ADAPTIVE COMPUTATION WITH CONFORMAL CERTIFICATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Scientific imaging data involving medical (e.g., 3D radiographic, microscopic images) and astronomical sensors pose different challenges from natural images. One key challenge is that these datasets are often orders of magnitude larger than natural images. Efficient training and inference on such data is challenging due to the increased hardware requirements, often needing bespoke solutions. Moreover, the relevant objects such as tumors and astronomical phenomena are sparsely distributed with respect to the background information. Standard efficiency techniques like quantization and pruning apply uniform scaling of all regions, potentially compromising the precision/fidelity of predictions. Thus, efficient methods that reduce computational requirements without losing resolution for relevant parts of the data and with uncertainty estimates on predictions are essential. We present *Conformal Adaptation* with risk guarantees for uncertainty-aware sparse computations. Our formalism is general, spanning dense pixel-level classification and regression tasks where hierarchical models that predict at low resolution, certify the signal using conformal risk control, and refine the certified predictions. We benchmark our method on multiple tasks such as 3D medical image segmentation, 2D galaxy flux prediction, and 1D calcium spike rate prediction, achieving 2.5–$72\times$ wall-clock speedup across the various tasks, including on commodity CPUs.

1 INTRODUCTION

Deep learning models have expanded far beyond benchmark vision and natural language processing tasks, becoming foundational components of high-throughput scientific pipelines ranging from drug discovery and weather forecasting to medical imaging and astronomical surveys. However, deploying these models in real-world environments frequently exposes a fundamental tension between predictive capacity and resource constraints (Jazbec et al., 2024). While frontier models deliver state-of-the-art accuracy, their practical utility rapidly degrades when constrained by edge hardware, strict latency budgets, or extreme data scales. As computational access becomes a bottleneck across disciplines, designing models that are efficient, scalable, and trustworthy has become a central challenge.

This bottleneck also carries severe environmental and structural costs. Maximizing benchmark performance via scale demands unsustainable energy and carbon footprints (Schwartz et al., 2020; Mehlin et al., 2023). It also raises steep capital barriers, concentrating research within well-funded enterprise labs while sidelining academic and clinical researchers (Guan et al., 2024). Because hardware scaling yields diminishing returns, algorithmic efficiency is critical for responsible scientific computing.

Practitioners pursue efficiency along two broad routes. Hardware specialization and kernel-level tuning preserve computational fidelity but require long development cycles and capital-intensive infrastructure. Algorithmic optimizations such as post-training quantization (Liu et al., 2024; Chen et al., 2025), structural pruning (Liu et al., 2018; Blalock et al., 2020), low-rank factorization (Hu et al., 2021; Dettmers et al., 2023), and distillation (Gou et al., 2021) trade some precision for large runtime reductions on commodity hardware. On standard vision and NLP benchmarks this tradeoff is usually benign, and sub-8-bit representations are now common practice. These techniques are increasingly applied in safety-critical settings (Pang et al., 2025; Zeng et al., 2026; Alwadee

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

et al., 2025), where they are typically validated by aggregate metrics but can still degrade performance (Alcaraz-Ortiz et al., 2026). Many dense scientific tasks, however, are characterized by extreme spatial sparsity: tumors in a positron emission tomography (PET) volume or rare transients in a sky survey occupy a tiny fraction of the data. In this regime an aggregate metric can remain high while revealing little about the few regions that matter, and the cost of an error (a missed lesion, a false discovery) is concentrated there. Practitioners therefore benefit from acceleration methods whose risk on the sparse signal is explicitly bounded.

We approach efficiency along a complementary axis. Rather than reducing the cost of each evaluation, we reduce the number of locations that require expensive evaluation and certify that decision. We introduce *Conformal Adaptation*, a framework for allocating compute across dense prediction tasks. A low-resolution pass identifies candidate regions of interest, and conformal risk control (Angelopoulos et al., 2024) calibrates this selection to provide finite-sample, distribution-free bounds on the expected fraction of signal missed, at a user-specified level. A hierarchy of sparse operators then refines only the selected regions during both training and inference, and we show that the calibrated bound is preserved through refinement under two stated conditions. As the coarse screening model is an ordinary network, *Conformal Adaptation* composes naturally with quantization, pruning, and other model-level compression techniques.

Contributions Our primary contributions are **(a)** an end-to-end framework combining conformalized candidate selection with sparse execution, delivering significant training and inference speedups backed by conformal risk control (CRC) guarantees, **(b)** a general formulation of admissible label aggregation operators that makes the framework applicable across tasks, and **(c)** case studies spanning multiple scientific domains, data modalities, and sparsity regimes that empirically validate the guarantees and quantify the compute savings.

The manuscript is organized as follows. We first establish common notation for dense predictions, sparse refinement solutions, and CRC in Sec. 2. We then detail the *Conformal Adaptation* algorithm in Sec. 3 and define our performance metrics in Sec. 4. We showcase the applicability of our method across segmentation (Sec. 5) and regression tasks (Sec. 6), review related work in Sec. 7, and discuss limitations, trade-offs, and future work in Sec. 8.

2 BACKGROUND

Notation. We focus on dense prediction tasks whose domains admit *hierarchical* partitionings. Inputs X and outputs Y are spatially discretized, with $X \in \mathbb{R}^{|\Omega| \times l}$ and $Y \in \mathbb{R}^{|\Omega| \times k}$, where Ω is a finite set of *sites* (pixels in 2D, voxels in 3D, or more generally any indexed cell) and l and k are the numbers of input and output channels, respectively. We say a coarse partitioning Ω_{coarse} *coarsens* a fine partitioning Ω_{fine}, written $\Omega_{\text{coarse}} \preceq \Omega_{\text{fine}}$, when every cell of Ω_{coarse} is a disjoint union of cells of Ω_{fine}; equivalently, the fine partitioning refines the coarse one, and each fine site lies in exactly one coarse cell.

We seek to learn a non-linear function $\mathcal{F}_\theta : \mathcal{X} \to \mathcal{Y}$. Due to hierarchical partitioning, we consider $\mathcal{F}_\theta$ that decomposes into a coarse model followed by a refinement operator, $\mathcal{F}_\theta(X) = \mathcal{G}\big(X^{\Omega_{\text{fine}}}, \mathcal{F}_{\Omega_{\text{coarse}}}(X^{\Omega_{\text{coarse}}})\big)$, so that predictions can be performed at multiple levels. It is far cheaper to predict on Ω_{coarse} and refine than to predict directly on Ω_{fine} with a full-resolution model $\mathcal{F}_{\Omega_{\text{fine}}}$.

Conformal Risk Control. We turn to CRC (Angelopoulos et al., 2024) as a distribution-free black-box method for providing prediction guarantees at coarse resolution.

Given a per-image loss $L(\lambda; X, Y)$ that, for every fixed (X, Y), is non-increasing and right-continuous in λ, bounded in $[0, B]$, and satisfies $L(1; X, Y) \leq \alpha$, together with calibration data $(X_i, Y_i)_{i=1}^n$ exchangeable with the test point (X_{n+1}, Y_{n+1}), the threshold

$$\hat{\lambda} \;=\; \inf\Big\{\lambda : \tfrac{1}{n+1}\big(\textstyle\sum_{i=1}^n L(\lambda; X_i, Y_i) + B\big) \leq \alpha\Big\} \tag{1}$$

guarantees,

$$\mathbb{E}\big[L(\hat{\lambda}; X_{n+1}, Y_{n+1})\big] \leq \alpha \tag{2}$$

The goal of *Conformal Adaptation* is to construct a loss L and hierarchical refinement operations, $\mathcal{G}$, satisfying the CRC requirements at partitioning Ω_{coarse} while presenting the output at partitioning Ω_{fine}. Operating on the lower resolution Ω_{coarse} reduces computation, and sparsifying Ω_{coarse} by selecting candidate cells for refinement with CRC further increases efficiency without compromising fidelity.

3 Conformal Adaptation

Conformal Adaptation proposes a trade-off between statistical guarantees and spatial fidelity to speed up computation through sparsification. We are primarily concerned with loss of fidelity due to prediction at the coarser level and seek to close the gap in prediction uncertainty with respect to the finest resolution. Hence, our current focus does not extend the CRC framework to the finest resolution. We first establish an abstract formulation to highlight the generalizability of our method, then provide specializations to individual use cases including segmentation and regression.

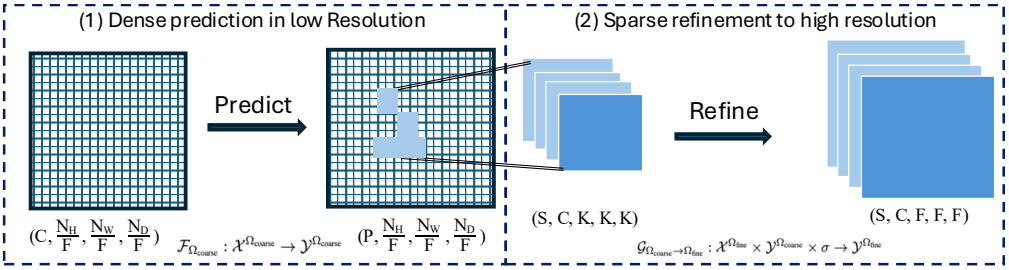

Figure 1: The two-stage process for *Conformal Adaptation* consists of **(1)** dense predictions at a spatial scale reduced by a factor r and **(2)** sparse refinement on the selected set S of coarse cells. The selection is based on thresholds calibrated using CRC, and the operating regime we target is $|S| \ll |\Omega_{\text{coarse}}|$.

Exploiting the spatial coherence of adjacent units in an image-like spatial grid, we reformulate dense prediction as a coarse prediction using an aggregation operator A followed by *selective* refinement using an operator $\mathcal{G}$ on relevant regions needing additional computation:

$$\mathcal{F}_{\Omega_{\text{coarse}}} : \mathcal{X}^{\Omega_{\text{coarse}}} \to \mathcal{Y}^{\Omega_{\text{coarse}}}, \tag{3}$$

$$\mathcal{G}_{\Omega_{\text{coarse}} \to \Omega_{\text{fine}}} : \mathcal{X}^{\Omega_{\text{fine}}} \times \mathcal{Y}^{\Omega_{\text{coarse}}} \times \{0,1\}^{\Omega_{\text{coarse}}} \to \mathcal{Y}^{\Omega_{\text{fine}}}, \quad \Omega_{\text{coarse}} \preceq \Omega_{\text{fine}} \tag{4}$$

We define a binary selection function $\sigma : \Omega_{\text{coarse}} \to \{0,1\}$ over the coarse partition $\Omega_{\text{coarse}} = \{c_1, \ldots, c_K\}$, where a coarse cell $c_k \in \Omega_{\text{coarse}}$ is selected for refinement if and only if $\sigma(c_k) = 1$. The resulting set of selected cells is $S_\sigma = \{c_k \in \Omega_{\text{coarse}} : \sigma(c_k) = 1\} \subseteq \Omega_{\text{coarse}}$. σ is derived from the coarse prediction $\hat{O}^{\Omega_{\text{coarse}}} = \mathcal{F}_{\Omega_{\text{coarse}}}(X^{\Omega_{\text{coarse}}})$. Our framework allows for dense pixel-level categorical (binary, multi-class) as well as continuous (regression) predictions. The refinement $\mathcal{G}_{\Omega_{\text{coarse}} \to \Omega_{\text{fine}}}\left(X^{\Omega_{\text{fine}}}, \hat{O}^{\Omega_{\text{coarse}}}, \sigma\right)$ acts piecewise on the *fine* sites of each *coarse* cell:

$$\mathcal{G}_{\Omega_{\text{coarse}} \to \Omega_{\text{fine}}}\big|_{c_k} = \mathcal{G}_{c_k}\left(X^{\Omega_{\text{fine}}}\big|_{c_k}, \hat{O}^{\Omega_{\text{coarse}}}[c_k]\right) \qquad c_k \in S_\sigma \tag{5}$$

Unselected cells of Ω_{coarse} are filled by the null value or simple upsampling, to preserve the prediction structure at fine level. The general inference algorithm is visualized in Fig. 1. The cost for the refinement scales with $|S_\sigma|$ rather than $|\Omega_{\text{fine}}|$, so a smaller selection is cheaper. The task is thus to make S_σ as small as possible without discarding signal that matters, a trade-off we control with a statistical guarantee as described in the following paragraphs. The detailed training, calibration, and inference algorithms are provided in Appendix A.

Conformal cell selection from the coarse prediction. The selection set must be a function of information available at test time, i.e. of the input through the coarse prediction. We obtain it from a coarse *score field* $s : \mathcal{X}^{\Omega_{\text{coarse}}} \to [0,1]^{\Omega_{\text{coarse}}}$, read out from $\hat{O}^{\Omega_{\text{coarse}}}$, thresholded at a level $\lambda \in [0,1]$:

$$\sigma_\lambda(X)[c_k] = \mathbb{I}\left[s(X^{\Omega_{\text{coarse}}})[c_k] \geq 1 - \lambda\right] \tag{6}$$

$$S_\lambda(X) = \{c_k : \sigma_\lambda(X)[c_k] = 1\} \tag{7}$$

Note that the selection is monotone in λ, with $\lambda_1 \leq \lambda_2 \Rightarrow S_{\lambda_1}(X) \subseteq S_{\lambda_2}(X)$, and $\lambda = 1$ selects every cell. Crucially, the selection flag, σ_λ, depends on X alone, through the coarse prediction.

Cell aggregation from fine labels. Since predictions must happen in lower resolution, we must define an *aggregation operator* A that maps the fine labels inside a coarse cell to a *cell weight*. The cell weight scores the 'amount' of signal in each coarse cell by a single non-negative scalar.

$$w_{c_k}(Y) \;=\; A\Big(Y^{\Omega_{\text{fine}}}\big|_{c_k}\Big) \;\geq\; 0, \tag{8}$$

where $Y^{\Omega_{\text{fine}}}|_{c_k}$ denotes the restriction of the fine-resolution label to the fine sites contained in the coarse cell c_k. Because $\Omega_{\text{coarse}} \preceq \Omega_{\text{fine}}$, each coarse cell is a disjoint union of fine sites. Larger weights represent more signal within the coarse cell.

The fine sites of a coarse cell $c \in \Omega_{\text{coarse}}$ are the set $\Omega_{\text{fine}}|_c = \{u \in \Omega_{\text{fine}} : u \subseteq c\}$. Let $\varnothing \in \mathcal{Y}$ denote the null value, corresponding to background labels in segmentation or zero signal in regression. We require A to satisfy two conditions:

Condition 1 (Nullity)**.** For every $c \in \Omega_{\text{coarse}}$ and every $y \in \mathcal{Y}^{\Omega_{\text{fine}}|_c}$, $y \equiv \varnothing \implies A(y) = 0$.

Condition 2 (Occupancy-preservation)**.** For every $c \in \Omega_{\text{coarse}}$ and every $y \in \mathcal{Y}^{\Omega_{\text{fine}}|_c}$, $\exists\, u \in \Omega_{\text{fine}}|_c$ with $y[u] \neq \varnothing \implies A(y) > 0$.

Connecting to CRC. We can now define a loss meeting the requirements of Eq. 1. Let the loss $L(\lambda; X, Y)$ be the fraction of total signal mass that lies in *unselected* cells.

$$L(\lambda; X, Y) \;=\; \frac{\displaystyle\sum_{c_k \notin S_\lambda(X)} w_{c_k}(Y)}{\displaystyle\sum_{c_k \in \Omega_{\text{coarse}}} w_{c_k}(Y)} \tag{9}$$

$$\mathbb{E}\left[\frac{\displaystyle\sum_{c_k \notin S_{\hat{\lambda}}(X)} w_{c_k}(Y)}{\displaystyle\sum_{c_k \in \Omega_{\text{coarse}}} w_{c_k}(Y)}\right] \;\leq\; \alpha. \tag{10}$$

Note that L is bounded in $[0,1]$, as $w \geq 0$, and $0/0 := 0$ for examples with zero total ground-truth signal weight (or empty predictions). We require L to be non-increasing and right-continuous in λ. Increasing λ enlarges $S_\lambda(X)$ by Eq. 6, which removes non-negative terms from the numerator.

At $\lambda = 1$ every cell is selected, so the numerator is empty and $L(1; X, Y) = 0 \leq \alpha$. All conditions for applying CRC are therefore satisfied. Substituting Eq. 9 into the calibration of Eq. 1 yields a threshold $\hat{\lambda}$, which gives us the guarantee in Eq. 2 reformulated for signal loss in Eq. 10. Under exchangeability with a calibration set, we ensure that the unselected cells contain, in expectation, at most an α fraction of the true signal mass. The CRC guarantee is established entirely at Ω_{coarse} as it depends on X and Y through the cell aggregates $w_{c_k}(Y)$ and $S_{\hat{\lambda}}(X)$. We decouple the presentation of the solution from the certification of the solution.

3.1 Theoretical Guarantees

We call an aggregation operator A *admissible* if it is fixed and satisfies non-negativity (Eq. 8) and Conditions 1 and 2. We write $\mathcal{O}(Y) = \{c \in \Omega_{\text{coarse}} : \exists\, u \in \Omega_{\text{fine}}|_c,\ Y[u] \neq \varnothing\}$ for the set of coarse cells *occupied* by the fine label Y. These are the coarse cells that have positive signal mass. $\hat{Y}_\sigma^{\Omega_{\text{fine}}} = \mathcal{G}_{\Omega_{\text{coarse}} \to \Omega_{\text{fine}}}(X^{\Omega_{\text{fine}}}, \hat{O}^{\Omega_{\text{coarse}}}, \sigma)$ is the output of the sparse pipeline under selection σ, with $\sigma \equiv \mathbf{1}$ recovering dense refinement.

The aforementioned definition of admissible aggregation operators provides sufficient conditions to show that (i) the solution sets are nested in coarse resolution and monotone in λ, (ii) the missed

signal is proportional to the weight and bounded, and (iii) we obtain a marginal guarantee over discarded signal mass. Formally, we have:

Theorem 3.1. *Let A be admissible, let $(X_1, Y_1), \ldots, (X_n, Y_n), (X, Y)$ be exchangeable, where $(X, Y) = (X_{n+1}, Y_{n+1})$ is the test pair, and let $\hat{\lambda}$ be the CRC threshold obtained by calibrating Eq. 9 at level α on the first n pairs. Write $D = \Omega_{coarse} \setminus S_{\hat{\lambda}}(X)$ for the discarded cells. Then:*

(i) Almost surely, $\hat{Y}^{\Omega_{fine}}_{\sigma_{\hat{\lambda}}(X)}$ and $\hat{Y}^{\Omega_{fine}}_{\mathbf{1}}$ agree at every fine site of every cell in $S_{\hat{\lambda}}(X)$; the two pipelines can differ only on $\bigcup_{c \in D} \Omega_{fine}|_c$.

(ii) The true signal mass carried by that region is controlled:

$$\mathbb{E}\left[\frac{\sum_{c \in D} w_c(Y)}{\sum_{c \in \Omega_{coarse}} w_c(Y)}\right] \leq \alpha.$$

(iii) With probability at least $1 - \alpha/t$ for any $t \in (\alpha, 1]$, the discarded region carries at most a t-fraction of the signal mass; and the event that it carries none is exactly $\{\mathcal{O}(Y) \subseteq S_{\hat{\lambda}}(X)\}$, for every admissible A.

Proof sketch. Informally, (i) is a consequence of selectivity of the refinement operation. Coarsening a sparse refinement is the same as masking the coarsening of a dense prediction. The statement (ii) is the result of the admissible operator being compatible with CRC. Finally, (iii) follows from Markov's inequality applied to the non-negative loss. The detailed proof can be found in Appendix B. □

4 Performance Metrics

We focus on both dense classification and regression tasks from varying domains. Segmentation tasks use Dice similarity coefficient (DSC) and lesion detection metrics as primary accuracy targets. The spike count prediction and galaxy super-resolution use mean absolute error (MAE). For performance metrics, we audit both wall-clock speedup as well as FLOPs. We use both CPUs and GPUs to obtain timing results.

5 Case Study: Conformal Adaptation for Segmentation

Definition 1 (Segmentation Cell Aggregation)**.** We use fine-grained cell occupation to define our aggregation operator. Each coarse cell $c \in \Omega_{\text{coarse}}$ contains the set of fine sites $\Omega_{\text{fine}}|_c$. We define the aggregation operator for segmentation maps, transporting masks from resolution Ω_{fine} to Ω_{coarse} as,

$$w_c(Y^{\Omega_{\text{fine}}}) = \bigvee_{u \in \Omega_{\text{fine}}|_c} Y^{\Omega_{\text{fine}}}[u], \qquad c \in \Omega_{\text{coarse}}.$$

In the case of tumor segmentation, the logical OR operator ensures positive tumor cells are not missed during coarsening. By construction, we can see that the OR-aggregation operator is zero-preserving, as it is 0 when all inputs are 0, and monotonically increasing when any cell is nonzero. We take any off-the-shelf segmentation prediction model providing softmax-like scores per cell, and we can define the selection set $S_\lambda(X)$.

5.1 Brain Tumor Segmentation

The brain tumor dataset from the Medical Segmentation Decathlon (MSD) (Antonelli et al., 2022) consists of 484 cases with brain tumors of varying sizes, occupying at most 3.5% of the image volume. Each case is a 4D volume combining pre- and post-contrast T1-weighted, T2-weighted, and fluid attenuation inverse recovery (FLAIR) MRI scans.

We apply *Conformal Adaptation* to three segmentation architectures: U-Net (Ronneberger et al., 2015), Swin-UNETR (Hatamizadeh et al., 2021; He et al., 2023), and VISTA3D (He et al., 2025). We train each randomly initialized architecture three times at full resolution (*Full*) and at $4\times$-downsampled resolution (*Coarse*). The coarse model then determines the sparse patches used to

Table 1: Performance comparison on MSD Task01. Values are reported as mean ± standard deviation over three seeds. Columns 2-4 are the DSC for each model arrangement. Capacity-constrained models like U-Net see an improvement in performance due to additional weights. Larger scans require sliding window inference, increasing the FLOPs required for a single sample.

Model	Full	Coarse	Coarse + Refine ($\alpha = 0.1$)	Cell Miss Rate	FLOPs Speedup
U-Net	0.63 ± 0.03	0.51 ± 0.00	0.68 ± 0.01	0.10 ± 0.01	$2.70\times$
Swin-UNETR	0.74 ± 0.01	0.61 ± 0.01	0.70 ± 0.01	0.09 ± 0.00	$54.03\times$
VISTA3D	0.74 ± 0.00	0.62 ± 0.01	0.71 ± 0.00	0.08 ± 0.01	$82.10\times$

train the second-stage refiner (*Coarse + Refine*). The data processing, architecture hyperparameters, and training methods are detailed in Appendix C.1. Across architectures and devices, the method provides substantial FLOPs and wall-clock reductions while retaining similar DSC (Fig. 2). In CPU-only inference, VISTA3D achieves up to $72\times$ wall-clock speedup and an $82.1\times$ reduction in FLOPs (Table 1). We see super-linear improvement as the data can use fewer forward passes with sliding window inference in coarse resolution.

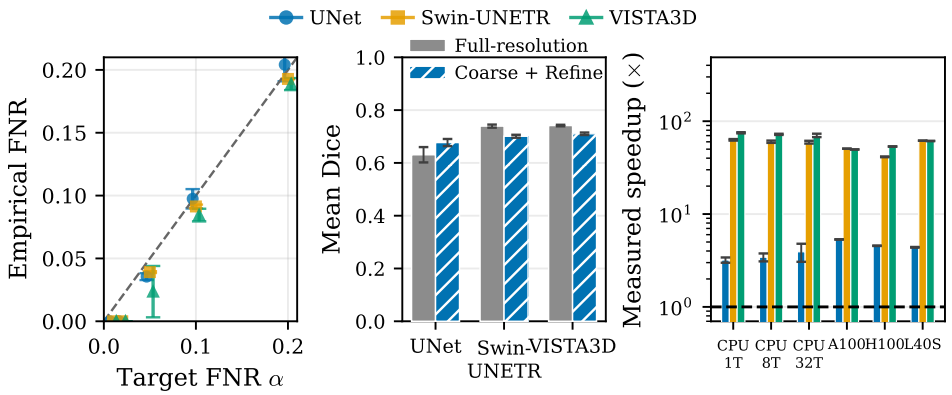

Figure 2: **Left:** Empirical verification of the CRC calibration quality. **Middle:** mean DSC of the Full baseline against the Coarse + Refine model at $\alpha = 0.1$. **Right:** Corresponding measured wall-clock speedup over the Full baseline on different devices.

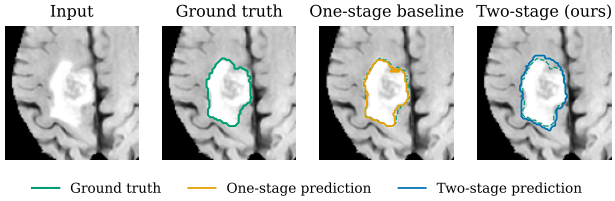

Figure 3: Predicted tumor boundaries on a held-out test scan (Swin-UNETR, $\alpha = 0.1$). The Coarse + Refine model tracks the boundary as closely as the Full baseline at a fraction of the FLOPs (Table 1).

5.2 Head and Neck Cancer and Metastatic Lymph Node Segmentation

Accurate segmentation of primary head and neck cancer (HNC) and affected lymph node volumes is essential for radiotherapy planning (RTP) as well as for monitoring tumor response to treatments. Computed tomography (CT) and 2-[18F]fluoro-2-deoxyglucose (FDG) PET are imaging modalities commonly used to localize and delineate tumors for RTP. We base our study on the multimodal HECKTOR 2026 grand challenge dataset (Saeed et al., 2025), which provides 782 FDG-PET and CT volumes spanning from the head to the pelvis. The size, structure, and sparsity of the HNC scans and tumors make the dataset an ideal candidate for *Conformal Adaptation*, as seen in Fig. 4. Each tumor spans significantly less than the total scanned area at the standard $1 \times 1 \times 1\,\mathrm{mm}^3$ spacing.

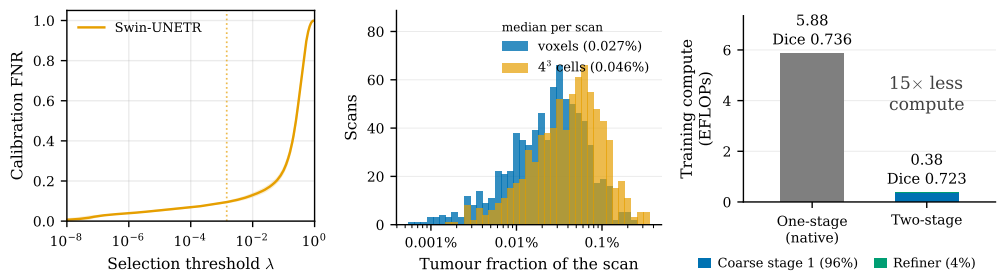

Figure 4: **(Left)**: False negative rates (FNR) can rapidly increase depending on chosen threshold, requiring a principled methods like CRC to choose appropriate λ (See Appendix C.3.**(Middle)** Tumor cells span a small fraction of the total scanned volume in most scans in the dataset. **(Right)** Training the coarse model on the aggregated samples and a separate refiner to convergence requires $15\times$ fewer FLOPs. All models are trained until validation loss stops decreasing with a set patience level.

We train and evaluate off-the-shelf models, including nnU-Net (Isensee et al., 2021), an industry standard that establishes a strong baseline, and the transformer-CNN-based Swin-UNETR (He et al., 2023). We use the standard CT and PET data pre-processing techniques, such as normalization to $[0, 1]$ and isotropic resampling. During training, we achieve a large speedup, as training the coarse model along with a sparse refiner takes $15\times$ fewer FLOPs than the full-resolution model (see Fig. 4). Full data-processing and training details are presented in Appendix C.2.

Once trained and calibrated, our Coarse + Refine model achieves results comparable to the Full model (DSC 0.72 vs. 0.74) while requiring $68\times$ fewer FLOPs (Fig. 5). Test-time augmentation with mirroring is used on all dense models. Furthermore, for lesion-detection metrics, where the predicted volume is used to detect lesions, our method achieves comparable recall and precision metrics but suffers from missing small adversarial GTVn cases (see Table 3).

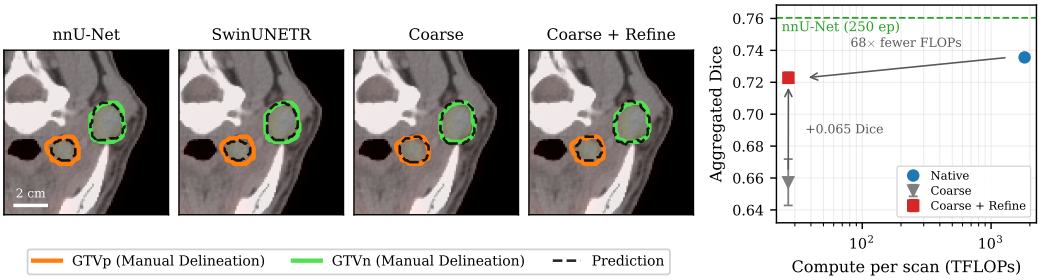

Figure 5: (Left) Visualizations of model segmentation with ground truth showing Coarse + Refine models with comparable contours. (Right) A coarse inference reduces per-scan compute cost while a sparse refinement improves prediction quality with little additional expense.

6 Case Study: Conformal Adaptation for Regression

We evaluate the generality of the cell weight mechanism defined in Eq. 8 for regression-type problems, where the prediction target is a continuous, non-negative signal (a rate, a count, a flux) rather than a discrete class label.

Definition 2 (Regression Cell Aggregation)**.** For a continuous, non-negative fine-resolution target $Y \in \mathcal{Y}^{\Omega_{\text{fine}}}$, we define the aggregation operator as the total signal mass inside the coarse cell, $w_c(Y^{\Omega_{\text{fine}}}) = \sum_{u \in \Omega_{\text{fine}}|_c} Y^{\Omega_{\text{fine}}}[u], \quad c \in \Omega_{\text{coarse}}$.

Summation is zero-preserving and, since $Y \geq 0$, monotonically non-decreasing in any added signal, satisfying the requirements of Eq. 8. The CRC loss of Eq. 9 then bounds, in expectation, the *fraction of total signal mass* that falls in unselected regions. Regression scores (e.g., a predicted upper

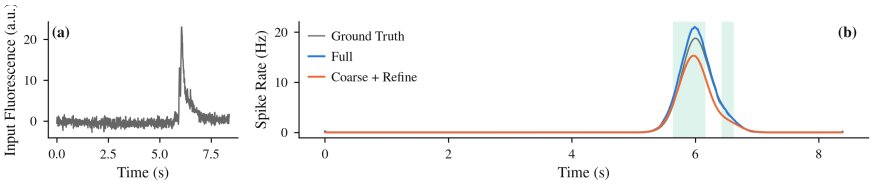

Figure 6: (a) Input signal. (b) Example test window: ground-truth spike rate overlaid with the Full prediction and the Coarse + Refine reconstruction, at a retained fraction of 8.4%.

quantile of cell mass) are not confined to $[0, 1]$, so we apply Eq. 6 after a fixed, strictly increasing map into $[0, 1]$. The map preserves score order, so the selected sets stay nested in λ as CRC requires, and we report $\hat{\lambda}$ by its equivalent raw-score cutoff.

6.1 CASCADE: 1D Calcium-Imaging Spike-Rate Inference

We use the CASCADE benchmark (Rupprecht et al., 2021) with 515 samples across 117 sources, in which the task is to convert fluorescence traces from calcium imaging into spike-rate curves. The coarse model $\mathcal{F}_{\Omega_{\text{coarse}}}$ block-averages the fluorescence trace by a factor of eight before applying the 1D U-Net of the full-resolution baseline $\mathcal{F}_{\Omega_{\text{fine}}}$. Note, we do not use the benchmark's decoder-free sliding-window regressor. CRC is calibrated on the coarse block counts, and the certified selection is refined with a count-predicting local refiner combined with a mass-preserving Gaussian reconstruction kernel, so bounding missed count mass at the coarse resolution also bounds missed mass in the reconstructed rate curve. An example input and output prediction is provided in Fig. 6. Table 6 reports the reference operating point. Across $\alpha \in \{0.01, 0.05, 0.1, 0.2\}$, CRC calibration is empirically valid (test risk $\leq \alpha$ at every level), and the reconstructed-curve MAE stays within 0.05–0.10 Hz of the full-resolution baseline (median MAE 0.250 Hz) while the retained fraction of coarse cells falls to 8.4% at α=0.1, an analytic FLOPs speedup of $4.77\times$. All details of the data pre-processing, model architecture, and calibration can be found in Appendix C.4.

6.2 Galaxy Flux Field Super-Resolution

Extending our experiments further, we also study a 2D galaxy-flux field task of predicting galaxy shape and flux from simulated surface-brightness maps. Simulated surface-brightness fields are foundational to observational astronomy. As the raw data provide no labeled truth, surveys characterize what their pipelines detect and miss by injecting sources of known flux into images and measuring recovery (Anbajagane et al., 2025; Team et al., 2026). The resulting completeness and photometric bias curves are used as inputs for correction terms on downstream detectors. Fields are generated with GalSim (Rowe et al., 2015), the community-standard image simulator validated against the precision requirements of Stage IV surveys. In practice surveys such as Rubin/LSST process on the order of 20 TB per night under a latency budget of seconds per visit (Ivezić et al., 2019) over fields that are overwhelmingly empty sky, making this setting well suited to our method.

We use a 2D U-Net as our base model. The Coarse + Refine pipeline is a coarse model sharing the base model's architecture but applied to an $8\times$-downsampled field, CRC certifying coarse cells by flux mass, and a sparse refiner predicting a non-negative spatial spread within each certified cell. The normalized refiner output is multiplied by the coarse net's own predicted median flux, making flux conservation between coarse and refined predictions exact by construction.

At the calibrated operating point (α=0.1, score cutoff 7.69), the pipeline retains only 2.2% of coarse cells on average, for an analytic FLOPs speedup of $61.6\times$ over the full-resolution baseline, at the cost of a higher per-pixel MAE (0.134 vs. 0.046 for the full-resolution baseline) with total-flux relative error of 5.6% vs. 2.3%, consistent with the certification. Figure 7 presents example predictions and measured wall-clock speedups across several devices, illustrating the method's potential for edge deployment.

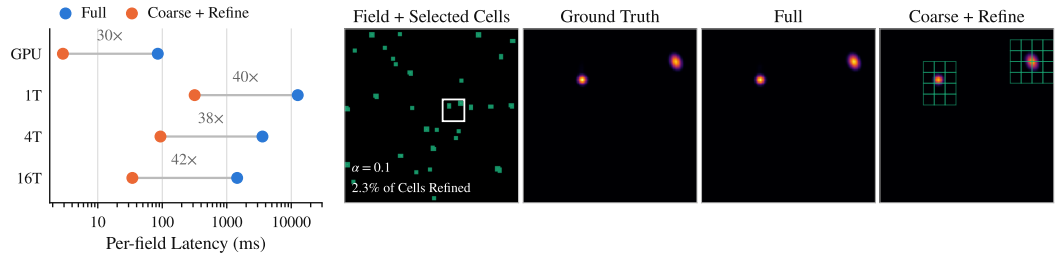

Figure 7: (Left) Achieved real-world wall-clock speedups on GPUs and CPU thread counts. (Right) Sample selection and comparison between the Full and Coarse + Refine models. The CRC-certified Coarse + Refine predictions have little discrepancy while costing a fraction of the compute and time.

7 Related Work

Static compression reduces inference cost by modifying the *model* independently of its input. Common approaches include pruning (Yu et al., 2017; Osawa et al., 2017), architectural simplification (Pang et al., 2023; 2025), static activation sparsification (Alanova et al., 2026), and quantization (Gupta et al., 2015; Sun et al., 2020; Liu et al., 2024; Chen et al., 2025). Zeng et al. (2026) propose token sparsification for efficient 3D segmentation, while static resolution scaling accelerates training by operating at fixed lower resolutions (Zhang et al., 2026). Unlike our method, these approaches evaluate error post hoc and provide no accuracy guarantees. Their gains are largely orthogonal to our input-adaptive sparsification.

Input-adaptive methods allocate compute based on the sample. Examples include token sparsification in vision transformers (Rao et al., 2021; Fayyaz et al., 2022; Uddin et al., 2025), region-wise token granularity selection (Havtorn et al., 2023), early exiting (He et al., 2021), and adaptive pixel refinement in PointRend (Kirillov et al., 2020). While these methods focus computation on informative regions, they rely on learned or heuristic selection mechanisms without guarantees on discarded information, making them susceptible to failure modes and adversarial attacks (Goyal et al., 2026). In contrast, our calibrated selection rule provably bounds discarded signal mass at risk level α. Conformal prediction has been extended to structured outputs through unified task formulations (Zhang et al., 2025), segmentation-specific spatial modeling (Viti et al., 2026; Angelopoulos et al., 2024), and input-conditional coverage (Luo & Zhou, 2025). These methods treat the predictor as fixed and use conformal prediction only to construct uncertainty sets, while we use the conformal certificate to guide computation. Wang et al. (2026), Schuster et al. (2022), Schuster et al. (2021), and Jazbec et al. (2024) apply distribution-free guarantees for adaptive inference and calibrate early-exit strategies to match full-model performance under compute constraints.

8 Discussion, Limitations, and Conclusion

Although we evaluate on uniform grids, the derivation in Sec. 3 is domain-agnostic and extends to non-uniform domains (e.g., graphs) whenever the refinement operators preserve the hierarchical partition. We use identical backbones for the native and coarse passes to isolate the gains due to conformalized selection itself. *Conformal Adaptation* decides *where* to compute, whereas quantization, pruning, sparse architectures, and token selection reduce the cost of *each* computation, so the two are orthogonal. Any stage may use quantized or pruned backbones (Liu et al., 2024; Yu et al., 2017), sparse architectures (Pang et al., 2023; 2025; Zeng et al., 2026), or token selection (Rao et al., 2021; Uddin et al., 2025) without changing the calibration procedure or the guarantee; whether their savings compound with ours is promising future work. We acknowledge that the guarantee has three limitations. First, it bounds the expected fraction of true signal in discarded cells, not the error of the refined prediction. Second, it holds marginally, so an individual scan may exceed α, and small lesions that are faint at coarse resolution may be missed. Third, it assumes exchangeability between calibration and test data and is therefore sensitive to covariate shift, e.g., across scanners or sites. Weighted CRC Tibshirani et al. (2019), high-probability (PAC-style) bounds (Angelopoulos et al., 2025), and per-lesion risk control are natural extensions. *Conformal Adaptation* offers a simple, calibrated way to spend compute only where it is required.

AI USE STATEMENT

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

Additionally, we used generative AI tools to edit code, improve readability, check grammar, and help identify relevant literature. We have reviewed all AI-assisted work. Author 1 has reviewed the implementation assisted with AI as well as the mathematical claims that were refined using AI. Authors 2 and 3 have further verified the notation and edits that were aided by generative AI, including checking mathematical claims, references, and manuscript edits. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

We take multiple steps to ensure all our results and analysis are reproducible. Our analysis uses publicly available data that can be freely downloaded from the cited sources. Although open to the public, the HECKTOR 2026 dataset requires signing a data use agreement and cannot be shared under its license terms. We detail the required data pre-processing in Appendices C.1, C.2, C.4, and C.5. The model definitions, training algorithms, and post-processing are also detailed in these appendices. Our main algorithmic contribution is detailed in Appendix A. To aid in reviewing mathematical correctness, the full proof is given in Appendix B. Finally, we provide all anonymized pre- and post-processing, training, and visualization code, along with all seeds used, as supplementary material.

REFERENCES

Shirin Alanova, Kristina Kazistova, Ekaterina Galaeva, Alina Kostromina, Vladimir Smirnov, Redko Dmitry, Alexey Dontsov, Maxim Zhelnin, Evgeny Burnaev, and Egor Shvetsov. Motivating Next-Gen Accelerators with Flexible N:M Activation Sparsity via Benchmarking Lightweight Post-Training Sparsification Approaches. In Yunyao Li, Georg Rehm, and Mei Tu (eds.), *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 6: Industry Track)*, pp. 242–258, San Diego, California, USA, July 2026. Association for Computational Linguistics. ISBN 9798891763944. doi: 10.18653/v1/2026.acl-industry.17. URL https://aclanthology.org/2026.acl-industry.17/.

Daniel Alcaraz-Ortiz, Juan Francisco Zapata-Pérez, and Juan Martinez-Alajarin. Cloud-to-edge deployment of optimized nnu-net for ischemic stroke lesion segmentation on resource-constrained embedded devices. *Sensors*, 26(11):3322, 2026.

Ebtihal J Alwadee, Xianfang Sun, Yipeng Qin, and Frank C Langbein. Latup-net: A lightweight 3d attention u-net with parallel convolutions for brain tumor segmentation. *Computers in biology and medicine*, 184:109353, 2025.

Dhayaa Anbajagane, M Tabbutt, J Beas-Gonzalez, B Yanny, S Everett, MR Becker, M Yamamoto, E Legnani, J De Vicente, K Bechtol, et al. Dark energy survey year 6 results: Synthetic-source injection across the full survey using balrog. *arXiv preprint arXiv:2501.05683*, 2025.

Anastasios Angelopoulos, Stephen Bates, Adam Fisch, Lihua Lei, and Tal Schuster. Conformal risk control. In *International conference on learning representations*, volume 2024, pp. 55198–55218, 2024.

Anastasios N Angelopoulos, Stephen Bates, Emmanuel J Candès, Michael I Jordan, and Lihua Lei. Learn then test: Calibrating predictive algorithms to achieve risk control. *The Annals of Applied Statistics*, 19(2):1641–1662, 2025.

Michela Antonelli, Annika Reinke, Spyridon Bakas, Keyvan Farahani, Annette Kopp-Schneider, Bennett A Landman, Geert Litjens, Bjoern Menze, Olaf Ronneberger, Ronald M Summers, et al. The medical segmentation decathlon. *Nature communications*, 13(1):4128, 2022.

Bastien Arcelin, Cyrille Doux, Eric Aubourg, and C'ecile Roucelle. Deblending galaxies with variational autoencoders: A joint multi-band, multi-instrument approach. *Monthly Notices of the Royal Astronomical Society*, 500(1):531–547, 2021.

Davis Blalock, Jose Javier Gonzalez Ortiz, Jonathan Frankle, and John Guttag. What is the state of neural network pruning? *Proceedings of machine learning and systems*, 2:129–146, 2020.

Mengzhao Chen, Wenqi Shao, Peng Xu, Jiahao Wang, Peng Gao, Kaipeng Zhang, and Ping Luo. EfficientQAT: Efficient Quantization-Aware Training for Large Language Models. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 10081–10100, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 9798891762510. doi: 10.18653/v1/2025.acl-long.498. URL https://aclanthology.org/2025.acl-long.498/.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. Qlora: Efficient finetuning of quantized llms. *Advances in neural information processing systems*, 36:10088–10115, 2023.

Mohsen Fayyaz, Soroush Abbasi Koohpayegani, Farnoush Rezaei Jafari, Sunando Sengupta, Hamid Reza Vaezi Joze, Eric Sommerlade, Hamed Pirsiavash, and Jürgen Gall. Adaptive Token Sampling for Efficient Vision Transformers. In Shai Avidan, Gabriel Brostow, Moustapha Cissé, Giovanni Maria Farinella, and Tal Hassner (eds.), *Computer Vision – ECCV 2022*, pp. 396–414, Cham, 2022. Springer Nature Switzerland. ISBN 9783031200830. doi: 10.1007/978-3-031-20083-0_24.

Jianping Gou, Baosheng Yu, Stephen J Maybank, and Dacheng Tao. Knowledge distillation: A survey. *International journal of computer vision*, 129(6):1789–1819, 2021.

Anadi Goyal, Nandish Chattopadhyay, Anupam Chattopadhyay, and Chandan Karfa. A Survey of Adversarial Efficiency Degradation for Vision Transformer by Exploiting Input-adaptive Optimization, August 2026. URL http://arxiv.org/abs/2608.05217. arXiv:2608.05217 [cs.CR].

Lei Guan, Shaofeng Zhang, and Yongle Chen. Energy-efficient deep learning training. *Advances in Energy Recovery and Efficiency Technologies*, pp. 1–18, 2024.

Suyog Gupta, Ankur Agrawal, Kailash Gopalakrishnan, and Pritish Narayanan. Deep Learning with Limited Numerical Precision. In *Proceedings of the 32nd International Conference on Machine Learning*, pp. 1737–1746. PMLR, June 2015. URL https://proceedings.mlr.press/v37/gupta15.html.

Ali Hatamizadeh, Vishwesh Nath, Yucheng Tang, Dong Yang, Holger R Roth, and Daguang Xu. Swin unetr: Swin transformers for semantic segmentation of brain tumors in mri images. In *International MICCAI brainlesion workshop*, pp. 272–284. Springer, 2021.

Jakob Drachmann Havtorn, Amélie Royer, Tijmen Blankevoort, and Babak Ehteshami Bejnordi. Msvit: Dynamic mixed-scale tokenization for vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 838–848, 2023.

Xuanli He, Iman Keivanloo, Yi Xu, Xiang He, Belinda Zeng, Santosh Rajagopalan, and Trishul Chilimbi. Magic Pyramid: Accelerating Inference with Early Exiting and Token Pruning, October 2021. URL http://arxiv.org/abs/2111.00230. arXiv:2111.00230 [cs.CL].

Yufan He, Vishwesh Nath, Dong Yang, Yucheng Tang, Andriy Myronenko, and Daguang Xu. Swinunetr-v2: Stronger swin transformers with stagewise convolutions for 3d medical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 416–426. Springer, 2023.

Yufan He, Pengfei Guo, Yucheng Tang, Andriy Myronenko, Vishwesh Nath, Ziyue Xu, Dong Yang, Can Zhao, Benjamin Simon, Mason Belue, et al. Vista3d: A unified segmentation foundation model for 3d medical imaging. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 20863–20873. IEEE, 2025.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

Fabian Isensee, Paul F Jaeger, Simon AA Kohl, Jens Petersen, and Klaus H Maier-Hein. nnu-net: a self-configuring method for deep learning-based biomedical image segmentation. *Nature methods*, 18(2):203–211, 2021.

Željko Ivezić, Steven M Kahn, J Anthony Tyson, Bob Abel, Emily Acosta, Robyn Allsman, David Alonso, Yusra AlSayyad, Scott F Anderson, John Andrew, et al. Lsst: from science drivers to reference design and anticipated data products. *The Astrophysical Journal*, 873(2):111, 2019.

Metod Jazbec, Alexander Timans, Tin Hadzi Veljkovic, Kaspar Sakmann, Dan Zhang, Christian Andersson Naesseth, and Eric T. Nalisnick. Fast yet Safe: Early-Exiting with Risk Control. In Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang (eds.), *Advances in Neural Information Processing Systems 37: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024. URL http://papers.nips.cc/paper_files/paper/2024/hash/ea5a63f7ddb82e58623693fd1f4933f7-Abstract-Conference.html.

Alexander Kirillov, Yuxin Wu, Kaiming He, and Ross Girshick. Pointrend: Image segmentation as rendering. In *2020 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pp. 9796–9805. IEEE, 2020.

Zechun Liu, Barlas Oguz, Changsheng Zhao, Ernie Chang, Pierre Stock, Yashar Mehdad, Yangyang Shi, Raghuraman Krishnamoorthi, and Vikas Chandra. LLM-QAT: Data-Free Quantization Aware Training for Large Language Models. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 467–484, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-acl.26. URL https://aclanthology.org/2024.findings-acl.26/.

Zhuang Liu, Mingjie Sun, Tinghui Zhou, Gao Huang, and Trevor Darrell. Rethinking the value of network pruning. *arXiv preprint arXiv:1810.05270*, 2018.

Rui Luo and Zhixin Zhou. Conditional conformal risk adaptation. *arXiv preprint arXiv:2504.07611*, 2025.

Vanessa Mehlin, Sigurd Schacht, and Carsten Lanquillon. Towards energy-efficient deep learning: An overview of energy-efficient approaches along the deep learning lifecycle. *arXiv preprint arXiv:2303.01980*, 2023.

Kazuki Osawa, Akira Sekiya, Hiroki Naganuma, and Rio Yokota. Accelerating Matrix Multiplication in Deep Learning by Using Low-Rank Approximation. In *2017 International Conference on High Performance Computing & Simulation (HPCS)*, pp. 186–192, July 2017. doi: 10.1109/HPCS.2017.37. URL https://ieeexplore.ieee.org/abstract/document/8035076.

Yan Pang, Jiaming Liang, Teng Huang, Hao Chen, Yunhao Li, Dan Li, Lin Huang, and Qiong Wang. Slim unetr: Scale hybrid transformers to efficient 3d medical image segmentation under limited computational resources. *IEEE Transactions on Medical Imaging*, 43(3):994–1005, 2023.

Yan Pang, Jiaming Liang, Junming Yan, Ying Hu, Hao Chen, and Qiong Wang. Slim unetrv2: 3d image segmentation for resource-limited medical portable devices. *IEEE Transactions on Medical Imaging*, 2025.

Yongming Rao, Wenliang Zhao, Benlin Liu, Jiwen Lu, Jie Zhou, and Cho-Jui Hsieh. DynamicViT: Efficient Vision Transformers with Dynamic Token Sparsification. In *Advances in Neural Information Processing Systems*, volume 34, pp. 13937–13949. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper/2021/hash/747d3443e319a22747fbb873e8b2f9f2-Abstract.html.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pp. 234–241. Springer, 2015.

Barnaby TP Rowe, Mike Jarvis, Rachel Mandelbaum, Gary M Bernstein, James Bosch, Melanie Simet, Joshua E Meyers, Tomasz Kacprzak, Reiko Nakajima, Joe Zuntz, et al. Galsim: The modular galaxy image simulation toolkit. *Astronomy and Computing*, 10:121–150, 2015.

Peter Rupprecht, Stefano Carta, Adrian Hoffmann, Mayumi Echizen, Antonin Blot, Alex C Kwan, Yang Dan, Sonja B Hofer, Kazuo Kitamura, Fritjof Helmchen, et al. A database and deep learning toolbox for noise-optimized, generalized spike inference from calcium imaging. *Nature neuroscience*, 24(9):1324–1337, 2021.

Numan Saeed, Salma Hassan, Shahad Hardan, Ahmed Aly, Darya Taratynova, Umair Nawaz, Ufaq Khan, Muhammad Ridzuan, Vincent Andrearczyk, Adrien Depeursinge, et al. A multimodal and multi-centric head and neck cancer dataset for segmentation, diagnosis and outcome prediction. *arXiv preprint arXiv:2509.00367*, 2025.

Tal Schuster, Adam Fisch, Tommi Jaakkola, and Regina Barzilay. Consistent Accelerated Inference via Confident Adaptive Transformers, September 2021. URL http://arxiv.org/abs/2104.08803. arXiv:2104.08803 [cs.CL].

Tal Schuster, Adam Fisch, Jai Gupta, Mostafa Dehghani, Dara Bahri, Vinh Tran, Yi Tay, and Donald Metzler. Confident adaptive language modeling. *Advances in Neural Information Processing Systems*, 35:17456–17472, 2022.

Roy Schwartz, Jesse Dodge, Noah A Smith, and Oren Etzioni. Green ai. *Communications of the ACM*, 63(12):54–63, 2020.

Xiao Sun, Naigang Wang, Chia-Yu Chen, Jiamin Ni, Ankur Agrawal, Xiaodong Cui, Swagath Venkataramani, Kaoutar El Maghraoui, Vijayalakshmi (Viji) Srinivasan, and Kailash Gopalakrishnan. Ultra-Low Precision 4-bit Training of Deep Neural Networks. In *Advances in Neural Information Processing Systems*, volume 33, pp. 1796–1807. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/hash/13b919438259814cd5be8cb45877d577-Abstract.html.

Vera C. Rubin Observatory Team, Tatiana Acero-Cuellar, Emily Acosta, Christina L Adair, Prakruth Adari, Jennifer K Adelman-McCarthy, Anastasia Alexov, Russ Allbery, Robyn Allsman, Yusra AlSayyad, et al. The vera c. rubin observatory data preview 1. *The Astronomical Journal*, 171(6): 360, 2026.

Ryan J Tibshirani, Rina Foygel Barber, Emmanuel Candes, and Aaditya Ramdas. Conformal prediction under covariate shift. *Advances in neural information processing systems*, 32, 2019.

Mohammad Helal Uddin, Liam Seymour, and Sabur Baidya. HEART-VIT: Hessian-Guided Efficient Dynamic Attention and Token Pruning in Vision Transformer, December 2025. URL http://arxiv.org/abs/2512.20120. arXiv:2512.20120 [cs.CV].

Bruno Viti, Elias Karabelas, and Martin Holler. Consign: Conformal segmentation informed by spatial groupings via decomposition. In *International Conference on Learning Representations*, volume 2026, pp. 43182–43204, 2026.

Xi Wang, Anushri Suresh, Alvin Zhang, Rishi More, William Jurayj, Benjamin Van Durme, Mehrdad Farajtabar, Daniel Khashabi, and Eric Nalisnick. Conformal thinking: Risk control for reasoning on a compute budget. *arXiv preprint arXiv:2602.03814*, 2026.

Xiyu Yu, Tongliang Liu, Xinchao Wang, and Dacheng Tao. On Compressing Deep Models by Low Rank and Sparse Decomposition. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 67–76, July 2017. doi: 10.1109/CVPR.2017.15. URL https://ieeexplore.ieee.org/abstract/document/8099498.

Sen Zeng, Hong Zhou, Zheng Zhu, and Yang Liu. Tokenseg: Efficient 3d medical image segmentation via hierarchical visual token compression. *arXiv preprint arXiv:2601.04519*, 2026.

Botong Zhang, Shuo Li, and Osbert Bastani. Conformal structured prediction. In *International Conference on Learning Representations*, volume 2025, pp. 8563–8583, 2025.

Jiaqi Zhang, Juntuo Wang, Zhixin Sun, John Zou, and Randall Balestriero. Fastdinov2: Frequency based curriculum learning improves robustness and training speed. *Advances in Neural Information Processing Systems*, 38:34154–34177, 2026.

A Conformal Adaptation: Training, Calibration, Inference

A.1 Training

Given an admissible operator A, a coarse predictor model $\mathcal{F}_{\Omega_{\text{coarse}}}$ is trained together with the refinement operator $\mathcal{G}$. In general, training the two models is a three-step process:

1. **Train the coarse model**:
   - Downsample the input to the coarse partitioning, and aggregate the labels into coarse cell weights with an admissible operator A
2. **Extract sparse training data**:
   - Rather than the complete fine-grained samples, the sparse refiner is trained on small patches (that will be selected by the coarse model)
   - The sparse models can be trained either with 'simulated' sparse data that is selected by an oracle or it can be selected using the now trained coarse model $\mathcal{F}_{\Omega_{\text{coarse}}}$ at a training selection level λ_{train}. Setting $\lambda_{\text{train}} = \hat{\lambda}$ (Algorithm 2) trains the refiner on the certified distribution it will see at inference, at the cost of one refiner per calibrated α
3. **Train the refiner model**
   - The refiner model is trained on the extracted sparse patch-level data.

Throughout this section we use $\mathcal{G}$ for $\mathcal{G}_{\Omega_{\text{coarse}} \to \Omega_{\text{fine}}}$ for conciseness.

Algorithm 1 Training the Coarse and Selective Refinement Models

Require: Training dataset $\mathcal{D}_{\text{train}} = \{(X_j^{\Omega_{\text{fine}}}, Y_j^{\Omega_{\text{fine}}})\}_{j=1}^N$, downsampling operator $\text{Downsample}(\cdot)$, admissible aggregation operator A, scoring function $s(\cdot)$, training selection level $\lambda_{\text{train}} \in [0, 1]$.
Ensure: Trained coarse model $\mathcal{F}_{\Omega_{\text{coarse}}}$, trained refinement model $\mathcal{G}$.

// Step 1: Coarse Data Preparation & Training

Construct downsampled coarse dataset:
$\mathcal{D}_{\Omega_{\text{coarse}}} \leftarrow \left\{\left(\text{Downsample}(X_j^{\Omega_{\text{fine}}}),\, w(Y_j^{\Omega_{\text{fine}}})\right)\right\}_{j=1}^N$, where $w(Y) = (w_c(Y))_{c \in \Omega_{\text{coarse}}}$
Train $\mathcal{F}_{\Omega_{\text{coarse}}} \colon \mathcal{X}^{\Omega_{\text{coarse}}} \to \mathcal{Y}^{\Omega_{\text{coarse}}}$ by minimizing standard empirical risk on $\mathcal{D}_{\Omega_{\text{coarse}}}$

// Step 2: Refinement Data Extraction via Coarse Selection

Initialize refinement dataset $\mathcal{D}_{\text{refine}} \leftarrow \emptyset$
for each sample $(X_j^{\Omega_{\text{fine}}}, Y_j^{\Omega_{\text{fine}}}) \in \mathcal{D}_{\text{train}}$ **do**
    Compute coarse prediction: $\hat{O}_j^{\Omega_{\text{coarse}}} \leftarrow \mathcal{F}_{\Omega_{\text{coarse}}}(X_j^{\Omega_{\text{coarse}}})$
    for each coarse cell $c_k \in \Omega_{\text{coarse}}$ **do**
        Select cell by Eq. 6: $\sigma_j[c_k] \leftarrow \mathbb{I}[s(X_j^{\Omega_{\text{coarse}}})[c_k] \geq 1 - \lambda_{\text{train}}]$
        (oracle variant: $\sigma_j[c_k] \leftarrow \mathbb{I}[w_{c_k}(Y_j) > 0]$)
        if $\sigma_j[c_k] = 1$ **then**
            Extract fine patch $x_{j,k}^{\Omega_{\text{fine}}} \leftarrow X_j^{\Omega_{\text{fine}}}\big|_{c_k}$
            Extract ground-truth fine label $y_{j,k}^{\Omega_{\text{fine}}} \leftarrow Y_j^{\Omega_{\text{fine}}}\big|_{c_k}$
            $\mathcal{D}_{\text{refine}} \leftarrow \mathcal{D}_{\text{refine}} \cup \left\{\left((x_{j,k}^{\Omega_{\text{fine}}}, \hat{O}_j^{\Omega_{\text{coarse}}}[c_k]),\, y_{j,k}^{\Omega_{\text{fine}}}\right)\right\}$
        end if
    end for
end for

// Step 3: Train Refinement Model

Train local refinement operator $\mathcal{G} \colon \mathcal{X}^{\Omega_{\text{fine}}}\big|_{c_k} \times \mathcal{Y}^{\Omega_{\text{coarse}}}[c_k] \to \mathcal{Y}^{\Omega_{\text{fine}}}\big|_{c_k}$ on $\mathcal{D}_{\text{refine}}$
return $\mathcal{F}_{\Omega_{\text{coarse}}}, \mathcal{G}$

A.2 Calibration

The CRC algorithm is run on the low-resolution partitioning as seen in Algorithm 2. It uses the missed-signal-mass loss of Eq. 9, which depends only on the coarse selection and the cell weights, so the refiner $\mathcal{G}$ plays no part in calibration.

Algorithm 2 Conformal Risk Calibration (Threshold Selection)

Require: Calibration set $\mathcal{D}_{\text{cal}} = \{(X_i^{\Omega_{\text{fine}}}, Y_i^{\Omega_{\text{fine}}})\}_{i=1}^n$, trained coarse model $\mathcal{F}_{\Omega_{\text{coarse}}}$, downsampling operator $\mathrm{Downsample}(\cdot)$, admissible aggregation operator A, scoring function $s(\cdot)$, risk level $\alpha \in (0,1)$, grid of candidates $\Lambda \subseteq [0,1]$ with $1 \in \Lambda$.

Ensure: Calibrated selection threshold $\hat{\lambda}$.

for $i = 1$ **to** n **do**
  $X_i^{\Omega_{\text{coarse}}} \leftarrow \mathrm{Downsample}(X_i^{\Omega_{\text{fine}}})$
  $\hat{O}_i^{\Omega_{\text{coarse}}} \leftarrow \mathcal{F}_{\Omega_{\text{coarse}}}(X_i^{\Omega_{\text{coarse}}})$
  Cell weights (Eq. 8): $w_{c_k}(Y_i) \leftarrow A\big(Y_i^{\Omega_{\text{fine}}}\big|_{c_k}\big)$ for each $c_k \in \Omega_{\text{coarse}}$
end for
for each candidate threshold $\lambda \in \Lambda$ **do**
  for $i = 1$ **to** n **do**
    Form selection vector $\sigma_\lambda \in \{0,1\}^{\Omega_{\text{coarse}}}$:
    $\sigma_\lambda[c_k] \leftarrow \mathbb{I}[s(X_i^{\Omega_{\text{coarse}}})[c_k] \geq 1 - \lambda]$
    Missed signal mass (Eq. 9, $0/0 := 0$):
    $L_i(\lambda) \leftarrow \sum_{k:\,\sigma_\lambda[c_k]=0} w_{c_k}(Y_i) \,/\, \sum_k w_{c_k}(Y_i)$
  end for
  Compute empirical risk: $\widehat{R}_n(\lambda) \leftarrow \frac{1}{n}\sum_{i=1}^n L_i(\lambda)$
end for
Select threshold guaranteeing controlled risk ($L \in [0,1]$, so $B = 1$):
$\hat{\lambda} \leftarrow \inf\left\{\lambda \in \Lambda : \frac{1}{n+1}\big(n\,\widehat{R}_n(\lambda) + 1\big) \leq \alpha\right\}$
return $\hat{\lambda}$

A.3 Inference

Once trained and calibrated, the coarse and refiner models $\mathcal{F}_{\Omega_{\text{coarse}}}$ and $\mathcal{G}$ can be used for inference along with the calibrated threshold $\hat{\lambda}$. The inference algorithm is detailed in Algorithm 3.

Algorithm 3 Conformal Adaptation Inference

Require: Test input $X^{\Omega_{\text{fine}}}$, models $\mathcal{F}_{\Omega_{\text{coarse}}}, \mathcal{G}$, calibrated threshold $\hat{\lambda}$, downsampling operator $\mathrm{Downsample}(\cdot)$, scoring function $s(\cdot)$.

Ensure: Sparsely refined prediction $\hat{Y}^{\Omega_{\text{fine}}}$.

Downsample coarse input: $X^{\Omega_{\text{coarse}}} \leftarrow \mathrm{Downsample}(X^{\Omega_{\text{fine}}})$
Run coarse model: $\hat{O}^{\Omega_{\text{coarse}}} \leftarrow \mathcal{F}_{\Omega_{\text{coarse}}}(X^{\Omega_{\text{coarse}}})$
Initialize fine canvas: $\hat{Y}^{\Omega_{\text{fine}}} \leftarrow \mathrm{Upsample}(\hat{O}^{\Omega_{\text{coarse}}})$
Determine sparse refinement set:
$S \leftarrow \left\{c_k \in \Omega_{\text{coarse}} : s(X^{\Omega_{\text{coarse}}})[c_k] \geq 1 - \hat{\lambda}\right\}$
for each coarse cell $c_k \in S$ **do** ▷ Execute refinement strictly on sparse locations
  $\hat{Y}^{\Omega_{\text{fine}}}\big|_{c_k} \leftarrow \mathcal{G}\left(X^{\Omega_{\text{fine}}}\big|_{c_k}, \hat{O}^{\Omega_{\text{coarse}}}[c_k]\right)$
end for
return $\hat{Y}^{\Omega_{\text{fine}}}$

B Derivation of Guarantees

Lemma 1. *For every (X, Y), the loss of Eq. 9 satisfies $L(\lambda; X, Y) \in [0, 1]$, is non-increasing and right-continuous in $\lambda \in [0, 1]$, and satisfies $L(1; X, Y) = 0$.*

Proof. Non-negativity of A makes every $w_c(Y) \geq 0$; the numerator sums a subset of the terms of the denominator, so $L \in [0, 1]$ (with $0/0 := 0$). The denominator does not depend on λ. If $\lambda_1 \leq \lambda_2$ then $S_{\lambda_1}(X) \subseteq S_{\lambda_2}(X)$ by Eq. 6, so passing from λ_1 to λ_2 deletes non-negative terms from the numerator and $L(\lambda_2; X, Y) \leq L(\lambda_1; X, Y)$. Since Ω_{coarse} is finite, the score field s takes finitely many values, and the closed threshold in Eq. 6 makes $\lambda \mapsto S_\lambda(X)$ right-continuous; hence so is L. At $\lambda = 1$ every cell satisfies $s(X^{\Omega_{\text{coarse}}})[c] \geq 0$, so $S_1(X) = \Omega_{\text{coarse}}$, the numerator is an empty sum, and $L(1; X, Y) = 0$. □

Lemma 2. *Under Conditions 1 and 2, for every $Y \in \mathcal{Y}^{\Omega_{fine}}$ and every $c \in \Omega_{coarse}$,*

$$w_c(Y) > 0 \iff c \in \mathcal{O}(Y).$$

Consequently, for every $\lambda \in [0, 1]$,

$$L(\lambda; X, Y) = 0 \iff \mathcal{O}(Y) \subseteq S_\lambda(X),$$

and the right-hand event depends on neither A nor λ except through $S_\lambda(X)$. In particular all admissible operators induce losses with the same zero set.

Proof. If $c \notin \mathcal{O}(Y)$ then $Y^{\Omega_{\text{fine}}}|_c \equiv \varnothing$ and Condition 1 gives $w_c(Y) = 0$. If $c \in \mathcal{O}(Y)$ then some fine site of c carries a non-null label and Condition 2 gives $w_c(Y) > 0$. This proves the equivalence.

For the second claim, suppose first that $\mathcal{O}(Y) = \emptyset$. Then all weights vanish, the convention $0/0 := 0$ gives $L = 0$, and the inclusion holds trivially. Otherwise the denominator of Eq. 9 is strictly positive, so $L(\lambda; X, Y) = 0$ iff $\sum_{c \notin S_\lambda(X)} w_c(Y) = 0$, iff $w_c(Y) = 0$ for every $c \notin S_\lambda(X)$ (all terms being non-negative), iff no occupied cell lies outside $S_\lambda(X)$ by the equivalence just proved. □

Lemma 3. *Let $\sigma \in \{0, 1\}^{\Omega_{coarse}}$ with selected set S, and write $\hat{Y}_\sigma^{\Omega_{fine}} := \mathcal{G}_{\Omega_{coarse} \to \Omega_{fine}}(X^{\Omega_{fine}}, \hat{O}^{\Omega_{coarse}}, \sigma)$. Then*

$$\hat{Y}_\sigma^{\Omega_{fine}}\big|_c = \hat{Y}_{\mathbf{1}}^{\Omega_{fine}}\big|_c \quad \text{for every } c \in S, \qquad \hat{Y}_\sigma^{\Omega_{fine}}\big|_c = \hat{Y}_{\mathbf{0}}^{\Omega_{fine}}\big|_c \quad \text{for every } c \notin S,$$

where $\hat{Y}_{\mathbf{0}}^{\Omega_{fine}} = \mathrm{Upsample}(\hat{O}^{\Omega_{coarse}})$. Consequently, for any cellwise functional ϕ (i.e. $\phi(Y)[c]$ depends only on $Y|_c$),

$$\phi\big(\hat{Y}_\sigma^{\Omega_{fine}}\big)[c] = \sigma[c]\, \phi\big(\hat{Y}_{\mathbf{1}}^{\Omega_{fine}}\big)[c] + \big(1 - \sigma[c]\big)\, \phi\big(\hat{Y}_{\mathbf{0}}^{\Omega_{fine}}\big)[c],$$

and in particular

$$\mathcal{O}\big(\hat{Y}_\sigma^{\Omega_{fine}}\big) = \big(S \cap \mathcal{O}(\hat{Y}_{\mathbf{1}}^{\Omega_{fine}})\big) \cup \big(\mathcal{O}(\hat{Y}_{\mathbf{0}}^{\Omega_{fine}}) \setminus S\big).$$

Sparse refinement is a cellwise selection between two fields, neither of which depends on σ.

Proof. By Eq. 5, the output of $\mathcal{G}_{\Omega_{\text{coarse}} \to \Omega_{\text{fine}}}$ on a selected cell c is $\mathcal{G}_c(X^{\Omega_{\text{fine}}}|_c, \hat{O}^{\Omega_{\text{coarse}}}[c])$. Its arguments do not depend on σ, so it coincides with the value on c when $\sigma = \mathbf{1}$. On an unselected cell, c is never overwritten, so the output retains the initial canvas $\mathrm{Upsample}(\hat{O}^{\Omega_{\text{coarse}}})|_c$ (Algorithm 3, line 3). This canvas is computed from $\hat{O}^{\Omega_{\text{coarse}}}$ alone, so it is independent of σ and equals $\hat{Y}_{\mathbf{0}}^{\Omega_{\text{fine}}}|_c$. The formula for a cellwise ϕ follows by evaluating ϕ on each cell in the two cases. The occupancy identity is the instance $\phi = \mathbb{I}[c \in \mathcal{O}(\cdot)]$. □

Remark 1. If Upsample maps every cell to $\varnothing$ (e.g. a background fill in segmentation), then $\mathcal{O}(\hat{Y}_{\mathbf{0}}^{\Omega_{\text{fine}}}) = \emptyset$ and the lemma reduces to $\mathcal{O}(\hat{Y}_\sigma^{\Omega_{\text{fine}}}) = S \cap \mathcal{O}(\hat{Y}_{\mathbf{1}}^{\Omega_{\text{fine}}})$. For regression, Upsample may be any fixed interpolant (nearest, bilinear, etc.). It need not be cell-local; it only needs to be computed from $\hat{O}^{\Omega_{\text{coarse}}}$ independently of S.

Theorem 3.1. *Let A be admissible, let $(X_1, Y_1), \ldots, (X_n, Y_n), (X, Y)$ be exchangeable, where $(X, Y) = (X_{n+1}, Y_{n+1})$ is the test pair, and let $\hat{\lambda}$ be the CRC threshold obtained by calibrating Eq. 9 at level α on the first n pairs. Write $D = \Omega_{coarse} \setminus S_{\hat{\lambda}}(X)$ for the discarded cells. Then:*

(i) Almost surely, $\hat{Y}^{\Omega_{fine}}_{\sigma_{\hat{\lambda}}(X)}$ and $\hat{Y}^{\Omega_{fine}}_{\mathbf{1}}$ agree at every fine site of every cell in $S_{\hat{\lambda}}(X)$; the two pipelines can differ only on $\bigcup_{c\in D}\Omega_{fine}|_c$.

(ii) The true signal mass carried by that region is controlled:

$$\mathbb{E}\left[\frac{\sum_{c\in D} w_c(Y)}{\sum_{c\in\Omega_{coarse}} w_c(Y)}\right] \;\le\; \alpha.$$

(iii) With probability at least $1-\alpha/t$ for any $t\in(\alpha,1]$, the discarded region carries at most a t-fraction of the signal mass; and the event that it carries none is exactly $\{\mathcal{O}(Y)\subseteq S_{\hat{\lambda}}(X)\}$, for every admissible A.

Proof. For (ii), we see that Lemma 1 verifies the hypotheses of the CRC guarantee (Eq. 1) for the family $\{L(\lambda;\cdot,\cdot)\}_{\lambda\in[0,1]}$, which under exchangeability yields $\mathbb{E}[L(\hat{\lambda};X,Y)]\le\alpha$. Claim (i) is Lemma 3 applied with $\sigma=\sigma_{\hat{\lambda}}(X)$, and it identifies $\bigcup_{c\in D}\Omega_{\text{fine}}|_c$ as the exact support of the discrepancy between the sparse and dense pipelines, which is the region whose mass (ii) bounds. The tail bound in (iii) is Markov's inequality applied to the non-negative variable $L(\hat{\lambda};X,Y)$, and the characterization of its zero set is Lemma 2. The discarded mass fraction is precisely the loss at the calibrated threshold:

$$L(\hat{\lambda};X,Y)=\frac{\sum_{c\in D} w_c(Y)}{\sum_{c\in\Omega_{\text{coarse}}} w_c(Y)},\qquad D=\Omega_{\text{coarse}}\setminus S_{\hat{\lambda}}(X),$$

with $0/0:=0$. By Lemma 1, $L(\hat{\lambda};X,Y)\ge 0$, and by (ii), $\mathbb{E}[L(\hat{\lambda};X,Y)]\le\alpha$. For $t\in(0,1]$, Markov's inequality gives

$$\Pr\big(L(\hat{\lambda};X,Y)\ge t\big)\;\le\;\frac{\mathbb{E}[L(\hat{\lambda};X,Y)]}{t}\;\le\;\frac{\alpha}{t},$$

and therefore

$$\Pr\big(L(\hat{\lambda};X,Y)\le t\big)\;\ge\;\Pr\big(L(\hat{\lambda};X,Y)<t\big)\;=\;1-\Pr\big(L(\hat{\lambda};X,Y)\ge t\big)\;\ge\;1-\frac{\alpha}{t}.$$

That is, with probability at least $1-\alpha/t$, the cells in D carry at most a t-fraction of the total signal mass $\sum_{c\in\Omega_{\text{coarse}}} w_c(Y)$.

For the zero set, Lemma 2 holds for every $\lambda\in[0,1]$, so it applies to each realisation of $\hat{\lambda}$:

$$L(\hat{\lambda};X,Y)=0\iff\mathcal{O}(Y)\subseteq S_{\hat{\lambda}}(X).$$

This event depends on A only through $S_{\hat{\lambda}}(X)$. Its form is therefore the same for every admissible A, although the event itself can still vary with A through $\hat{\lambda}$. □

C Experiment Details

C.1 MSD Brain Tumor Segmentation

The MSD experiment's data, model, and calibration details are presented in this section.

C.1.1 Dataset Details

We use the MSD *Task01: Brain Tumour* dataset (Antonelli et al., 2022), comprising 484 multimodal brain MRI scans, each with four co-registered $240\times240\times155$-voxel sequences (FLAIR, T1-weighted, contrast- enhanced T1-weighted, and T2-weighted) and a voxel-wise annotation into three positive tumor subregions (edema, non-enhancing tumor, and enhancing tumor) plus background. The official MSD test split ships without public labels, so, following standard practice, we instead use MONAI's deterministic 80/20 split of the released *training* set: the resulting 96-scan validation subset is held out entirely and used as our test set, while the remaining 388-scan training subset is further split into a 320-scan model-fitting set and a 68-scan (17.5%) calibration set, sampled with a fixed random seed independent of the 80/20 split. The calibration set is held out of all first-stage model training; it is used exclusively to calibrate the conformal risk-control threshold $\hat{\lambda}$.

C.1.2 Data Preprocessing

Each modality is independently normalized by clipping its non-zero voxels to their own 99.5th percentile and rescaling to $[0, 1]$; background (zero) voxels are left untouched. Volumes are then spatially padded to the nearest multiple of the coarse downsampling factor $r = 4$ on every axis, so that the coarse pass's $4 \times 4 \times 4$ average/max pooling divides the volume exactly. The coarse (low-resolution) image is obtained by trilinear downsampling of the padded volume by r; the corresponding coarse label is obtained by non-overlapping max-pooling over the same $4 \times 4 \times 4$ blocks, so that a positive (tumor) voxel is never averaged away at low resolution. For training the first-stage (coarse and full-resolution) segmentation models, we sample fixed $64 \times 64 \times 32$-voxel crops per scan, biased 1:1 toward foreground-containing regions, followed by random flips along each of the three spatial axes ($p = 0.5$ each) and a random 90° in-plane rotation ($p = 0.5$), then pad back to the fixed crop size. Evaluation, calibration, and the refinement stage instead operate on the full padded volume via sliding-window inference with the same $64 \times 64 \times 32$ window.

C.1.3 Model Architecture

We compare three segmentation architectures, all trained from scratch (no pretrained weights) and used identically as the coarse-pass model and as the full-resolution baseline – only the spatial resolution of the input differs between the two, isolating the effect of input resolution from any change in model capacity.

- **U-Net** (Ronneberger et al., 2015): a 3-level 3D U-Net with channel widths $(16, 32, 64)$ and stride-2 downsampling at each level.
- **Swin-UNETR** (Hatamizadeh et al., 2021; He et al., 2023): a Swin Transformer encoder with a convolutional U-Net-style decoder, at its default configuration.
- **VISTA3D** (He et al., 2025): a segmentation foundation-model architecture composed of a SegResNet-based image encoder (downsampling blocks $(1, 2, 2, 4, 4)$, instance normalization, 48-channel embedding) and a class-mapping classification head. We run VISTA3D in its automatic (class-vector) segmentation mode only, without any point prompts, so that it can be invoked identically to the other two architectures.

The region-refinement stage uses a separate, lightweight two-branch 3D convolutional network, common to all three first-stage architectures. Each selected coarse cell is a $4 \times 4 \times 4$ block; adding a one-voxel halo on every side gives a $6 \times 6 \times 6$ high-resolution window. One branch encodes this high-resolution image window and the other encodes the upsampled coarse-model mask over the same window (two $3 \times 3 \times 3$ convolutions, 16 channels, GELU activations); the two branches are fused by addition and passed through two further convolutions before a $1 \times 1 \times 1$ output head. The refiner's full $6 \times 6 \times 6$ output is center-cropped to the central $4 \times 4 \times 4$ block, which is what is scored against the ground truth and scattered back into the full-resolution prediction; the halo voxels serve only as context for the convolutions.

C.1.4 Training and Calibration Details

Each of the three first-stage architectures is trained at both full and $4\times$-downsampled resolution, for three random seeds, with Adam (learning rate 10^{-4}, batch size 8, up to 100 epochs), minimizing an equally weighted sum of Dice loss and cross-entropy. A single region-refinement network, shared by all architectures and seeds, is trained for one epoch with Adam (learning rate 10^{-3}, batch size 256), minimizing cross-entropy on the central $4 \times 4 \times 4$ block only. Its training windows are the cells the coarse model selects at its $\alpha = 0.1$ calibrated threshold on the 388-scan training subset, with that model's softmax as the coarse-mask input. CRC is calibrated on each architecture's coarse-pass foreground probability, per training seed, using the missed-positive-cell-fraction loss of Eq. 9 on the 68-scan calibration set described above; the calibrated threshold $\hat{\lambda}$, the resulting retained cell fraction, and downstream DSC and FLOPs are then evaluated on the fully held-out 96-scan test set.

C.1.5 Qualitative Examples

Fig. 3 in the main text (Sec. 5) compares predicted tumor boundaries directly; Fig. 8 here shows, for the same three test scans, how the region CRC sends to refinement (Eq. 1) shrinks toward the tumor

as the target miss rate α grows, together with the resulting measured per-scan cell miss rate. Thus, the coverage guarantee behind Table 1's cell miss rate column is visible at the level of a single slice, not only as an aggregate statistic.

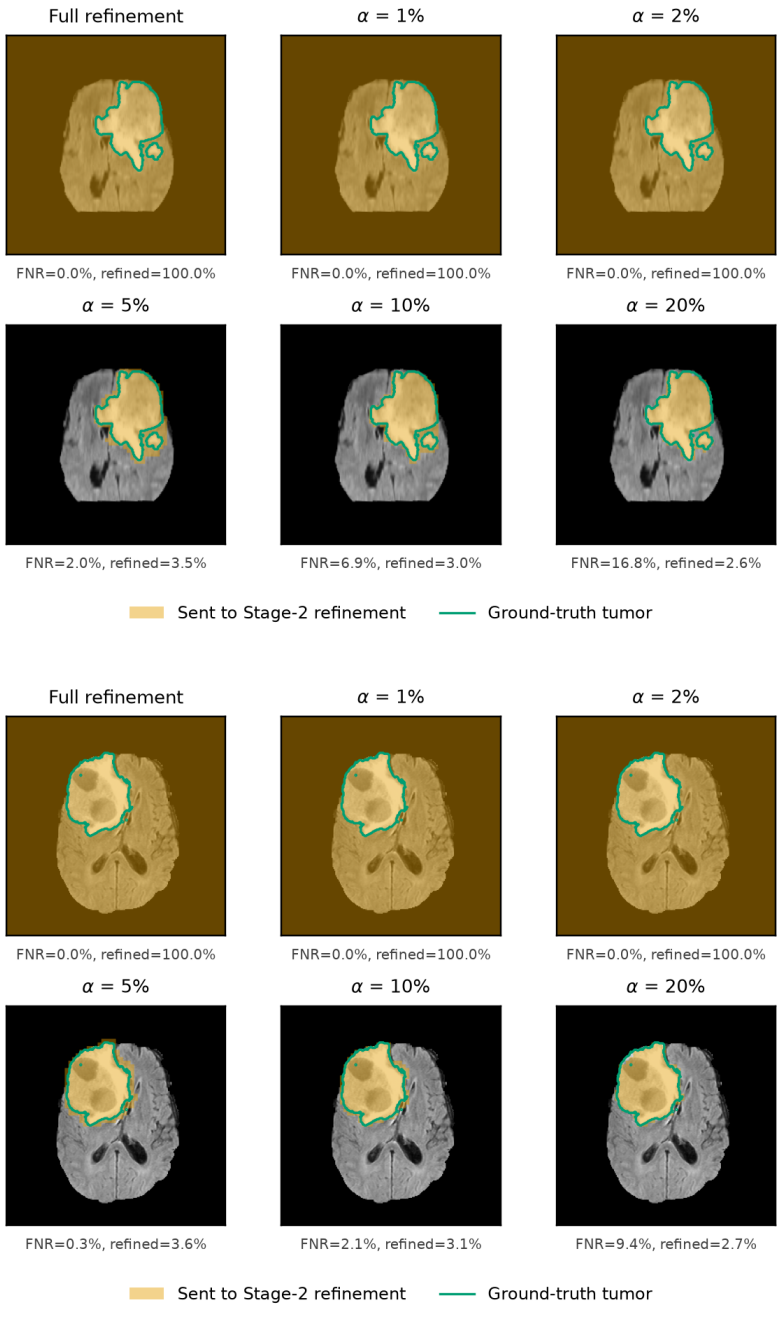

Figure 8: Selected (refined) region against the calibrated threshold for two held-out test scans (Swin-UNETR, seed 0). "Full refinement" is the reference at foreground-probability threshold 0, i.e. $\lambda = 1$ in Eq. 6 (every foreground-probable cell refined); at $\alpha = 1\%$ and 2%, CRC's calibrated threshold is itself 0 ($\hat{\lambda} = 1$) for this architecture/seed. As α relaxes, the selected region shrinks toward the tumor boundary and the measured cell miss rate rises toward its target.

C.2 HECKTOR 2026 Head and Neck Tumor Segmentation

C.2.1 Data Preprocessing

There are three NIfTI volumes per case: the original CT, the FDG-PET, and the segmentation label. The labels are included as 0 for background, 1 for GTVp (primary), and 2 for GTVn (node).

The 782 cases are split across 531 train, 125 validation, and 126 test samples. The 125-case validation split is the **CRC calibration set**, not a model-selection set. Note that validation splits are obtained randomly from the training set for the Coarse and Refine models.

C.2.2 Results

Full test-set results for the HNC study are provided in Tables 2 and 3. It is important to note that the there is a drop in accuracy especially for GTVn predictions when using the Coarse + Refine mechanism. This is particularly due to the fact that there small lesions that are being missed by the coarse model and being enabled by the CRC loss on cell-level misses. We acknowledge this as a negative result for our method as it is a particularly adversarial task given many small lesions in HNCs. A custom designed coarse model and a lesion-based CRC loss can potentially significantly improve our current lesion-based results as highlighted before, is a significant area of future work that can be pursued.

Table 2: DSC and FLOPs speedup across models. All rows except nnU-Net are Swin-UNETR. Mean DSC is the unweighted average of GTVp and GTVn; speedup is the ratio of Full-model FLOPs to pipeline FLOPs.

System	GTVp	GTVn	Mean	Speedup
nnU-Net (250 ep)	**0.7938**	**0.7271**	**0.7604**	–
Full	0.7755	0.6957	0.7356	1.0×
Coarse	0.7487 ± 0.0117	0.5660 ± 0.0307	0.6573 ± 0.0145	68.3×
Coarse + Refine (α=0.1)	0.7810 ± 0.0005	0.6646 ± 0.0068	0.7228 ± 0.0034	68.1×

Table 3: Lesion-specific results for our HNC models. All rows except nnU-Net are Swin-UNETR. Unlike the volumetric DSC, we perform post-processing to detect lesions from predicted volumes to calculate lesion detection metrics.

	GTVp			GTVn		
System	Recall	Prec.	DSC	Recall	Prec.	DSC
nnU-Net (250 ep)	**0.766**	0.572	0.819	**0.615**	0.378	**0.773**
Full	0.685	0.625	**0.822**	0.553	**0.526**	0.772
Coarse	0.599	0.628	0.799	0.339	0.471	0.745
Coarse + Refine (α=0.1)	0.680	**0.638**	0.814	0.465	0.475	0.732

Qualitative results for the predictions are presented in Fig. 9.

C.3 Threshold selection for Segmentation

Depending on the model and dataset, the threshold can result in significantly varied FNR as seen in Tab 4. This highlights the need for using an established threshold finding algorithm like CRC. Naively choosing λ can either result in selecting too many cells, diminishing the speedup achieved, or too few cells, resulting an underestimated error. Furthermore, as Tab. 4 shows, different models and tasks have different thresholds. Thus effectively selecting a threshold requires a held-out calibration set, and CRC is a natural algorithm for choosing such a threshold. Also, using CRC allows us to expand our choices beyond solely threshold based methods. The pre-requisite for using CRC

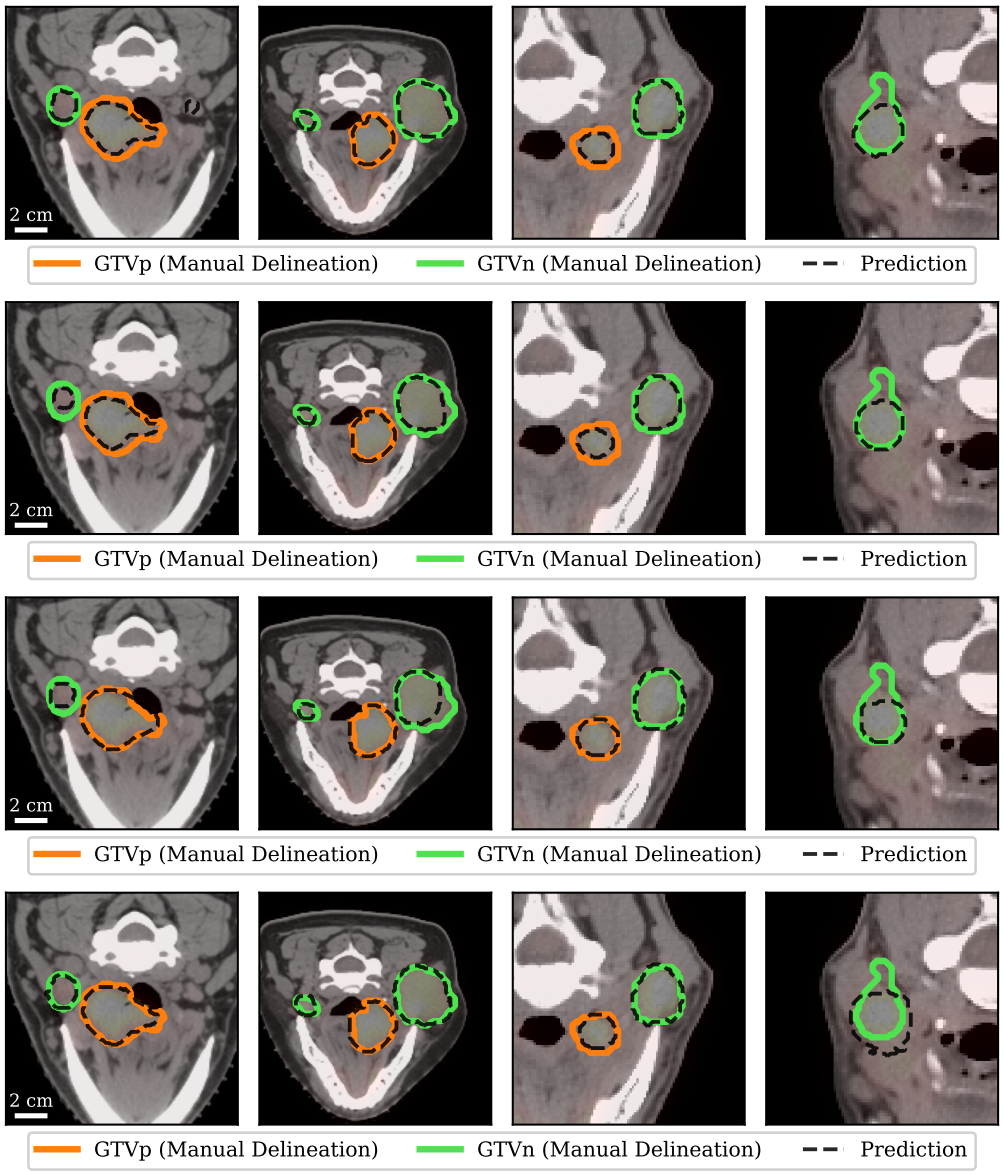

Figure 9: Grid of HECKTOR predictions. From top to bottom: nnU-Net, Swin-UNETR native resolution, Swin-UNETR Coarse, and Swin-UNETR Coarse + Refine.

is monotonically increasing prediction sets which can be accomplished by other selection criterion as well.

Table 4: A naively chosen like threshold can result in high FNR as the predicted scores are non-linear compared to the selection. CRC

		FNR at fixed λ			FNR with CRC ($\alpha = 0.1$)	
Dataset	Model	$\lambda = 0.5$	$\lambda = 0.1$	$\lambda = 0.01$	Calib.	Test
Brain Tumor	UNet	0.17 $\pm$ 0.01	0.072 $\pm$ 0.008	0.021 $\pm$ 0.003	0.085 $\pm$ <0.001	0.097 $\pm$ 0.006
	Swin-UNETR	0.11 $\pm$ 0.02	0.07 $\pm$ 0.01	0.031 $\pm$ 0.006	0.086 $\pm$ <0.001	0.091 $\pm$ 0.001
	VISTA3D	0.11 $\pm$ 0.01	0.07 $\pm$ 0.01	0.04 $\pm$ 0.01	0.086 $\pm$ 0.001	0.084 $\pm$ 0.004
HECKTOR	Swin-UNETR	0.86 $\pm$ 0.02	0.25 $\pm$ 0.02	0.13 $\pm$ 0.01	0.092 $\pm$ 0.001	0.072 $\pm$ 0.009

C.4 CASCADE Calcium Spike Prediction

The CASCADE experiment data processing, model design, and calibration methods are presented in this section.

C.4.1 Dataset Details

We use the *primary* group of the CASCADE calcium-imaging benchmark: datasets DS30, DS31, and DS32, recorded with the GCaMP8f/8m/8s indicators respectively. Each dataset file contains one or more trials, where a trial pairs a fluorescence ($\Delta F/F$) trace with ground-truth spike times obtained from simultaneous electrophysiological recording; frame rates vary by recording session, ranging from 8–60 Hz across the benchmark. The primary group comprises 117 source `.mat` files spanning 515 valid trials (trials with fewer than 2 recorded spike events or fewer than 10 fluorescence samples are discarded), at a mean spike-mass occupancy (fraction of non-empty coarse blocks at $r{=}8$) of 3.2–4.0%.

Because repeated trials from the same source file are highly dependent draws from the same underlying neuron (identical indicator, noise level, and firing statistics), we split *source files*, not individual windows or trials, into train/calibration/test partitions at a $60/20/20$ ratio, stratified per dataset so all three indicators appear in every partition. This file-level split is the exchangeable unit for the CRC calibration; every downstream step (coarse model training, CRC calibration, refiner training, evaluation) consumes the same fixed seed-0 split, and fluorescence windows are unique between splits. This split rules out test-set leakage; the standard sample-level calibration algorithm is then applied unchanged.

C.4.2 Data Preprocessing

Each trial's raw spike times are binned onto the trial's own fluorescence frame timestamps via a histogram, and the resulting spike-count sequence is convolved with a Gaussian kernel (standard deviation 200 ms, the CASCADE benchmark's own default) and rescaled to spikes/second to obtain the fine-resolution regression target $Y^{\Omega_{\text{fine}}}$. The coarse target $Y^{\Omega_{\text{coarse}}}$ is obtained by *sum*-pooling (not averaging) the raw spike counts by the coarse block factor $r{=}8$, so that each coarse cell's value is exactly the expected spike count inside it, matching the aggregation operator of Eq. 8 for regression. The CRC loss is the fraction of missed spike-count mass. The fluorescence trace itself is z-score normalized per source file (median and $1.4826\times$MAD, computed once across all of a file's trials to avoid leaking window-local statistics), then windowed into fixed-length segments of 1024 fine samples (128 coarse blocks at $r{=}8$), with stride 512 for training, calibration, and test.

C.4.3 Model Architecture

Both the baseline and coarse networks in the pipeline are 1D convolutional nets sharing a common building block (`ConvBlock`: two $\mathrm{Conv1d}(k{=}3)$+GroupNorm+GELU layers) and a common output head design: a `QuantileHead` predicting $Q{=}4$ quantiles ($\tau \in \{0.05, 0.5, 0.9, 0.95\}$) by regressing the $\tau{=}0.05$ quantile plus non-negative (softplus) increments, cumulatively summed to guarantee monotonicity.

- **Full baseline** $\mathcal{F}_{\Omega_{\text{fine}}}$ (`FineUNet1D`): a 4-level 1D U-Net, base width 64, with $\mathrm{MaxPool1d}(2)$ downsampling and nearest-neighbor upsample + skip-concatenation on the decoder side, applied to the full 1024-sample window.
- **Coarse model** $\mathcal{F}_{\Omega_{\text{coarse}}}$ (`FineOnLowRes`): reuses $\mathcal{F}_{\Omega_{\text{fine}}}$'s architecture and parameter count *exactly*, but is fed the trace block-averaged down to $\mathrm{window}/r = 128$ samples before the first convolution, rather than a separate, narrower network. This isolates the coarse model's speedup to resolution reduction alone: at identical capacity to $\mathcal{F}_{\Omega_{\text{fine}}}$, this gives an $8.0\times$ FLOPs reduction for the coarse pass (exactly r, since a fixed-width conv net's FLOPs scale linearly with sequence length), so no reviewer can attribute the pipeline's speedup to simply swapping in a smaller-capacity model.
- **Refiner** $\mathcal{G}_{\Omega_{\text{coarse}}\to\Omega_{\text{fine}}}$ (`LocalRefinerUNet1D`): a 2-level 1D U-Net (base width 32) over a context-padded crop of length $L = r(1 + 2 \times \mathrm{context_blocks}) = 136$ samples ($\mathrm{context_blocks}{=}8$) around each retained coarse block, whose full-length decoder output

is center-cropped to the middle $r{=}8$ samples. Rather than predicting the fine-resolution rate curve $Y^{\Omega_{\text{fine}}}$ directly (which would hard-zero discarded neighboring blocks regardless of proximity to a confirmed detection), the refiner uses a `CountHead` (global-average-pool + linear projection) to predict a single non-negative spike-*count* estimate per retained block, matching $Y^{\Omega_{\text{coarse}}}$'s own resolution. The full continuous rate curve is then reconstructed by convolving this sparse, mostly-zero per-block count estimate with the same Gaussian kernel used to build $Y^{\Omega_{\text{fine}}}$ from raw spike counts. Since that kernel is mass-preserving, bounding missed count mass at the CRC level α (calibrated against $Y^{\Omega_{\text{coarse}}}$) also bounds the missed mass of the reconstructed continuous curve at the same α. A detection's estimated count can be smeared physically into its discarded neighbors rather than being hard-zeroed there during post processing and presentation. **Note that the certification region is decoupled from the presentation in high resolution.**

Table 5 summarizes parameter counts and analytic FLOPs (at $\text{window}{=}1024$) for the coarse model and full-resolution baseline, computed with each model's default constructor arguments.

Table 5: CASCADE model architectures, parameter counts, and analytic FLOPs. The total FLOPs for the refiner depends on the number of selected windows and is therefore omitted.

Model	Role	Params	FLOPs (window=1024)
`FineUNet1D`	Full baseline ($\mathcal{F}_{\Omega_{\text{fine}}}$)	10,475,204	2,668,494,848
`FineOnLowRes`	Coarse ($\mathcal{F}_{\Omega_{\text{coarse}}}$)	10,475,204 (same)	333,561,856 (8.0× fewer)
`LocalRefinerUNet1D`	Refiner (count head)	158,721	–

C.4.4 Training and Calibration Details

All three networks (baseline, coarse model, refiner) are trained with Adam (learning rate 10^{-3}) for 30 epochs, minimizing the quantile pinball loss summed over $\tau \in \{0.05, 0.5, 0.9, 0.95\}$ for the baseline and coarse model, and mean-squared error against the block spike count for the refiner's count head. CRC is calibrated on the coarse model's predicted $\tau{=}0.9$ quantile score field (empirically the best-performing of the median/q90/ q95 candidates by screening ROC AUC), using the missed-signal-mass loss of Eq. 9, with a retention halo of 1 coarse block on either side of every selected block (chosen heuristically). For the reported operating point ($\alpha{=}0.1$), CRC yields a calibrated $\hat{\lambda}$ with a raw-score cutoff of 0.877 and a retained fraction of 8.4% of coarse blocks on held-out test data, corresponding to the $4.77\times$ FLOPs speedup reported in Table 6. A separate refiner is trained per calibrated α (each refiner's training crops are drawn from that α's own retention distribution), and reconstructed-curve accuracy is reported as mean absolute error (MAE) against $Y^{\Omega_{\text{fine}}}$ on the test split, compared against the full-resolution baseline's own median ($\tau{=}0.5$) prediction on the same windows.

C.4.5 Results

The reconstruction results with varying α, their corresponding retained fraction, and final error is shown in Tab.6.

Table 6: Reconstructed MAE is reported in Hz, with the gap to the full-resolution baseline (0.250 Hz) in parentheses.

α	Retained fraction	FLOPs speedup	Reconstructed MAE (Δ)
0.01	1.00	0.89×	0.30 (+0.05)
0.05	0.48	1.65×	0.34 (+0.09)
0.10	0.08	4.77×	0.32 (+0.06)
0.20	0.07	5.21×	0.35 (+0.10)

C.5 Galaxy Flux Field Super-Resolution

The galaxy flux field experiment's data generation, model design, training, and calibration methods are presented in this section.

C.5.1 Dataset Details

Each sample is a simulated 1024×1024 sky field at a pixel scale of $0.2''$, covering roughly $3.4' \times 3.4'$. The number of sources per field is Poisson-distributed with mean 60, so the realized count varies field to field and occupancy is a statistic over fields rather than a fixed per-field quantity. We generate 3000 fields in total, partitioned $2000/500/500$ into train, calibration, and test splits. One field is one exchangeable unit for conformal risk control: unlike the calcium-imaging study there is no windowing, so a field is never split across partitions.

Sources are real galaxies drawn from the COSMOS 25.2 magnitude-limited sample ($81{,}499$ objects) rather than point sources. We use the analytic Sérsic fits to the catalog objects rather than the raw telescope pixel data, so each source carries the real size, shape, orientation, and Sérsic-index distribution of an observed galaxy but is a smooth, noiseless profile.

We avoid the raw pixel data because it carries the original exposure's own correlated noise: rescaling a source to this experiment's flux prior can require a factor of more than four orders of magnitude, which would amplify that embedded noise by the same factor and introduce spatially non-stationary contamination unrelated to the sky and read-noise model the experiment is built around.

The three splits draw from *disjoint contiguous slices* of the catalog, in the proportions of the split sizes. We ensure the same galaxy shape (same catalog object, differing only in flux, position, rotation, and noise) cannot appear in both a training field and a test field, since every field otherwise samples uniformly from the whole catalog. This is a known failure mode for this catalog, and holding out objects rather than only images follows standard practice for COSMOS-based generative and deblending studies (Arcelin et al., 2021).

C.5.2 Data Generation Pipeline

Every field is reproducible from a single integer seed; all randomness is drawn from one seeded generator, so caching amounts to calling the generator once per seed. For each field we draw the source count and then, for each source, draw a flux log-uniformly on $[10^2, 10^5]$ counts, a position uniform in the field interior (a 20-pixel margin is excluded so sources are not truncated at the boundary), a catalog object from that split's slice, and a uniform rotation. Each source is rendered into a 65×65 pixel stamp centered on its position.

The same object is rendered *twice*, in two different roles, and the two blurring operators are kept strictly separate:

- **Input image.** The source is convolved with the instrument point spread function, a Kolmogorov profile of full width at half maximum $0.8''$, i.e. the forward model of a real observation.
- **Regression target.** The same source is convolved instead with a small idealized Gaussian kernel with a standard deviation of 1.5 pixels ($0.3''$), and no noise is added.

Neither kernel is derived from the other. The target is therefore a deconvolved, noise-free surface-brightness map of the same scene: the task is simultaneously deblurring and denoising, not merely upsampling.

After all sources are drawn, synthetic detector noise is added to the input image only, as Poisson noise about a sky level of 50 counts plus Gaussian read noise of 5 counts at unit gain. This is the only noise in the experiment. The known sky level is subtracted from the image before it is presented to any network, so a network's input is a background-subtracted, noisy, PSF-convolved field and its target is the same field's noiseless, sharply rendered flux.

The coarse partition is the 8×8-pixel cell grid, giving $128 \times 128 = 16{,}384$ coarse cells per field. The coarse target is obtained by *sum*-pooling the fine target over each cell, not averaging, so a cell's value is exactly the total flux inside it and matches the regression aggregation operator of Eq. 8;

the CRC loss is then literally the fraction of missed flux. The coarse model's input is obtained by *average*-pooling the observed image by the same factor, because the image is a surface brightness rather than a total.

C.5.3 Model Architecture

All three networks share a common convolutional block (two 3×3 convolutions, each with group normalization and a GELU nonlinearity) and, for the two predicting networks, a common output head predicting $Q = 4$ quantiles ($\tau \in \{0.05, 0.5, 0.9, 0.95\}$) of the log-transformed flux. The head regresses the τ=0.05 quantile plus non-negative softplus increments, cumulatively summed, so predicted quantiles are monotone by construction and can never cross. All flux regression is performed in $\log(1 + \cdot)$ space, which is necessary given the three-decade flux prior.

- **Full baseline.** A four-level two-dimensional U-Net of base width 64, with max-pool downsampling and nearest-neighbor upsampling with skip concatenation, applied to the full 1024×1024 field.
- **Coarse pass.** Reuses the full-resolution baseline's architecture and parameter count *exactly*, applied to the $8\times$ average-pooled 128×128 field, rather than a separate narrower network. This isolates the coarse pass's savings to resolution alone: at identical capacity it is a $64\times$ reduction in floating-point operations (exactly the square of the downsampling factor, since a fixed-width convolutional network's cost scales with the number of sites), so the speedup cannot be attributed to substituting a smaller-capacity model. Its output is one predicted flux total per coarse cell.
- **Refiner.** A small local operator applied independently to each certified cell, taking the 12×12 full-resolution window formed by the 8×8 cell plus a two-pixel context border, together with the coarse model's own predicted median flux for that cell as a scalar conditioning input, and returning an 8×8 flux patch. Rather than re-predicting *how much* flux the cell contains, which the coarse model already estimates in a calibrated way, the refiner predicts only *where within the cell* the flux lies. Its output is rectified, normalized to sum to one over the cell, and then multiplied by the coarse model's predicted total. Flux conservation between the coarse and refined stages is therefore exact by construction rather than something the refiner must learn, and the refiner is left with the purely spatial distribution of the problem, which is what makes a network of $17{,}809$ parameters sufficient. Uncertified cells are filled by smoothly upsampling the coarse prediction, so a selection miss degrades to coarse-model quality rather than to blank sky.

Table 7 summarizes parameter counts and analytic operation counts for the three networks.

Table 7: Galaxy flux field model sizes and analytic floating-point operation counts, per 1024^2 field for the full-resolution baseline and the coarse model, and per certified cell for the refiner.

Model	Role	Params	FLOPs
2D U-Net, base width 64	Full baseline	31,389,764	1.78×10^{12}
Same U-Net, $8\times$-downsampled input	Coarse	31,389,764 (same)	2.78×10^{10} ($64\times$ fewer)
Local refinement operator	Refiner	17,809	3.05×10^{6} per cell

C.5.4 Training and Calibration Details

All three networks are trained with Adam at a learning rate of 10^{-3} for 30 epochs on the 2000 training fields, holding out a fixed 10% of them as an internal validation split, with all randomness seeded. The full-resolution baseline and the coarse model minimize the quantile pinball loss summed over $\tau \in \{0.05, 0.5, 0.9, 0.95\}$, against the log-transformed fine target and the log-transformed sum-pooled coarse target respectively. The refiner minimizes an L_1 loss in $\log(1 + \cdot)$ space between its predicted patch and the true full-resolution flux patch. The refiner's output is anchored to the coarse model's calibrated total.

The refiner is trained on the *certified* distribution rather than on uniformly sampled cells. We run the deployed, already-calibrated coarse model over the training split, apply the calibrated threshold and

halo, and draw the refiner's training crops from the resulting certified regions; this yields $716{,}629$ cells over 2000 fields, 2.19% of all cells. Halo cells are deliberately included, since they are near-empty by construction and are where boundary truncation errors appear. Each cell's coarse-predicted median flux is cached at the same time.

CRC is calibrated on the 500 calibration fields. The score field is the coarse model's predicted $\tau{=}0.9$ quantile of cell flux. We use an upper quantile rather than the median so that a cell whose median flux is low but whose upper tail is high is still certified. Cell weights are flux mass, the aggregation operator of Eq. 8, so the calibrated loss of Eq. 9 is the fraction of a field's true flux that falls in uncertified cells. The certified set is dilated by a halo of one coarse cell (chosen heuristically) in an eight-connected neighborhood before the loss is evaluated. Calibration scans an ascending grid of 500 candidate thresholds and takes the largest whose finite-sample corrected risk $(n\hat{R}+1)/(n+1)$ remains at or below α.

At the reported operating point $\alpha{=}0.1$, calibration returns a $\hat{\lambda}$ with a raw-score cutoff of 7.69 in $\log(1+\mathrm{flux})$ units, a flux total of approximately 2175 counts per cell. The calibration risk is 0.040 (0.042 after the finite-sample correction), and on the held-out test split the empirical risk is 0.040 (0.042 corrected), comfortably within α, at a certified fraction of 2.18% of coarse cells. Certifying 2.18% of cells leaves the pipeline at $61.6\times$ fewer floating-point operations than the full-resolution baseline, against a ceiling of $64\times$ set by the coarse pass alone; the refinement of certified cell in an average field costs about 4% of what the coarse pass costs, so the coarse pass, not the refinement, bounds the achievable speedup.

As a reference for how tight the conformal selection is, we also compute an α-matched oracle, the threshold applied directly to the *true* coarse flux totals that attains exactly risk α under the same halo. That oracle certifies 1.43% of cells, so the calibrated coarse model certifies $1.53\times$ the cells an oracle selector would need at the same risk level.

C.5.5 Results

We provide an assortment of predictions from our model in Fig. 10.

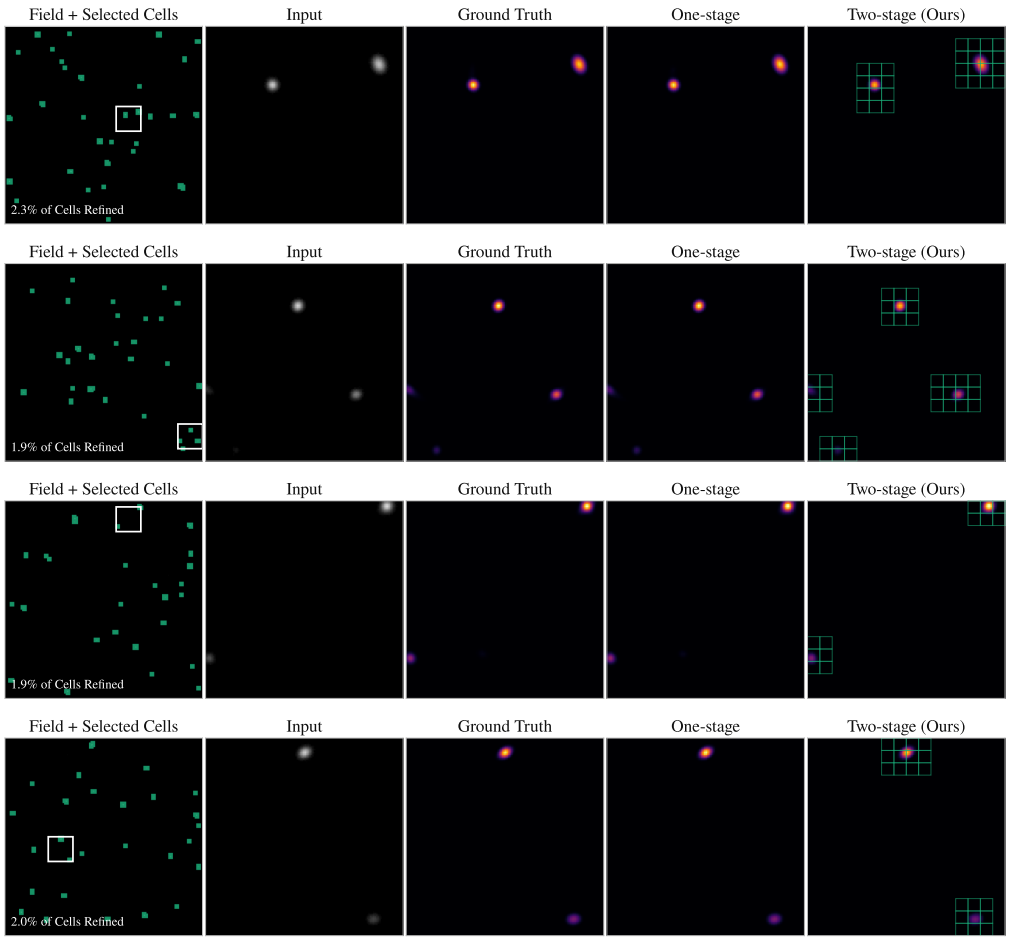

Figure 10: Grid of galaxy flux field predictions.