Threshold Calibration and Multi-Seed Ensembling for Robust Embedding Aggregation on a Frozen CT Foundation Model

Aviral Kaintura[1][0009−0005−9508−2997]

National Forensic Sciences University, Delhi Campus, India
102ctbmti2122019@nfsu.ac.in

Abstract. We address Task 2 (Embedding Aggregation Optimization, EAO), Vision-Only track, of the CVPR 2026 Challenge on Foundation Models for General CT Image Diagnosis. The challenge fixes both the backbone choice (CT-NEXUS or VoCo-L) and the classification head architecture (a multi-layer cross-attention pooling head), so the available lever for improvement is entirely in the training recipe and post-processing applied on top of frozen embeddings. Using CT-NEXUS as a frozen feature extractor, we train per-target heads with mixup and label smoothing, ensemble across three random seeds (with a fourth added for the five weakest targets) and three learning rates per target, and apply two distinct post-processing steps: a per-target decision-threshold sweep over observed validation probabilities, and a defensive polarity-inversion rule for any target whose validation AUROC falls below 0.5. Starting from a single-seed default-threshold baseline of 0.5745 balanced accuracy, this recipe reaches **0.7086 balanced accuracy and 0.7280 AUROC** on the validation set, averaged over 17 binary classification targets spanning chest and abdominal CT. We show that threshold calibration alone accounts for the majority of this improvement, and we report which components of the recipe transfer cleanly versus which should be read with caution.

Keywords: CT foundation model · Embedding aggregation · Threshold calibration · Class imbalance

1 Introduction

CT foundation models have recently demonstrated strong representation quality across a range of clinical diagnostic tasks, but two problems persist in practice. First, generalization across anatomical regions and imaging protocols remains unreliable: a backbone trained predominantly on one body region or scanner family does not automatically transfer its representation quality elsewhere. Second, downstream evaluation in realistic clinical settings is dominated by severe class imbalance, in which many disease findings are rare by construction, and a downstream classifier evaluated under a fixed 0.5 decision threshold can appear to have learned nothing even when its underlying ranking of cases is informative. Optimal adaptation strategies for turning a frozen foundation model's

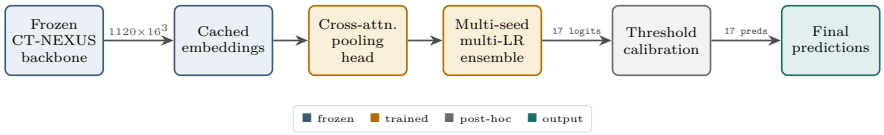

Fig. 1. Inference pipeline. The CT-NEXUS backbone (slate, frozen) is used only to produce cached spatial embeddings; a learnable cross-attention pooling head (amber, trained) aggregates these into per-target logits. A multi-seed/multi-learning-rate ensemble averages predictions, after which a post-hoc per-target threshold calibration (grey) yields final predictions (teal), which are evaluated by balanced accuracy and AUROC. An AUROC-gated polarity-inversion safety net is applied during calibration but was never triggered on any of the 17 validation targets in this submission.

embeddings into a usable per-disease classifier, under this imbalance, remain comparatively underexplored relative to the pretraining methods themselves.

A growing set of CT foundation models targets the vision-only setting, including VoCo [17], CT-FM [13], 3DINO [18], and Curia [14], alongside vision-language models such as Merlin [3], CT-CLIP [7], and Pillar-0 [1], and hybrid approaches such as SPECTRE [5]. Across this literature, adaptation to a downstream task is typically studied through linear probing [2] or lightweight fine-tuning such as LoRA [10] or adapters [9]. These studies focus on the adaptation mechanism and seldom report the downstream *recipe* in detail: how a frozen backbone's embeddings are aggregated into a calibrated, ensembled, deployable classifier under the kind of severe per-target class imbalance that real diagnostic panels exhibit.

This paper studies that recipe directly, under conditions where the backbone and head architecture are not free choices: the CVPR 2026 EAO Vision-Only track restricts the backbone to CT-NEXUS or VoCo-L and fixes the classification head to a multi-layer cross-attention pooling design. Given this fixed setup, our contribution is in the training recipe and post-processing. We show that (i) decision-threshold calibration per target, swept over the set of observed validation probabilities rather than a fixed grid, is the single largest contributor to final performance; (ii) a defensive polarity-inversion rule, applied whenever a target's validation AUROC falls below 0.5, is a cheap safety net worth including even though it never fired on this challenge's 17 targets; and (iii) multi-seed, multi-learning-rate ensembling with AUROC-based (rather than accuracy-based) checkpoint selection brings the recipe to overall performance comparable to the organizer's published CT-NEXUS baseline, from a substantially weaker single-seed starting point, while one frequently-assumed component of our own recipe, stochastic weight averaging, in practice rarely engaged given the early-stopping schedule we used.

2 Method

2.1 Embedding Aggregation Strategy

We use CT-NEXUS [11] as a frozen feature extractor for the Vision-Only track. CT-NEXUS is an organizer-provided baseline backbone for this challenge; it has no separate publication or model card, and is referenced here via its distribution repository. We did not finetune the backbone at any point: every parameter update in this work is confined to the downstream head described below.

CT-NEXUS preprocesses each input volume with Z-score intensity normalization and resampling to isotropic 1 mm spacing, using two volume-size variants: a 320^3 crop for region-of-interest (ROI) diseases (using the foreground mask supplied with the challenge data) and a 160^3 crop otherwise. It emits a spatial embedding of shape $1120 \times 16 \times 16 \times 16$ per case, cached to disk as HDF5, so that all downstream head training operates on fixed embeddings rather than repeatedly re-running the backbone.

On top of these cached embeddings, we use the challenge-specified *MultiLayersCrossAttentionPooling* head: two stacked cross-attention layers, four attention heads, two learnable query vectors, and a feed-forward width equal to the embedding dimension. A final linear layer maps the concatenated query outputs to two per-target class logits. This head contains approximately 15.1 million trainable parameters, and because every architectural choice in this component is fixed by the challenge rules, the training recipe and post-processing described next are the only components we vary.

2.2 Classification Head and Loss

Each of the 17 binary targets is trained independently. We use cross-entropy loss with label smoothing ($\varepsilon = 0.1$) [16], combined with mixup ($\alpha = 0.2$) [20] to regularize the head given the small number of positive cases for several targets. For 11 of the 17 AMOS-derived targets that are localized to a specific organ or structure (e.g. adrenal hyperplasia, gallstone, kidney stone), we use the foreground mask supplied by the challenge to constrain the backbone's input region to the relevant anatomy; the remaining targets, including both chest-region targets (COVID-19 and lung nodule malignancy), use the whole-volume embedding.

To address class imbalance during training, we use a weighted random sampler so that minority-class cases are seen proportionally more often within a batch, rather than relying on a loss reweighting scheme. For each target we train several heads in parallel at different learning rates (a multi-LR sweep) so that targets with different optimal learning dynamics are not forced to share one setting, and we additionally enable stochastic weight averaging over the final portion of training. In practice, given the early-stopping patience we used (Table 2), weight averaging rarely had the opportunity to engage before training stopped, and several targets' best checkpoints reflect no averaging at all.

Beyond the multi-LR sweep, we repeat this entire training procedure across three random seeds, and at inference time average the softmax probabilities of

all surviving checkpoints for a given target, following the deep ensembles approach [12]: different seeds reach different regions of the loss landscape, so averaging their predictions reduces variance that no single run's checkpoint selection can correct for on its own. A focused retrain using one additional seed, targeting only the five weakest-performing targets after the main sweep, is layered onto this ensemble rather than replacing it (Table 2).

We select checkpoints by validation AUROC rather than balanced accuracy. This choice is deliberate: balanced accuracy is degenerate exactly in the regime several of our targets exhibited before threshold calibration, where a model defaults to predicting the majority class under the default 0.5 threshold and balanced accuracy collapses to 0.5 regardless of how informative the model's underlying ranking actually is [8]. AUROC, being threshold-independent, lets us select genuinely better-ranking checkpoints even when the operating threshold has not yet been chosen.

2.3 Post-processing

We apply two distinct post-processing steps and describe them separately, because they address different failure modes; treating them as a single "threshold optimization" step would conflate the two.

Threshold calibration. For each target independently, we sweep every value that actually occurs among the validation set's predicted probabilities, rather than a fixed grid, as a candidate decision threshold, and select the value that maximizes validation balanced accuracy [15]. This mirrors empirical thresholding approaches that pick a decision boundary from observed scores rather than assuming a fixed cutoff is appropriate under unknown class priors. Sweeping the observed values rather than a coarse fixed grid matters in practice: several of our targets have skewed probability distributions where the optimal threshold sits well away from 0.5 (e.g. a target with most predicted probabilities clustered near 0.1), and a coarse grid can miss the exact value at which the balanced-accuracy-maximizing class boundary falls.

Polarity-inversion safety net. For any target whose validation AUROC falls below 0.5, which indicates the model's probability ranking is anti-correlated with the true label (worse than random), we negate the predicted probability ($p \to 1 - p$) before threshold selection. This exploits the elementary fact that a ROC curve lying below the diagonal can be reflected above it by inverting the classifier [6]; it is included as a defensive guard rather than an expected mechanism, and on this submission it never triggered: all 17 targets had validation AUROC above 0.5 in the final ensemble, so no target's predictions were actually inverted in the version we submitted.

Both thresholds and the inversion decision are derived once from validation labels and merged directly into the inference script as fixed per-target constants, so that evaluation requires a single script invocation rather than a multi-step pipeline.

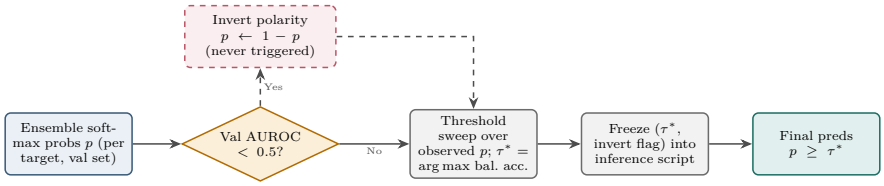

Fig. 2. Per-target post-processing, derived once from validation labels and frozen into the inference script. An AUROC gate optionally inverts a target's probabilities (a defensive safety net, never triggered on this submission) before a threshold sweep over the observed probability values selects the balanced-accuracy-maximising cutoff. Final predictions apply the frozen per-target threshold.

3 Experiments

3.1 Dataset and evaluation metrics

The challenge provides a downstream training and validation dataset of 17 binary classification targets across three datasets (AMOS, COVID-CT, LUNA25), spanning both chest and abdominal anatomy. We submit to Task 2, Vision-Only track, using CT-NEXUS as the fixed backbone. Of the 17 targets, two are chest-region (COVID-19 diagnosis from COVID-CT; lung nodule malignancy from LUNA25) and 15 are abdomen-region, derived from AMOS. Per-target validation set sizes range from 52 to 265 cases, with substantial class imbalance throughout.

We report **balanced accuracy** (the average of sensitivity and specificity) and **AUROC** (area under the receiver operating characteristic curve) as classification metrics, and per-case **running time**, measured including Docker startup, as the efficiency metric.

3.2 Implementation details

Preprocessing. All volume-level preprocessing (intensity normalization, resampling, ROI cropping) is performed inside the frozen CT-NEXUS feature extractor, as described in Section 2.1; we do not modify this step. Spatial embeddings are cached to HDF5 once per (target, case) pair, so that all subsequent head training, multi-seed/multi-LR sweeps, and threshold calibration operate on disk-cached features rather than repeatedly invoking the backbone.

Environment settings. Feature extraction and head training were performed on a rented cloud GPU instance; code development, packaging, and smoke testing were performed locally. Table 1 reports both environments, since they differ and conflating them would misstate what produced the submitted results.

The cloud GPU (24 GB) was deliberately chosen to match the organizers' stated evaluation hardware (2× NVIDIA RTX 4090, 24 GB each): a Docker container that fits within a 24 GB budget during development is far less likely to fail with an out-of-memory error during organizer-side evaluation.

Table 1. Development and training environments: the cloud instance used for feature extraction and head training, and the local machine used for development and packaging.

Component	**Cloud (training / feature extraction)**	**Local (development / packaging)**
System	Ubuntu 22.04 LTS	Pop!_OS 24.04 LTS
CPU	30 vCPUs (cloud instance)	16-core / 22-thread laptop CPU
RAM	200 GB	15 GB
GPU	1× NVIDIA A10, 24 GB	1× NVIDIA RTX 4060 Laptop GPU, 8 GB
Programming language	Python 3.12	Python 3.12.3
Deep learning framework	PyTorch	PyTorch 2.11.0

Training protocols. Data augmentation (mixup) and class-imbalance handling (weighted random sampling) are described in Section 2.2. We use AUROC-based checkpoint selection as the model-selection criterion, for the reasons given in Section 2.2. Table 2 reports the remaining training protocol, split into our main three-seed sweep across all 17 targets and a smaller focused retrain applied only to the five weakest-performing targets after the main sweep.

3.3 Quantitative results on the validation set

Figure 3 shows the progression of mean validation balanced accuracy and AUROC across six development iterations, from a single-seed default-threshold baseline through the final submission. Per-target threshold calibration is the single largest contributor: removing it from the final ensemble reduces mean balanced accuracy from 0.7086 to 0.5848, a +0.1238 difference that accounts for the majority of the total gain over the baseline. The same calibration step raises the single-seed baseline from v0 to v2 (+0.0830), with multi-seed ensembling and the focused retrain of the weakest targets contributing the remainder of the climb to 0.7086.

Table 3 reports final (v4) results by anatomical region against the organizer's published CT-NEXUS baseline. The organizer reports per-target baseline scores but no per-region or overall aggregate, so we compute region and overall means from their per-target values using the same unweighted mean over targets that defines the challenge metric [4], placing baseline and our results on the same footing. Under this comparison our recipe exceeds the baseline in every region, by +0.0099 balanced accuracy and +0.0128 AUROC overall, starting from a substantially weaker single-seed baseline of 0.5745. Because the head architecture and backbone are fixed, this margin is attributable to the training recipe and post-processing rather than to model capacity. Cost is identical for both rows

Table 2. Training protocol for Task 2 (Embedding Aggregation Optimization), Vision-Only track: hyperparameters for the main three-seed sweep over all 17 targets and the focused retrain of the five weakest targets.

Setting	**Main sweep (17 targets)**	**Focused retrain (5 weak targets)**
Backbone	CT-NEXUS (frozen)	CT-NEXUS (frozen)
Seeds	42, 43, 44	50
Batch size	8	8
Learning rates (parallel heads)	5e-4, 1e-3, 5e-3	5e-5, 1e-4, 5e-4
Epoch cap	60	100
Early-stopping patience	15	30
SWA start fraction	0.6*	0.6*
Mixup α	0.2	0.0
Label smoothing ε	0.1	0.0
Optimizer	AdamW, no LR schedule	AdamW, no LR schedule
Checkpoint selection	Validation AUROC	Validation AUROC
Loss	Cross-entropy + label smoothing	Cross-entropy
Trainable parameters	15.1M per head	15.1M per head

* Early stopping frequently triggered before this point was reached; see Section 2.2.

and we report it once rather than per row: feature extraction runs at 12–16 s per case, within the organizer's 40 s/case Docker budget, the head has 15.1M trainable parameters, and head inference and ensembling run outside Docker and add under one second per case. We do not report a VoCo-L baseline row, since we did not empirically benchmark VoCo-L within this work: CT-NEXUS was selected for engineering reasons (an already-working baseline Docker), not from a measured comparison, and we state this as a limitation rather than imply a comparison we did not run.

Figure 4 breaks performance down to the individual target level. Per-target balanced accuracy ranges widely, from fatty liver (0.8462) and hydronephrosis (0.8281) at the top to lung nodule malignancy (0.6100) and liver cyst (0.6222) at the bottom. The two chest-region targets bracket this range, since COVID-19 sits near the median while lung nodule malignancy is the single weakest target overall, which indicates that the difficulty is target-specific rather than cleanly split by anatomy. For most targets AUROC sits slightly above balanced accuracy, the expected gap between a threshold-independent ranking metric and a calibrated operating point.

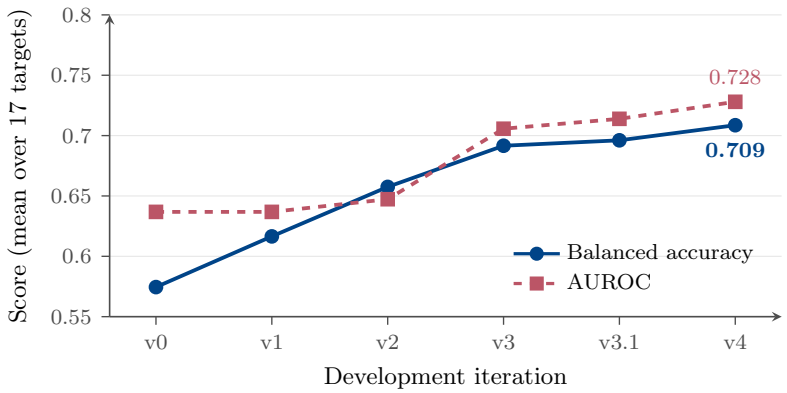

Fig. 3. Validation balanced accuracy and AUROC, averaged over all 17 targets, across six development iterations: v0 (single-seed baseline, default 0.5 threshold), v1 (per-target threshold sweep on a fixed linear grid), v2 (finer unique-probability threshold sweep with the AUROC-gated polarity-inversion safety net added), v3 (3-seed ensemble), v3.1 (pruned checkpoint selection), and v4 (focused retrain of the five weakest targets, final submission). Annotated values are the final-iteration (v4) scores. AUROC is unchanged between v0 and v1 since threshold choice does not affect ranking quality.

Table 3. Task 2 (EAO), Vision-Only track: mean balanced accuracy and AUROC over the 17 validation targets (↑ higher is better) by region, with per-case runtime and trainable head parameters. Both rows share the frozen CT-NEXUS backbone and fixed attention-pooling head, differing only in recipe and post-processing. **Bold** marks the better value per region; baseline means are our aggregation of the organizer's per-target scores [4].

Region	Method	BalAcc	AUROC
Chest (2 targets)			
Organizer baseline	Default recipe	0.6705	0.6700
Ours	Calibrated + ensembled	**0.6878**	**0.6865**
Abdomen (15 targets)			
Organizer baseline	Default recipe	0.7025	0.7213
Ours	Calibrated + ensembled	**0.7114**	**0.7336**
Overall (17 targets)			
Organizer baseline	Default recipe	0.6987	0.7152
Ours	Calibrated + ensembled	**0.7086**	**0.7280**

3.4 Qualitative analysis on the validation set

Figure 5 shows one easy (correctly classified, high-confidence) and one hard (misclassified) case for each of three targets spanning both anatomical regions. For

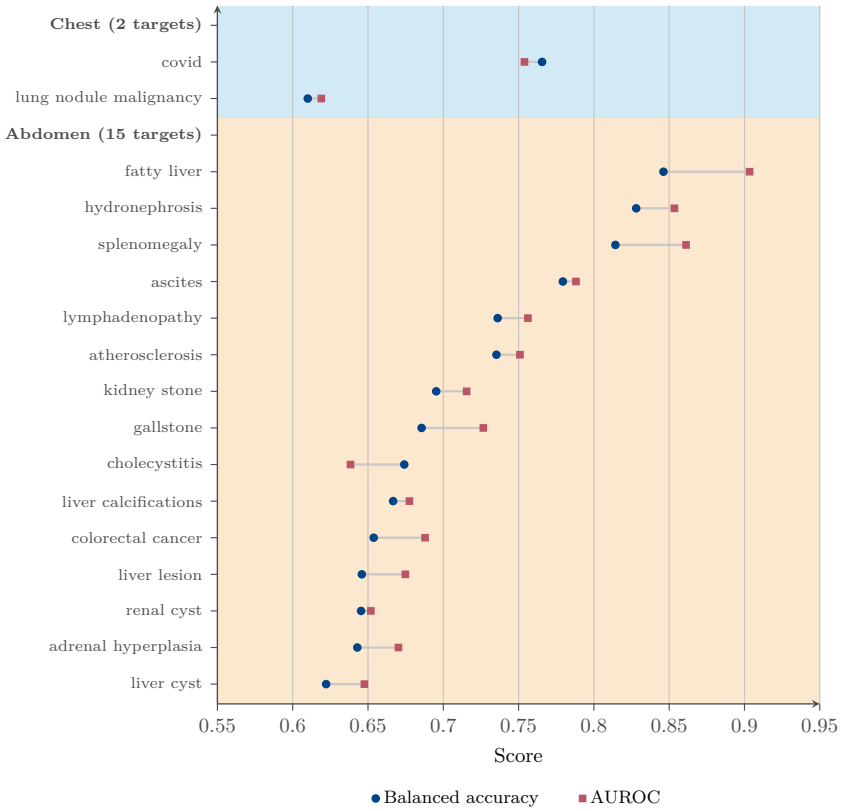

Fig. 4. Validation balanced accuracy (blue circles) and AUROC (red squares) for all 17 binary targets, from our final submitted model (development iteration v4, Fig. 3). Targets are grouped by anatomical region, Chest (2 targets) and Abdomen (15 targets), matching the grouping in Table 3, and sorted by balanced accuracy within each region. The connecting line spans the two metric values per target; all 17 targets exceed chance (0.5).

COVID-19, the hard case is a false negative with a predicted probability (0.4714) just below the calibrated threshold despite a true positive label, illustrating that threshold calibration narrows but does not eliminate boundary-case errors. For lung nodule malignancy, the hard case is a false positive with a visually subtle nodule appearance similar to the easy true-positive case, consistent with the comparatively low AUROC (0.6190) this target achieves relative to others. For fatty liver, both cases have high predicted probability; the hard case is a false positive, suggesting the model has learned a visual pattern (e.g. diffuse liver attenuation change) that is present but not diagnostic in this particular case.

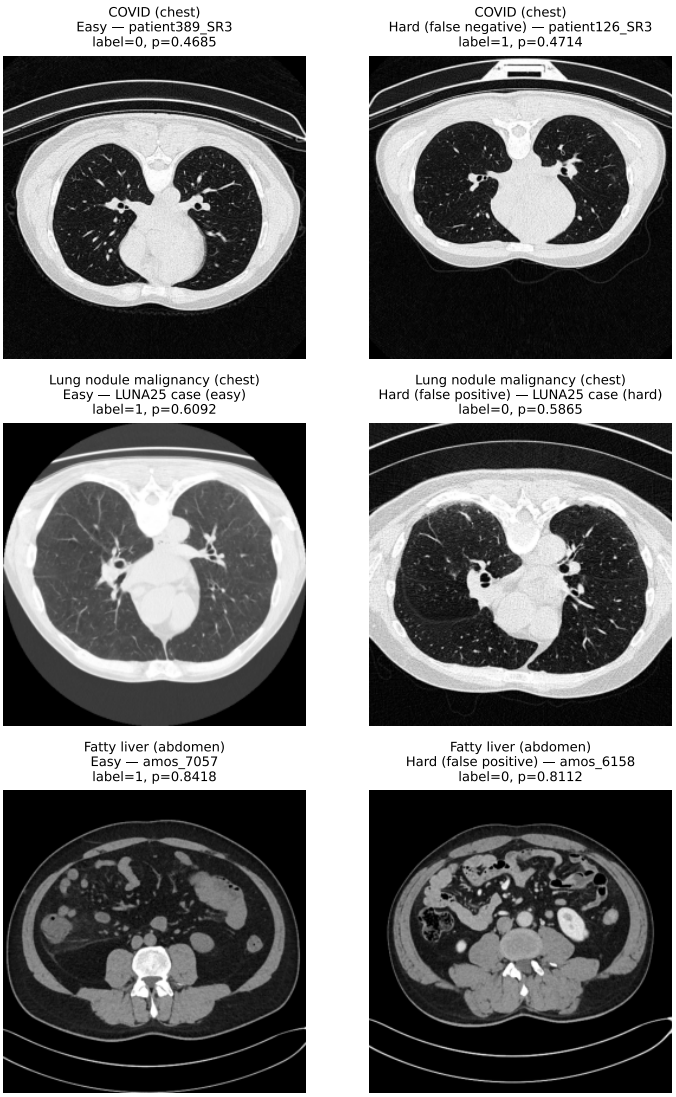

Fig. 5. Qualitative examples: one easy (correctly classified) and one hard (misclassified) case for COVID-19 and lung nodule malignancy (chest) and fatty liver (abdomen), shown as a representative axial slice with the predicted probability of the positive class and the true label.

3.5 Results on the final testing set

The hidden test-set results were not available at submission time, so this section will be completed during the revision phase, once the challenge results are announced.

3.6 Limitation and future work

This work has several limitations we state plainly. First, we did not finetune or otherwise adapt the CT-NEXUS backbone itself; all improvement comes from the downstream recipe, which means our results say nothing about whether a different backbone choice (VoCo-L, the other permitted Vision-Only backbone) would perform better or worse under the same recipe; we did not run that comparison. Second, our use of stochastic weight averaging engaged inconsistently across targets and seeds, since early stopping frequently triggered before the averaging window opened; we report SWA as part of our configuration but do not claim it materially contributed to the final result, and a future iteration with a longer patience budget could test this more cleanly. Third, per-target threshold calibration on validation sets as small as 52 cases carries a real risk of fitting validation-set noise rather than a transferable decision boundary; the polarity-inversion safety net in particular never triggered on this challenge's targets, so its value here is unverified beyond serving as a cheap guard against a failure mode that did not occur. Fourth, we did not attempt the Vision-Language track (Merlin or Pillar-0) or Task 1 (Linear Probing); both were deliberately out of scope given the available development time. Finally, the hidden test-set result for this submission was not yet available at the time of writing; Section 3.5 will be updated with that result during the revision phase, and the validation-to-test generalization gap, if any, is correspondingly unknown at this time.

4 Conclusion

Within a challenge that fixes both the backbone and the classification head architecture, we show that post-processing and ensembling alone can move mean validation balanced accuracy from 0.5745 to 0.7086 across 17 imbalanced binary classification targets, with per-target threshold calibration, swept over observed validation probabilities rather than a coarse grid, responsible for the majority of this gain. We report this recipe, and its individual components' measured contributions, with explicit attention to which parts of it (threshold calibration, AUROC-based checkpoint selection) we are confident in, and which (stochastic weight averaging, the unused polarity-inversion safety net) should be read as configured-but-not-validated on this particular submission.

Code and data availability. This submission builds on the official challenge code.[1] Our trained per-target heads, the single-script inference (`predict_eao_ensemble.py`, with the per-target thresholds baked in), and the feature-extraction Docker are publicly available at https://huggingface.co/datasets/procodec/cvpr26-vo-eao under an MIT license for our code, with the CT-NEXUS backbone redistributed as provided by the organizers.

[1] https://github.com/kmin940/CVPR26-3DCTFMCompetition

Acknowledgements We thank the data owners for making the AMOS, COVID-CT, and LUNA25 datasets publicly available, the organizers of the CVPR 2026 Workshop on Foundation Models for Medical Vision for hosting this challenge, and CodaLab [19] for hosting the challenge platform.

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., Liu, L., Lian, L., et al.: Pillar-0: A new frontier for radiology foundation models. arXiv preprint arXiv:2511.17803 (2025) 2
2. Alain, G., Bengio, Y.: Understanding intermediate layers using linear classifier probes. In: ICLR Workshop Track (2017), arXiv:1610.01644 2
3. Blankemeier, L., Kumar, A., et al.: Merlin: a computed tomography vision-language foundation model and dataset. Nature (2026). https://doi.org/10.1038/s41586-026-10181-8, originally released as arXiv:2406.06512, 2024 2
4. Challenge Organizers: CVPR 2026 challenge on foundation models for general CT image diagnosis: Task 2 (embedding aggregation optimization), vision-only track. https://www.codabench.org/competitions/12650/, codabench competition 12650, "Baselines" tab. Organizer-published per-target CT-NEXUS EAO baseline scores; region and overall means reported in this work are our own unweighted mean over targets, matching the challenge metric. Accessed 2026-07-01 6, 8
5. Claessens, C., Viviers, C., D'Amicantonio, G., Bondarev, E., van der Sommen, F.: Scaling self-supervised and cross-modal pretraining for volumetric ct transformers. arXiv preprint arXiv:2511.17209 (2025) 2
6. Fawcett, T.: An introduction to roc analysis. Pattern Recognition Letters **27**(8), 861–874 (2006). https://doi.org/10.1016/j.patrec.2005.10.010 4
7. Hamamci, I.E., et al.: A foundation model utilizing chest ct volumes and radiology reports for supervised-level zero-shot detection of abnormalities. arXiv preprint arXiv:2403.17834 (2024) 2
8. He, H., Garcia, E.A.: Learning from imbalanced data. IEEE Transactions on Knowledge and Data Engineering **21**(9), 1263–1284 (2009). https://doi.org/10.1109/TKDE.2008.239 4
9. Houlsby, N., Giurgiu, A., Jastrzebski, S., Morrone, B., De Laroussilhe, Q., Gesmundo, A., Attariyan, M., Gelly, S.: Parameter-efficient transfer learning for nlp. In: Proceedings of the 36th International Conference on Machine Learning (ICML). vol. 97, pp. 2790–2799 (2019) 2
10. 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 preprint arXiv:2106.09685 (2021) 2
11. kmin940: Ct-nexus. https://github.com/kmin940/CT-NEXUS, organizer-provided baseline backbone for the CVPR 2026 Challenge on Foundation Models for General CT Image Diagnosis (Codabench competition 12650); no separate publication or model card exists 3
12. Lakshminarayanan, B., Pritzel, A., Blundell, C.: Simple and scalable predictive uncertainty estimation using deep ensembles. In: Advances in Neural Information Processing Systems (NeurIPS) (2017), arXiv:1612.01474 4

13. Pai, S., Hadzic, I., Bontempi, D., Bressem, K., Kann, B.H., Fedorov, A., Mak, R.H., Aerts, H.J.W.L.: Vision foundation models for computed tomography. arXiv preprint arXiv:2501.09001 (2025) 2
14. Raidium: Curia: A multi-modal foundation model for radiology. arXiv preprint arXiv:2509.06830 (2025) 2
15. Sheng, V.S., Ling, C.X.: Thresholding for making classifiers cost-sensitive. In: Proceedings of the 21st National Conference on Artificial Intelligence (AAAI). pp. 476–481 (2006) 4
16. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: Rethinking the inception architecture for computer vision. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2016), arXiv:1512.00567 3
17. 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). https://doi.org/10.1109/TPAMI.2025.3639593, arXiv:2410.09890 2
18. Xu, T., Hosseini, S., Anderson, C., Rinaldi, A., Krishnan, R.G., Martel, A.L., Goubran, M.: A generalizable 3d framework and model for self-supervised learning in medical imaging. npj Digital Medicine **8**, 639 (2025). https://doi.org/10.1038/s41746-025-02035-w 2
19. Xu, Z., Escalera, S., Pavaõ, A., Richard, M., Tu, W.W., Yao, Q., Zhao, H., Guyon, I.: Codabench: Flexible, easy-to-use, and reproducible meta-benchmark platform. Patterns **3**(7), 100543 (2022). https://doi.org/10.1016/j.patter.2022.100543 12
20. Zhang, H., Cisse, M., Dauphin, Y.N., Lopez-Paz, D.: mixup: Beyond empirical risk minimization. In: International Conference on Learning Representations (ICLR) (2018), arXiv:1710.09412 3

Table 4. Checklist Table. (Delete this Table in the camera-ready submission)

Requirements	Answer
A meaningful title	Yes
The number of authors ($\leq$6)	1
Author affiliations and ORCID	Yes
Corresponding author email is presented	Yes
Task and track (LP All / LP Coreset / EAO VO / EAO VL) are clearly stated in the abstract	Yes
Validation BalAcc and AUROC are presented in the abstract	Yes
Introduction includes at least three parts: background, related work, and motivation	Yes
A pipeline/network figure is provided	Figure 1
Pre-processing	Sec. 3.2.1
Strategies for data augmentation	Sec. 2.2
Strategy for class imbalance	Sec. 2.2
Strategies to improve model inference (under 40s/case)	Sec. 3.2 (Table 3 footnote)
Post-processing	Sec. 2.3
Environment setting table is provided	Table 1
Training protocol table is provided (Task 1 and/or Task 2)	Table 2
Ablation study	Sec. 3.3, Figs. 3–4
Efficiency evaluation results (running time) are provided	Table 3
Qualitative example (success and failure) is provided	Figure 5
Limitation and future work are presented	Sec. 3.6
Reference format is consistent	Yes
Main text $\geq$ 8 pages (not including references and appendix)	Yes (confirmed at compile time)