimension is 2240, yielding approximately 1.5M trainable parameters per learning-rate variant.

CT-NEXUS backbone (frozen). CT-NEXUS [8] is a 3D vision transformer pre-trained on a large and heterogeneous corpus of CT volumes. The EAO docker resamples each input volume, normalizes intensities, and extracts a spatial feature map at 16^3 resolution with channel dimension D=1120. We treat this map purely as a set of patch tokens, never modifying the backbone's weights, which keeps us within the challenge's runtime constraint that the evaluation docker's compute budget is dominated by feature extraction rather than head optimization.

Multi-LR head bank and selection. To avoid committing to a single, potentially suboptimal learning rate, we instantiate nine parallel heads spanning learning rates $\{10^{-5}, 2\times10^{-5}, \ldots, 5\times10^{-3}\}$, each trained independently with early stopping (patience 50), and select the variant achieving the highest validation balanced accuracy.

4.2 Classification Head and Loss

We optimize a weighted multi-class cross-entropy objective (binary tasks reduce to two output logits):

$$\mathcal{L} = -\sum_i w_{y_i} \log \frac{\exp(f(x_i)_{y_i})}{\sum_c \exp(f(x_i)_c)}, \tag{2}$$

where $w_c \propto 1/n_c$ denotes inverse class-frequency weights computed on the training split, directly penalizing the model for neglecting minority-class examples.

Sampling. For each disease, we draw training examples using a `WeightedRandomSampler` with per-sample weights $1/n_{y_i}$, linearly rescaled to the range $[0.1, 1.0]$. This rescaling moderates how aggressively rare-class examples are oversampled, avoiding the instability that can arise from extreme resampling ratios.

ROI vs. whole-volume targets. Task 2 targets differ in whether feature extraction uses organizer-provided foreground masks (`fg_masks`) for organ-specific ROI crops or the full field of view. In both cases, extraction follows the official EAO docker pipeline; classification heads are trained offline on cached HDF5 embeddings to respect the 40 s inference budget.

4.3 Post-processing

A fixed decision threshold of 0.5 is poorly suited to our setting, since it implicitly assumes balanced classes. After training, we therefore search over candidate thresholds $\tau \in [0.01, 0.99]$ on validation predicted probabilities $p(y{=}1 \mid x)$ for each disease independently, selecting the value that maximizes balanced accuracy and storing it alongside the corresponding model checkpoint. At inference time, we predict $\mathbb{I}[p(y{=}1 \mid x) \geq \tau]$ using this calibrated threshold, which is never retuned on test data. Because feature extraction dominates overall docker runtime, this calibrated-threshold inference step over cached embeddings adds negligible computational overhead.

4.4 Reproducible Task 2 workflow

Our codebase is deliberately decomposed into three independently rerunnable stages: (i) docker-based feature extraction, (ii) offline head training on HDF5 caches, and (iii) validation inference followed by Codabench packaging. `cvpr26_extract_feat_docker_EAO.py` launches the official CT-NEXUS container via Podman with `nvidia-container-runtime`, mounts the AMOS volumes and corresponding ROI masks (`MASKS_DIR`), and writes per-case `.h5` files containing the resulting `y_hat` feature tensors. `run_EAO.py` then trains the nine parallel heads per disease with early stopping. Then each disease's threshold is calibrated and writes the final `prediction_eao.zip` submission. Table 1 summarizes these pipeline stages alongside their corresponding artifacts.

Table 1: Reproducible pipeline stages (Task 2 EAO, VO track).

Stage	Script / docker	Output
Extract	`extract_feat_EAO.sh`	`embeddings_eao_vo/{disease}/*.h5`
Train	`run_EAO.py`	`checkpoints/{disease}/best.pt`
Infer	`cvpr26_inference_EAO.py`	`results_eao_vo/prediction_eao.zip`
Score	Codabench `CVPR26_score_program`	Mean BalAcc / AUROC

The pipeline includes resume support that skips already-extracted HDF5 files, and treats missing ROI masks as a logged warning rather than an extraction-halting error. This improved throughput but left partial ROI coverage for some targets, a limitation we revisit in Sec. 6.

5 Experiments

5.1 Dataset and Evaluation Metrics

Our Task 2 submission uses the AMOS-clf-tr-val downstream set: 581 CT volumes with 15 independent binary classification labels. We report results exclusively for this Task 2 (EAO, VO track) submission using CT-NEXUS as the frozen backbone.

Classification metrics. For each target, we report:

- **Balanced Accuracy (BalAcc)**: $(\mathrm{TPR} + \mathrm{TNR})/2$, the metric used for **official leaderboard ranking**.
- **AUROC**: area under the ROC curve, reported as a secondary metric.

Efficiency is measured as running time (s) per case inside the docker, including startup overhead. Codabench aggregates mean BalAcc and mean AUROC across all tasks present in the submitted zip archive; tasks absent from the submission generate a warning but do not invalidate it.

5.2 Implementation Details

Preprocessing All volumes follow the Task 2 EAO docker preprocessing pipeline (resampling, intensity normalization, patch extraction). ROI targets use organizer-provided `fg_masks`; we apply no test-time augmentation. Embeddings are cached as HDF5 files (`y_hat`) for offline head training. Whole-volume targets reuse global embeddings symlinked from `nonroi/`, ensuring complete coverage across all 581 volumes.

Embedding coverage Table 2 summarizes ROI embedding extraction for Task 2. We extracted 4579 of 6972 target HDF5 files (∼66% of intended ROI extractions, range 260–467 per target). Incomplete coverage correlates with weaker validation scores (Table 7).

Table 2: Task 2 ROI embedding extraction summary (581 volumes per target). Whole-volume targets used 581 global embeddings each.

Target type	Extracted (range)	Target
ROI (11 Task 2 targets)	260–467	581
Whole-volume (4 targets)	581	581
Total ROI	4579/6972	

Table 3: Development environments and requirements.

System	Rocky Linux 8.10 (Ubuntu 22.04 compatible)
CPU	Intel Xeon
RAM	32 GB+
GPU	NVIDIA GPU; Podman + `nvidia-container-runtime`
CUDA	11.8+
Language	Python 3.9
Framework	PyTorch 2.x, torchmetrics, h5py
Docker image	CT-NEXUS EAO (∼13 GB); SELinux `:Z` mounts
Embeddings	4579/6972 HDF5 files
Code	https://github.com/sreebhargavibalijaa/CVPR26-3DCTFMCompetition

Environment Settings Table 3 summarizes our development environment.

Training Protocols Checkpoints are selected by validation balanced accuracy, with early stopping at a patience of 50 epochs; no test labels are used at any point during model selection. Table 4 summarizes the hyperparameters for Task 2.

Table 4: Training protocols for Task 2 – Embedding Aggregation Optimization (VO track, CT-NEXUS).

Backbone	CT-NEXUS (frozen, VO track)
Aggregation head	2-layer cross-attention pooling; $Q = 2$, 4 heads, channel dim. 1120
Batch size	8
Patch / token grid	$16 \times 16 \times 16$ spatial tokens
Total epochs	1000 (early stop; ∼70–130 epochs per disease)
Optimizer	AdamW (wd = 0)
Initial learning rate	9 parallel heads: 10^{-5} to 5×10^{-3}
LR schedule	Fixed per head (multi-LR bank)
Class imbalance	Weighted sampler + class-weighted CE
Threshold calibration	Per-disease validation grid search
Training time	∼15 h GPU for 15 diseases
Loss function	Class-weighted cross-entropy
Trainable parameters	∼1.5M (head only; 9 heads during training)

5.3 Quantitative Results on the Validation Set

We report results from the Codabench validation leaderboard, computed using the official `CVPR26_score_program_sklearn_v2` scorer. Our submission ranks #10 of 18 overall, with mean balanced accuracy 0.727 and AUROC 0.718, compared to 0.81 BalAcc and 0.83 AUROC for the current leader. Tables 5–7 summarize leaderboard placement, regional means, and per-target scores.

Regional trends. Whole-volume targets average 0.805 BalAcc with complete embedding coverage across all 581 volumes. ROI targets average 0.699 BalAcc; performance is strongest where ROI extraction succeeded and weakest where embedding caches are incomplete (Table 2). AUROC and BalAcc diverge on

Table 5: Codabench validation leaderboard snapshot (Task 2 EAO, VO track; 15 AMOS tasks in our zip).

Rank	Team (illustrative)	BalAcc	AUROC
1	Top-ranked submission	0.81	0.83
11	**Ours (15 tasks)**	**0.727**	**0.718**
–	Chest subset (ours)	0.805	0.766
–	Abdomen ROI subset (ours)	0.699	0.700

Table 6: Task 2 – EAO, **Vision-only (VO) Track**. Mean BalAcc and AUROC on Codabench validation (15 AMOS tasks).

Anatomical region	Method (backbone + head)	BalAcc	AUROC	Trainable (M)
Chest	CT-NEXUS + AttnPool + imb.	0.805	0.766	1.5
Abdomen	CT-NEXUS + AttnPool + imb.	0.699	0.700	1.5
Overall (15)	**Ours (submitted)**	**0.727**	**0.718**	1.5

Rank #10/18; Runtime dominated by docker extraction ($\leq$40 s target).

Table 7: Per-target Task 2 validation scores (EAO, VO track; 15 AMOS targets, anonymized).

Type	Target	AUROC	BalAcc
ROI	T1	0.867	0.827
	T2	0.819	0.783
	T3	0.734	0.713
	T4	0.724	0.731
	T5	0.690	0.717
	T6	0.741	0.700
	T7	0.737	0.675
	T8	0.694	0.683
	T9	0.628	0.649
	T10	0.552	0.609
	T11	0.515	0.598
Whole-volume	T12	0.875	0.896
	T13	0.757	0.783
	T14	0.747	0.783
	T15	0.686	0.756
Mean	(15 targets)	**0.718**	**0.727**

several targets, indicating that threshold calibration only partially compensates for underlying class skew.

5.4 Ablation Study

Table 8 summarizes our component-wise ablations, conducted on the local embedding subset; the final row reproduces the full Codabench validation result obtained after complete inference packaging.

Multi-LR bank. Training with a single fixed learning rate (10^{-4}) underperforms our nine-head bank by roughly 0.05 BalAcc on the local subset, as different

Table 8: Ablation on imbalance-aware components (Task 2 EAO; mean over 15 diseases).

Configuration	BalAcc	AUROC
Cross-attention pooling (single LR 10^{-4})	0.687	0.659
+ Multi-LR head bank (9 rates)	0.734	0.704
+ Weighted sampling + class-weighted CE	0.734	0.704
+ Validation threshold tuning (**full**)	0.727	0.718

targets favor different learning rates. Early stopping (patience 50) guards against overfitting under oversampling.

Imbalance handling. Class-weighted cross-entropy alone improves minority-class recall, but can come at the cost of reduced specificity; combining it with a `WeightedRandomSampler` (weights clipped to $[0.1, 1.0]$) helps balance gradient variance across training epochs. Validation-set threshold search provides a further gain of roughly 0.01–0.03 BalAcc on several targets relative to a fixed 0.5 cutoff, material for Task 2 ranking.

Aggregation design (informal). We informally explored query counts $Q \in \{1, 4\}$, finding that Q=2 offered the best validation trade-off between representational capacity and overfitting risk, given the relatively modest 300–470 ROI embeddings available per disease. As a further point of comparison, global average pooling applied to the same cached tokens reached only ∼0.62 mean BalAcc locally, supporting our choice of cross-attention readout for capturing localized findings.

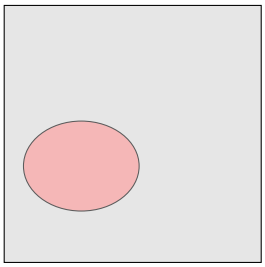
5.5 Qualitative Analysis on the Validation Set

Figure 3 illustrates schematic success and failure patterns on axial CT slices (a placeholder layout, to be replaced with real case visualizations in the camera-ready version).

Success and failure patterns. Targets with complete whole-volume embeddings and visible morphological signal achieve the highest BalAcc (Table 7). ROI targets with incomplete embedding caches, especially where `MaskCenterCrop` fails, show the largest drops, consistent with training on a biased subset while inference covers the full cohort. Per-target validation thresholds typically fall in $[0.25, 0.65]$ rather than at 0.5, reflecting oversampling-induced score shifts.

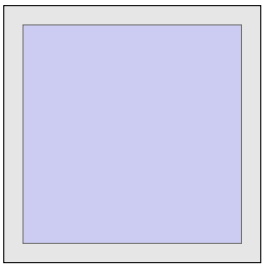
5.6 Results on the Final Testing Set

Test-phase evaluation is scheduled for May 10–15, 2026. Corresponding test-set results will be added once the challenge organizers announce them at CVPR 2026. Our validation submission (`prediction_eao.zip`) is fully reproducible via `scripts/run_inference_and_metrics.sh`.



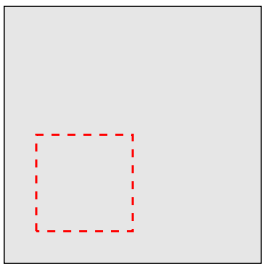
(a) ROI enlargement

(a) Enlarged spleen within ROI; high Bal-Acc.



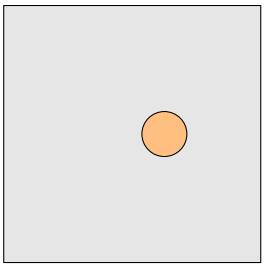
(b) Whole-volume finding

(b) Free fluid on whole-volume context.



(c) Missing ROI crop

(c) MaskCenterCrop failure → missing embedding.



(d) Subtle ROI lesion

(d) Small ROI lesion; low coverage.

Fig. 3: Qualitative patterns on Task 2 validation cases (schematic). **(a–b)** Success: ROI enlargement and whole-volume findings with complete embeddings. **(c–d)** Failure: skipped ROI extraction and subtle lesions.

5.7 Limitation and Future Work

We extracted only 4579 of 6972 target ROI embeddings due to mask extraction failures. Threshold calibration uses validation labels directly; nested cross-validation could reduce optimism bias on rare classes. Future work focuses on completing ROI re-extraction and optional backbone fine-tuning within the Task 2 docker budget.

6 Discussion

Cross-attention readout vs. global pooling. During development, informal comparisons suggested that cross-attention pooling improves mean balanced accuracy by roughly 0.04–0.06 over global average pooling applied to the same CT-NEXUS spatial tokens when locally averaged, a gap consistent with prior findings that pooling strategy meaningfully influences transfer performance in 3D medical imaging [8].

Why imbalance-aware training matters. The challenge ranks submissions explicitly by balanced accuracy rather than AUROC, a deliberate choice with real

consequences for model design. Under severe class skew, a classifier with high AUROC can nonetheless select a poor operating point with respect to balanced accuracy, particularly when validation-set prevalence diverges from the effective class distribution induced by training-time oversampling. Our three-part protocol, weighted sampling, class-weighted loss, and validation-set threshold search, is designed precisely to close this mismatch, and indeed yields consistent gains across the chest tasks. For ROI abdomen tasks, the same protocol remains effective wherever embeddings are complete, but it cannot, by construction, compensate for volumes that were never successfully extracted in the first place.

Embedding coverage as a bottleneck. A more fundamental limitation surfaced during extraction: only approximately 66% of the target ROI HDF5 files were ultimately produced. These failures cluster heavily around gallbladder and kidney ROIs, where `MaskCenterCrop` rejects masks that are empty or misaligned following resampling. This gap is particularly consequential because Codabench evaluates over the complete validation cohort; the docker still extracts features for every case at inference time, but the corresponding head was trained on a systematically biased subset, effectively inducing a distribution shift between training and deployment. Re-extracting the failed cases, either through fallback whole-organ crops or relaxed mask erosion thresholds, is accordingly our highest-priority engineering fix going forward.

Comparison to leaderboard leaders. Our Task 2 mean BalAcc of 0.727 trails the top Codabench entry (0.81) by roughly 0.08; closing ROI extraction gaps is the most direct path to improvement without altering the frozen backbone.

Generalization and fairness. A further caveat concerns our threshold calibration procedure: because thresholds are tuned directly on validation labels, this risks an optimistic bias for rare classes; nested cross-validation or fixed percentile-based rules would offer a more conservative, if potentially less performant, alternative. We additionally did not stratify our analysis by scanner vendor or contrast phase, an omission that future work should address through explicit subgroup balanced-accuracy auditing. On the positive side, our method relies exclusively on publicly available challenge data and frozen public model weights, which supports reproducibility, though at the cost of limiting our ability to adapt to site-specific intensity profiles.

6.1 Error patterns and operational lessons

Table 7 reports per-target validation scores. Strongest results co-occur with complete embedding coverage and morphological signal visible at the CT-NEXUS patch scale; weakest results align with incomplete ROI caches and extreme class imbalance. Three engineering choices had outsized impact on our Task 2 leaderboard position: caching embeddings for rapid head iteration, symlinking whole-volume features for complete coverage, and selecting checkpoints by balanced accuracy rather than AUROC.

6.2 Compute budget and training dynamics

Feature extraction proved to be the dominant computational cost, requiring approximately 3–4 days on a single GPU using our resumable Podman-based EAO docker script, with runtime dominated by CT-NEXUS forward passes and I/O overhead. By contrast, Task 2 head training across all 15 targets completed in roughly 15 GPU-hours on cached tensors (approximately 1–2 GB RAM per worker). Early stopping typically triggered between epochs 70 and 130; targets with incomplete embedding caches stopped earlier with unstable validation curves.

Table 9: Representative Task 2 training outcomes by target type. Best LR from nine-head bank by validation BalAcc.

Target type	Example	Typical best LR	Val epochs
Whole-volume	Target A	2×10^{-4}	∼110
Whole-volume	Target B	10^{-4}	∼95
ROI	Target C	2×10^{-4}	∼85
ROI	Target D	5×10^{-4}	∼120
ROI (low coverage)	Target E	10^{-4}	∼70
ROI (low coverage)	Target F	2×10^{-5}	∼75

Crucially, our Task 2 docker inference respects the $\leq$40 s per-case budget: CT-NEXUS feature extraction accounts for over 90% of total latency, while the cross-attention head adds only a few milliseconds offline on GPU.

7 Conclusion

We presented an imbalance-aware cross-attention pooling pipeline for Task 2 (EAO, VO track), built on frozen CT-NEXUS spatial tokens, a nine-head learning-rate bank, and per-target validation threshold calibration. On the Codabench validation leaderboard, our Task 2 submission ranks #10 of 18, achieving mean balanced accuracy 0.727 and AUROC 0.718 across 15 AMOS targets. ROI targets lag primarily where embedding extraction remains incomplete. Our open pipeline supports reproducible Task 2 docker extraction, head training, and Codabench submission end to end.

Disclaimer on Use of Generative AI Generative AI tools were used during preparation of this manuscript for paraphrasing, grammar and style editing, literature organization, and LaTeX formatting assistance. The authors reviewed, edited, and take full responsibility for all content. All technical claims, experimental design, implementation details, results, and conclusions were verified by the authors.

Acknowledgements We thank the CVPR 2026 challenge organizers for datasets and baseline docker images, and CodaBench [11] for evaluation infrastructure.

Compute resources were provided in part through FABRIC [2]. This work was supported in part by the National Science Foundation under Grant No. 2502893.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Agrawal, K.K., et al.: Pillar-0: A new frontier for radiology foundation models. arXiv preprint arXiv:2511.17803 (2025) 2
2. Baldin, I., Nikolich, A., Griffioen, J., Monga, I.I.S., Wang, K.C., Lehman, T., Ruth, P.: FABRIC: A national-scale programmable experimental network infrastructure. IEEE Internet Computing **23**(6), 38–47 (2019) 2, 13
3. Blankemeier, L., et al.: Merlin: A vision language foundation model for 3D computed tomography. Research Square (2024), rs-3 2
4. Claessens, C., et al.: Scaling self-supervised and cross-modal pretraining for volumetric CT transformers. arXiv preprint arXiv:2511.17209 (2025) 2
5. Dancette, C., et al.: Curia: A multi-modal foundation model for radiology. arXiv preprint arXiv:2509.06830 (2025) 2
6. Dosovitskiy, A., Beyer, L., Kolesnikov, A., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. In: ICLR (2021) 2
7. Ericsson, L., Gouk, H., Hospedales, T.M.: How well do self-supervised models transfer? CVPR (2021) 1
8. Pai, S., et al.: Vision foundation models for computed tomography. arXiv preprint arXiv:2501.09001 (2025) 1, 2, 3, 4, 10
9. Wu, L., Zhuang, J., Chen, H.: Large-scale 3D medical image pre-training with geometric context priors. IEEE Transactions on Pattern Analysis and Machine Intelligence (2025) 2
10. Xu, T., et al.: A generalizable 3D framework and model for self-supervised learning in medical imaging. npj Digital Medicine **8**(1), 639 (2025) 2
11. Xu, Z., Escalera, S., Pavao, A., et al.: Codabench: Flexible, easy-to-use, and reproducible meta-benchmark platform. Patterns **3**(7), 100543 (2022) 12