Partial VOROS: A Cost-aware Performance Metric for Binary Classifiers with Precision and Capacity Constraints

Christopher Ratigan[1], **Kyle Heuton**[2], **Carissa Wang**[2], **Lenore Cowen**[1 2], **Michael C. Hughes**[2]
[1]Department of Mathematics, Tufts University, Medford, MA, USA
[2]Department of Computer Science, Tufts University, Medford, MA, USA

Abstract

The ROC curve is widely used to assess binary classifiers. Yet for some applications, such as alert systems for monitoring hospitalized patients, conventional ROC analysis cannot meet two key deployment needs: enforcing a constraint on precision to avoid false alarm fatigue and imposing an upper bound on the number of predicted positives to represent the capacity of hospital staff. The usual area under the curve metric also does not reflect asymmetric costs for false positives and false negatives. In this paper we address all three of these issues. First, we show how the subset of classifiers that meet precision and capacity constraints occupy a feasible region in ROC space. We establish the polygon-shaped geometry of this region. We then define the *partial area of lesser classifiers*, a performance metric that is monotonic with cost and only accounts for the feasible region. Averaging this area over a desired distribution for cost parameters results in the partial volume over the ROC surface, or *partial VOROS*. In experiments predicting mortality risk from vital sign history on several datasets, we show this cost-aware metric can outperform alternatives at ranking classifiers for in-hospital alerts.

1 INTRODUCTION

When a classifier of binary events is deployed in a high-stakes application, it must respect important operational constraints. First, context-specific costs mean that false positive predictions usually have different consequences than false negatives in a given task. Developing and evaluating classifiers in a cost-aware way is key to deployment success (Provost and Fawcett, 1997; Drummond and Holte, 2000). Second, stakeholders may specify viable ranges for certain performance metrics for the system to be beneficial. Finally, stakeholders may have capacity constraints, in terms of the overall number of positive predictions or negative predictions they can handle smoothly when deployed.

In this work, our goal is to develop a performance metric that can effectively rank classifiers when costs, performance constraints, and capacity constraints all matter. Previous work has suggested many metrics and visuals for evaluating binary classifiers, surveyed later in Sec. 7. We take as a starting point a cost-aware analysis of the receiver-operating-characteristic (ROC) curve. Recent work by Ratigan and Cowen (2025) lifts the classic 2D ROC curve into a 3D surface where the third axis defines the cost. They proposed a performance metric, the volume over the ROC surface or *VOROS*, that can identify when a binary classifier outperforms another given a task-specific range of estimated costs. Our work here extends this analysis to incorporate constraints on precision and capacity, which are critical in many applications such as fraud detection, credit scoring, and early warning systems.

As a motivating task of interest, consider the evaluation of alert systems for monitoring the health of hospitalized patients. Here, a classifier must take in recent data about a patient's health and determine whether or not to alarm. An alarm indicates the patient's health may be deteriorating, so doctors or nurses should check on that patient soon. An ever-increasing body of literature has developed such systems via machine learning (Abella Álvarez et al., 2013; Hyland et al., 2020; Sendak et al., 2020; Muralitharan et al., 2021; Edelson et al., 2024). Careful evaluation of such systems is critical to ensure they balance tradeoffs

Proceedings of the 29th International Conference on Artificial Intelligence and Statistics (AISTATS) 2026, Tangier, Morocco. PMLR: Volume 300. Copyright 2026 by the author(s).

Code: github.com/tufts-ml/partial-VOROS.

appropriately and provide a net benefit to the patient population and the hospital.

Hospital-based alert systems naturally have the three aforementioned operational constraints:

- First, there are asymmetric costs to the different kinds of mistakes. A false positive has some cost by taking valuable time from clinical staff that could be spent on other patients with greater needs. A false negative incurs even more cost, as this means a patient did get sicker or even die and an alert that might have helped was never issued.
- Second, hospital staff often express a key performance metric constraint: the alert classifier's *precision*, the fraction of all alarms that are true positives, needs to meet some minimum value for the system to be viable (Harrison et al., 2015; Rath and Hughes, 2022). A survey of critical care physicians in South Korean hospitals (Park et al., 2022) found that too many false positives were the top concern about early warning systems; the median response requested a precision of at least 28.5%. Clinical staff may learn to ignore many or all alarms from the system altogether if its precision is not tolerable, a problem known as *alarm fatigue* (Cvach, 2012). Ignoring alarms entirely due to low precision has been a documented safety concern for decades (Sendelbach and Funk, 2013; Albanowski et al., 2023).
- Finally, the staff's capacity to respond to alarms must be accounted for. In a typical U.S. ICU, the patient-to-doctor ratio is on average 11.8 and almost always above 6 (Kahn et al., 2023). If alarms for all patients happened at once, only a fraction could be triaged right away. It is crucial to ensure system evaluation takes into account such constraints.

In this paper, we develop a new performance metric, the *Partial VOROS*, which extends the VOROS from Ratigan and Cowen (2025) to account for precision and capacity constraints while preserving the original VOROS' ability to handle asymmetric costs. In Sec. 3, we formalize how these constraints narrow the entirety of ROC space to a feasible region satisfying all constraints. In Sec. 4, we provide formulas for evaluating areas of lesser classifiers and volumes over this feasible region using desired cost distributions. In Sec. 5 and 6, we examine prediction tasks on real health records data, showing how our partial VOROS can identify promising classifiers by accounting for constraints and costs in ways other metrics cannot.

2 BACKGROUND

We consider an observed dataset $\mathcal{D} = \{X_i, Y_i\}_{i=1}^{N_*}$ of N_* total pairs indexed by i of a feature vector $X_i \in \mathcal{X}$ and its associated binary label $Y_i \in \{0, 1\}$. Let $\mathcal{P}$ denote the subset of this dataset with positive labels $Y_i = 1$, and $\mathcal{N} = \mathcal{D} \backslash \mathcal{P}$ denote the subset with negative labels.

We wish to use dataset $\mathcal{D}$ to evaluate a score-producing binary classifier $\mathcal{F} : \mathcal{X} \to \mathbb{R}$. This score can be thresholded to produce a binary prediction $\hat{Y}_i$. We refer to a classifier using threshold τ to make binary predictions as a **binarized classifier**, denoted $\mathcal{F}_\tau$. By comparing predictions to true labels over all N_* examples, we can count the number of true positives N_{TP}, false positives N_{FP}, true negatives N_{TN}, and false negatives N_{FN}.

We now establish concepts needed for a cost-aware ROC analysis, following Ratigan and Cowen (2025).

Definition 1 (ROC Space)**.** *The performance of binarized classifier $\mathcal{F}_\tau$ on dataset $\mathcal{D}$ at a particular threshold τ can be represented as a point (h, k) in two-dimensional space where h is the false positive rate and k the true positive rate. We refer to the set of all possible such (h, k) points, which span the unit square $[0.0, 1.0] \times [0.0, 1.0]$, as **ROC space**.*

ROC space is useful for assessing classifier performance with respect to cost. Let C_{FP} define the cost of a false positive and C_{FN} the cost of a false negative. Given a dataset, the total cost of all possible mistakes is given by $C_{\text{FP}}|\mathcal{N}| + C_{\text{FN}}|\mathcal{P}|$. Naturally, assigning costs to points in ROC space depends on the *ratio* of costs $C_{\text{FP:FN}} = \frac{C_{\text{FP}}}{C_{\text{FN}}}$ and the relative sizes of $|\mathcal{P}|$ and $|\mathcal{N}|$. Following Ratigan and Cowen (2025), we can capture this dependency in one parameter $t \in [0.0, 1.0]$.

Definition 2 (Fractional Cost Parameter)**.** *Let $t = \dfrac{C_{\text{FP}}|\mathcal{N}|}{C_{\text{FP}}|\mathcal{N}| + C_{\text{FN}}|\mathcal{P}|}$ denote the portion of aggregate misclassification cost due to false positives. We have $0 \leq t \leq 1$.*

When the dataset $\mathcal{D}$ is fixed, an increase in t implies the cost ratio $C_{\text{FP:FN}}$ is increasing. We cannot compare t across datasets with different class balances.

Definition 3 (Normalized Cost)**.** *The **normalized cost** of a ROC point (h, k) for dataset $\mathcal{D}$ given parameter t is*

$$\text{Cost}_t(h, k) = th + (1-t)(1-k) = \frac{C_{\text{FP}} N_{\text{FP}} + C_{\text{FN}} N_{\text{FN}}}{C_{\text{FP}}|\mathcal{N}| + C_{\text{FN}}|\mathcal{P}|}.$$

The worst possible point in ROC space, $(1, 0)$, would have cost of 1.0. The best point $(0, 1)$ would have cost 0.0. All other points have cost in between 0.0 and 1.0, reflecting their cost relative to the worst point.

This formulation of cost as a linear function of ROC coordinates (h, k) is well-known (Drummond and Holte, 2000). However, Ratigan and Cowen (2025) was the first paper to add a separate axis based on the t parameter defined above to ROC space to compare classifiers

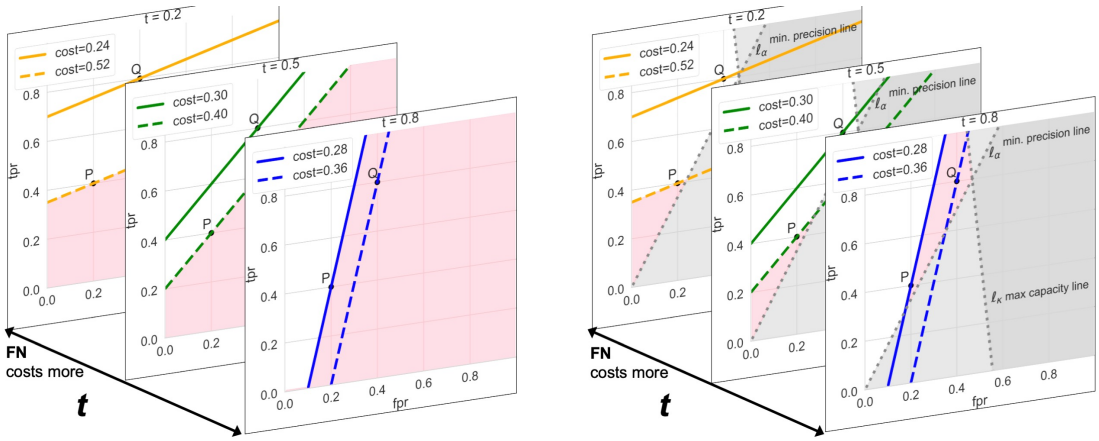

Figure 1: Overview of VOROS (left) and our new partial VOROS (right). *Left:* At each fractional cost parameter $t \in [0, 1]$, we draw iso-performance lines for points P and Q in ROC space. Solid lines have lower cost than dashed lines. Points below a line have higher cost than that line. A point's *area of lesser classifiers*, colored light pink here for point P, is the area below that point's iso-performance line. The VOROS (Ratigan and Cowen, 2025) for a classifier is computed by finding at each t the maximum area of lesser classifiers for any point in its ROC curve, then integrating over a desired distribution $p(t)$. *Right:* Partial VOROS **excludes** regions in gray that do not achieve a minimum precision α or exceed a maximum capacity κ (too many positive predictions). These limits correspond to linear constraints in ROC space.

with respect to variable costs. The key notion here was the ROC surface defined below.

Definition 4 (ROC Surface)**.** *Let (h, k) be a point in ROC space, the **ROC surface** associated to (h, k) is the saddle surface in 3D space with coordinates x, y, t given equivalently by $t = \frac{y-k}{y-k+x-h}$ or $y = \frac{t}{1-t}(x-h)+k$, where t is the fractional cost from Def. 2.*

Given a fixed t value, this formulation allows us to find all points (x, y) in ROC space with the same cost as the point (h, k). This same-cost set is known as an *iso-performance line* (Provost and Fawcett, 2001).

Definition 5 (Iso-performance Line)**.** *Let (h, k) be a point in ROC space and let t be a fixed fractional cost parameter. Then the line in ROC space*

$$y = \frac{t}{1-t}(x-h) + k$$

*represents all points (x, y) with the same cost as (h, k) using t and is called an **iso-performance line**.*

Each ROC panel in Fig. 1 visualizes for a specific t the iso-performance lines for the same two points P and Q. Varying t adjusts the iso-performance line's slope.

3 BOUNDS ON ROC SPACE

A common criticism of ROC space and the area under the ROC curve is that it fails to measure the performance of a binary classifier under appropriate operational constraints. These criticisms still hold for the ROC surface and the volume over the ROC surface measure of Ratigan and Cowen (2025). In this section we introduce two conditions – a bound on precision and a bound on capacity – that restrict the allowed classifiers in ROC space. These constraints are motivated by early warning classifiers for hospitalized patient monitoring (Harrison et al., 2015; Rath and Hughes, 2022). We then define a portion of ROC space meeting these constraints and other key assumptions.

3.1 Precision Bound

Definition 6 (Precision)**.** *The precision of the binarized classifier $\mathcal{F}_\tau$ on dataset $\mathcal{D}$ is the fraction of positive predictions that are correct:*

$$Prec = \frac{N_{\text{TP}}}{N_{\text{TP}} + N_{\text{FP}}}$$

Precision is also known as positive predictive value.

Definition 7 (Precision Bound)**.** *To be feasible, a classifier must satisfy a **minimum precision bound**:*

$$Prec \geq \alpha$$

Here, the desired precision $\alpha > 0$ can be set by talking with stakeholders about their tolerance for false alarms. This constraint is motivated by applications of ML to early warning alert systems, especially in hospitals (Harrison et al., 2015; Rath and Hughes, 2022).

For dataset $\mathcal{D}$, let $p = \frac{|\mathcal{P}|}{|\mathcal{D}|}$ define the **prevalence**, the fraction of all examples that are positive. Using p, then we can recast the definition of precision itself, as well as the bound above, in terms of ROC coordinates.

Lemma 8 (Precision Bound in ROC Space)**.** *Let $\mathcal{F}_\tau$ be a binarized classifier with ROC coordinates (x, y) on a dataset with prevalence p. Then*

$$Prec \geq \alpha \quad if \quad y \geq \frac{\alpha(1-p)}{(1-\alpha)p}x.$$

We call the line in ROC space where this holds with equality the ***minimum precision line*** *ℓ_α.*

Proof. We write precision in terms of x, y and p, using $N_{\text{TP}} = |\mathcal{P}|y$ and $N_{\text{FP}} = |\mathcal{N}|x$. Then, solve for y:

$$\frac{py}{py + (1-p)x} \geq \alpha \;\Rightarrow\; y \geq \frac{\alpha(1-p)}{(1-\alpha)p}x.$$

This algebra is valid when $1-\alpha > 0, p > 0$. □

This inequality means that if we require a priori that classifiers have at least α precision, then rather than the entire unit square of ROC space, we only consider the region above the minimum precision line $\ell_\alpha : y = \frac{\alpha(1-p)}{(1-\alpha)p}x$ through the origin. When p is small or when α is large, then we ignore a large portion of ROC space.

3.2 Capacity Bound

Beyond precision, another issue prevalent in applications is that the system that responds to alerts (positive predictions) has a maximal *capacity*. Unlike precision, which is a rate, capacity is often an absolute cutoff due to limited resources. In hospital alert systems, capacity constraints arise due to limited time to tend to patients by existing staff. In information retrieval, there may be limited time to handle relevant documents. Throughout, we assume that *not alarming* has no impact on capacity. This is reasonable in hospitals, as no resources beyond standard care need be allocated when there is no alarm.

Definition 9 (Capacity Bound)**.** *To be feasible, the total number of predicted positives a classifier produces must not exceed a provided* ***maximum capacity****:*

$$N_{\text{TP}} + N_{\text{FP}} \leq \kappa$$

Here, the value of $\kappa > 0$ can be set by stakeholders, indicating the maximum number of alerts that can be handled by the system given typical resources.

Lemma 10 (Capacity Bound in ROC Space)**.** *Let $\mathcal{F}_\tau$ be a binarized classifier with ROC coordinates (x, y) for dataset $\mathcal{D}$. If capacity bound κ is satisfied, then*

1. *The number of predicted positives is: $|\mathcal{P}|y + |\mathcal{N}|x$.*
2. *We have $y \leq \frac{1}{|\mathcal{P}|}(\kappa - |\mathcal{N}|x)$.*

We call the line in ROC space where this holds with equality the **maximum capacity line**, ℓ_κ.

Proof. The number of predicted positives $N_{\text{TP}}+N_{\text{FP}}$ is $|\mathcal{P}|y+|\mathcal{N}|x$ by definition. Bound satisfaction means $|\mathcal{P}|y+|\mathcal{N}|x \leq \kappa$. Solving for y completes the proof. □

3.3 Feasible Region

We now consider enforcing both precision and capacity constraints, along with some practical assumptions typical in our target applications. These constraints narrow down ROC space to a particular smaller region we call the feasible region. This is similar to what Morasca and Lavazza (2020) call a "region of interest", though ours incorporates precision and capacity.

Definition 11 (Practical Assumptions)**.** *In order to focus on the cases that arise in practical settings, we make some assumptions going forward (remaining edge cases are handled in App. B). From here on, in the main paper, we assume our dataset, precision limit α, and capacity limit κ satisfy:*

- $|\mathcal{P}| < |\mathcal{N}|$*: Negatives are more common.*
- $p < \alpha < 1.0$ *: Precision is reasonable.*
- $0 < \kappa < |\mathcal{D}|$*: Capacity is non-trivial.*
- $t < \frac{\alpha|\mathcal{N}|}{\alpha|\mathcal{N}|+(1-\alpha)|\mathcal{P}|}$*: Never-alarm has maximal cost.*

The strict inequality $\kappa < |\mathcal{D}|$ in item 3 ensures the always-alarm baseline is not feasible. The t bound in the last item ensures that the never-alarm baseline maximizes cost for the feasible region (see App. C).

Definition 12 (Feasible Classifier)**.** *Let $\mathcal{F}_\tau$ be a classifier with ROC coordinates (h, k). Given a dataset, precision limit α, and capacity limit κ that satisfy Def. 11, we call $\mathcal{F}_\tau$ a* ***feasible classifier*** *and (h, k) a* ***feasible point*** *if $Prec \geq \alpha$ and $N_{\text{TP}}+N_{\text{FP}} \leq \kappa$.*

Definition 13 (Feasible Region)**.** *For a fixed dataset and limits α, κ, the* ***feasible region*** *in ROC space is the set of all feasible points.*

3.4 Geometry of the Feasible Region

We can define the geometry of the feasible region via the minimum precision line ℓ_α and the maximum capacity line ℓ_κ. The line ℓ_α has its y-intercept at the origin, and slope within $(1.0, +\infty)$ for $\alpha \in (p, 1.0)$ by Def. 11. Similarly, ℓ_κ has fixed slope $-\frac{|\mathcal{N}|}{|\mathcal{P}|}$ (always <-1), with free y-intercept $\frac{\kappa}{|\mathcal{P}|}$ for $\kappa \in (0, |\mathcal{D}|)$. Where these lines intersect determines what kind of polygon the feasible region forms.

The 3 cases below exhaustively cover all possible κ for a given feasible value of $\alpha \in (p, 1.0)$ subject to the practical assumptions made in Definition 11, as diagrammed in Figure 2.

- (1) $0 < \kappa < |\mathcal{P}|$: Triangle that excludes (0,1).

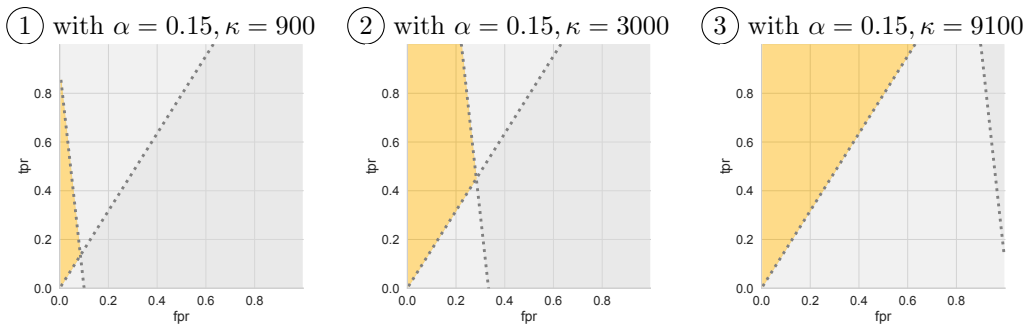

Figure 2: Cases for distinct feasible region polygons (colored in gold) under the practical conditions of Def. 11. We set $|\mathcal{P}| = 1000$, $|\mathcal{N}| = 9000$ so $p = 0.1$, roughly matching the mortality alert applications in Sec. 6.

Here, ℓ_α and ℓ_κ intersect inside ROC space. The capacity bound eliminates the perfect classifier.

- (2) $|\mathcal{P}| \leq \kappa < \frac{1}{\alpha}|\mathcal{P}|$. Quadrilateral including (0,1). ℓ_α and ℓ_κ intersect inside ROC space, with capacity allowing the perfect classifier.
- (3) $\frac{1}{\alpha}|\mathcal{P}| \leq \kappa < |\mathcal{D}|$. Right Triangle including (0, 1). ℓ_α and ℓ_κ intersect above ROC space (above the y=1 upper edge), so the capacity constraint is made redundant and precision alone dominates.

Edge cases that violate the assumptions of Def. 11 are handled in App. B. Focusing on 3 main cases here simplifies presentation for most practical situations.

We can now define the polygon enclosing the feasible region in terms of specific vertices for each case:

Definition 14 (Notation for Vertices)**.** *Let $i, j \in \{0, 1\}$ and let $\beta \in \{\alpha, \kappa\}$. Define the following 9 vertices for all possible i, j, β*

(1-4) v_{ij} is the intersection of $x = i$ and $y = j$
(5-6) $v_{\beta j}$ is the intersection of ℓ_β and $y = j$
(7-8) $v_{i\beta}$ is the intersection of $x = i$ and ℓ_β.
(9) $v_{\alpha\kappa}$ is the intersection of ℓ_α and ℓ_κ.

Definition 15 (Feasible Region Polygon)**.** *Moving clockwise from the origin, the bounding vertices of the region of interest are $v_{00}v_{0\kappa}v_{\alpha\kappa}$ in case 1, $v_{00}v_{01}v_{\kappa 1}v_{\alpha\kappa}$ in case 2, and $v_{00}v_{01}v_{\alpha 1}$ in case 3.*

Area of Feasible Region. We can compute the area of the feasible region by applying the well-known shoelace formula (Lee and Lim, 2017; Zwillinger, 2018), which computes the area of a polygon given its boundary vertices (see App. B.3).

Dependence on Class Balance. The slopes of both lines ℓ_α and ℓ_κ and thus the feasible region's area depends explicitly on the sizes of $\mathcal{P}$ and $\mathcal{N}$. These sizes must be fixed and known prior to model evaluation. This dependence is fundamental to evaluating precision and capacity; neither can be checked solely from a normalized confusion matrix.

4 PARTIAL AREA AND VOLUME

We now study how to perform *cost-aware* ranking of classifiers in the feasible region. Ratigan and Cowen (2025) introduced two key ideas for cost-aware ranking in their unconstrained setting: the notion of a lesser classifier and the area of lesser classifiers in ROC space. In Sec. 4.1, we extend these ideas to enforce a minimum precision α and a maximum capacity κ. Next, in Sec. 4.2, we explain how to lift this analysis to 3D space (x, y, t). Finally, we show how to integrate over a given distribution $p(t)$ to compute a partial volume.

4.1 Partial Area of Lesser Classifiers

Definition 16 (Lesser Classifier)**.** *Let $\mathcal{F}_1, \mathcal{F}_2$ be feasible binarized classifiers with ROC coordinates $(h_1, k_1), (h_2, k_2)$ on dataset $\mathcal{D}$. Then $\mathcal{F}_1$ is a **lesser classifier** of $\mathcal{F}_2$ at cost parameter t if it has worse cost: $Cost_t(h_1, k_1) > Cost_t(h_2, k_2)$.*

Definition 17 (Partial Area of Lesser Classifiers)**.** *Let $\mathcal{F}_\tau$ be a feasible classifier and let t be a fractional cost parameter. The **partial area of lesser classifiers**, $A^*_t(\mathcal{F}_\tau)$ is the area of the portion of the feasible region of ROC space consisting of $\mathcal{F}_\tau$'s lesser feasible classifiers using cost parameter t.*

By Definition 16 the partial area of lesser classifiers is cost monotone. Under some values of α and κ, feasible classifiers may occupy only a small region of ROC space. The numerical value of the partial area may thus be small even for the best feasible classifiers. For a metric whose value is easy to interpret as good or bad regardless of α, κ, we recommend a normalization:

Definition 18 (Normalized Partial Area)**.** *Let $\mathcal{F}_\tau$ be a feasible classifier and let t be a fractional cost parameter. The **normalized partial area** of lesser classifiers is the ratio of the partial area $A^*_t(\mathcal{F}_\tau)$ to the partial area of all feasible points in ROC space.*

$$A_t^*(h,k) = \begin{cases} \frac{1}{2}\left(A'(h+k) + \left(\frac{C'B'h}{C'-D'} - B'(h+k)\right)\frac{1}{C't-D'} + \left(\frac{B'h}{C'-D'} - A'h\right)\frac{1}{1-t}\right) \\ \frac{1}{2}\left(\frac{(1-\alpha)\kappa^2}{|\mathcal{P}||\mathcal{N}|} - \frac{A\kappa}{|\mathcal{P}|} + A(h+k) - B\left(\frac{\kappa}{|\mathcal{P}|} - (h+k) + \frac{Ch}{C+D}\right)\frac{1}{Ct+D} - \left(Ah + \frac{Bh}{C+D}\right)\frac{1}{1-t}\right) \\ \frac{1}{2}\left(\frac{(1-\alpha)\kappa}{|\mathcal{N}|} + \left(1 - \frac{\alpha\kappa}{|\mathcal{P}|}\right)\left(\frac{\kappa - |\mathcal{P}|}{|\mathcal{N}|}\right) + (1-h-k)^2 - \frac{(1-k)^2}{t} - \frac{h^2}{1-t}\right) \\ \frac{1}{2}\left(\frac{(1-\alpha)|\mathcal{P}|}{\alpha|\mathcal{N}|} + (1-h-k)^2 - \frac{(1-k)^2}{t} - \frac{h^2}{1-t}\right) \end{cases}$$

Figure 3: Formula for partial area of lesser classifiers for a given (h,k) location in ROC space and cost parameter $t \in [0,1]$. The 4 cases here depend on the geometry of the feasible region and the iso-performance line ℓ_t through (h,k). The constants A,B,C, D, A',B', C', and D' are explained in Lemma 19 and derived explicitly in Lemma C.4. The cases above are, in order, the areas of $v_{00}v_{\alpha t}v_{0t}$, $v_{00}v_{\alpha\kappa}v_{\kappa t}v_{0t}$, $v_{00}v_{\alpha\kappa}v_{\kappa 1}v_{t1}v_{0t}$, and $v_{00}v_{\alpha 1}v_{t1}v_{0t}$ respectively, using the vertex definitions in Def. 14. App. C identifies which case is needed for given inputs.

We can compute the denominator in this ratio directly using the area of the feasible region (see App. B.3). If the perfect classifier located at $(0,1)$ in ROC space is feasible, then this denominator is *equivalent* to the partial area of lesser classifiers for the perfect classifier.

Our notion of normalized partial area of lesser classifiers is similar in spirit to the ratio of relevant areas defined by Morasca and Lavazza (2020).

Lemma 19 (Form of the Partial Area)**.** *The partial area of lesser classifiers is a sum of quotients of linear functions of t as given in Figure 3.*

Proof. A full proof is in App. C. The area depends on how the iso-performance line ℓ_t through (h,k) intersects the feasible region. The key insight is that the x-coordinate of $\ell_t \cap \ell_\alpha$ can be expressed as $A' + \frac{B'}{C't-D'}$ and that of $\ell_t \cap \ell_\kappa$ as $A + \frac{B}{Ct+D}$, for suitable scalars $A, B, C, D, A', B', C', D'$. □

4.2 Partial Volume over the ROC surface

When stakeholders have a range or distribution of cost parameters t in mind as well as many possible thresholds τ, Ratigan and Cowen (2025) showed how integrating the area of lesser classifiers over that range of t leads to a performance metric called the volume over the ROC surface (VOROS). Importantly, at each t, they sensibly use the largest area over all thresholds τ. We now extend this idea to a *partial volume* that only considers the feasible region imposed by precision and capacity constraints.

Definition 20 (Partial VOROS or PV)**.** *Let $\mathcal{F}$ be a score-producing classifier, let $p(t)$ be a valid probability density function over [0,1] for the cost parameter t, and let α, κ define precision and capacity limits so that the feasible region is well-defined with area A^*. Then the* ***partial volume over the ROC surface*** *(partial VOROS) is the normalized partial area of lesser classifiers, averaged over the provided cost distribution:*

$$PV(\mathcal{F}) = \frac{1}{A^*}\int_{t=0}^{1} p(t) \max_\tau (A_t^*(\mathcal{F}_\tau))dt.$$

Here, the maximum is taken over the set of thresholds for $\mathcal{F}$ that produce distinct feasible points (h,k).

Overall, PV is a higher-is-better performance metric. Its range is between 0.0 and 1.0 regardless of the dataset or constraints α, κ. The best possible PV is 1.0, achieved by the highest feasible classifier on the y-axis. The worst value of 0.0 comes from the never-alarm baseline at (0,0), assuming Def. 11 holds.

PV can be viewed as an expectation over the distribution $p(t)$, $PV = \frac{1}{A_*}\mathbb{E}_{p(t)}[\max_\tau(A_t^*(\mathcal{F}_\tau))]$. Users could simply set $p(t)$ to a uniform distribution over the fractional cost t, perhaps confined to a task-relevant interval $[a,b] \subseteq [0,1]$. Stakeholders may prefer to define a distribution over cost ratio, $p(C_{\text{FP:FN}})$, which implies a distribution over t via Def. 2. The User Guide (App. A) gives practical advice on computing PV.

Runtime. The maximum in the definition is computable in $O(h_c)$-time from the convex hull of a ROC curve, where h_c is the number of feasible points in the curve's convex hull and thus the number of possible thresholds. See App. D for an algorithm. Recall that A_t^* is monotonically decreasing in Cost_t. We use this fact to speed up our calculation. Instead of directly assessing area $A_t^*(\mathcal{F}_\tau)$ at each of the h_c thresholds τ, we can find the threshold that minimizes Cost_t. Thus, only one area calculation is needed per t value.

PV and ROC Dominance. Similar to the VOROS and the traditional area under the ROC curve, the partial VOROS will always assign a higher value to a ROC curve that dominates.

Lemma 21 (Dominance)**.** *Let $y = f_1(x)$, $y = f_2(x)$ be two ROC curves. If for all x, $f_1(x) \geq f_2(x)$, then the VOROS and partial VOROS of f_1 will be higher than that of f_2.*

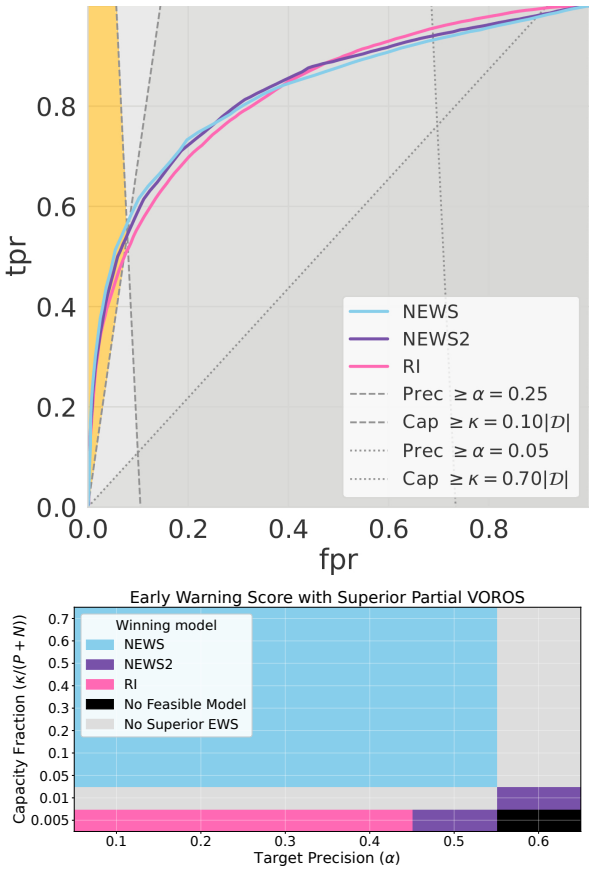

Figure 4: **Re-analysis of Early Warning Scores for Deterioration**, using ROC curves from Edelson et al.. Top: ROC curves with strict (– lines) and more lenient (:) constraints. Bottom: Heatmap of showing which model has the best PV score over possible α, κ limits.

Proof. Since the curve for $f_1(x)$ is above and to the left of the curve $f_2(x)$, for any fractional cost parameter t, the best performing point on $f_2(x)$ will be in the area of lesser classifiers of some point on $f_1(x)$. □

5 MODEL EVALUATION WITH PV

We examine partial VOROS as a post-hoc metric for existing clinical early warning scores. We re-analyze ROC curves published by Edelson et al. (2024), measuring the performance of widely-used risk scores on 362,926 patient stays in 7 hospitals in Connecticut, USA. The outcome of interest is deterioration, meaning transfer to the ICU or death within 24 hours. The prevalence p is 4.6% in Edelson et al.'s data.

For this case study, we focus on 3 scoring methods: NEWS (Royal College of Physicians, 2012), NEWS2 (Royal College of Physicians, 2017), and the Rothman Index (RI, Rothman et al. (2013)). We visualize ROC curves in Fig. 4: each curve crosses others, and it is thus imperative to consider costs and constraints to determine the best operating point of all methods. We consider capacity κ from $0.5\% - 70\%$ of $|\mathcal{D}|$ and precision limits $\alpha \in (0.1, 0.6)$. When calculating partial VOROS, we assume a uniform distribution over cost ratios $C_{\text{FP:FN}}$ that map to $t \in (0.01, 0.7)$, so that Def. 11 is satisfied.

Over a range of α, κ limits, we plot a heatmap in Fig. 4 indicating where different scoring methods have clear wins and regions where they are indistinguishable (normalized PV is within 0.01). We find that NEWS2 preforms best at high α and low capacity, RI has superior recall at lower α and low capacity, and NEWS performs best once capacity is greater than 5% of all examples. Edelson et al. (2024) previously concluded that NEWS "outperformed" the two other scores here. Our analysis with partial VOROS adds nuance: Each method could be a clear winner depending on precision and capacity limits.

6 MODEL SELECTION WITH PV

We now illustrate how our PV metric can be used for model selection, not just post-hoc evaluation. We will show how different performance metrics can lead to distinct selections for the best ROC curve and best threshold on that curve on a healthcare modeling task. We focus on in-hospital mortality prediction with the MIMIC-IV (Johnson et al., 2023) and eICU (Pollard et al., 2018) datasets. We aim for competitive *reproducible* classifiers, not state-of-the-art on past benchmarks. We'll assess how cost-aware selection strategies like our PV perform against cost-unaware alternatives, in terms of the ultimate cost on test data.

Task Description. For the MIMIC-IV task, we follow Rath and Hughes. At each ICU patient-stay we observe basic facts at admission (age, insurance type, weight, gender) as well as time-varying signals over the first 48 hours (6 vitals and 7 labs). The prediction task is to identify if the patient will die during the rest of their hospital stay. We use an open-source pipeline (Gupta et al., 2022) to obtain a cohort of patient-stays, each represented as a 0-48 hr time series with measured values every 2 hours. We keep only stays produced by Gupta et al.'s code with at least 3 vitals or labs measured in the last 16 hours. The eICU data task is also mortality prediction after 48 hours. Detailed setups for both tasks are in App. E.

For MIMIC-IV, our train/valid./test sets contain 15474/7802/7861 patient-stays with prevalence p of .104/.103/.107. This split was done by patient-id in a label-stratified way. We favor larger test sets than usual to be sure we can measure precision well despite low prevalence. For eICU, we have 38532/19267/19267

ICU-stays with prevalence p of 0.116/0.114/0.114.

Featurization. For MIMIC-IV, for each of 13 time-varying univariate channels, we extract features that represent 7 summary functions (was ever measured, time since last measurement, mean, variance, min, median, max, slope) over 3 windows (0-48 hours, 24-48 hours, 32-48 hours). Missing values in extracted features are filled with the population mean, then rescaled to the 0.0-1.0 range. For eICU, we use similar processing of the features provided by Tang et al. (2021).

Classifiers. We examine logistic regression, multi-layer perceptron, and random forests, using sklearn (Pedregosa et al., 2011). For each, we conduct an extensive grid search of 24-80 hyperparameter configurations to avoid overfitting (details in App. E).

Scenarios for Costs and Constraints. We consider two scenarios for costs and constraints. In each one, we fix particular α, κ bounds, and define a distribution over the cost ratio $C_{\text{FP:FN}}$, which implies a distribution over t via the invertible mapping in Def. 2.

1. The first scenario is relatively permissive. We fix $\alpha = 0.15, \kappa = 0.5|\mathcal{D}|$, and model the cost ratio as $C_{\text{FP:FN}} \sim \text{Unif}(\frac{1}{9}, \frac{1}{6})$. Here, false negatives are more costly than false positives, but by less than 10x. On MIMIC-IV, this implies a non-uniform density for $t \in (0.5, 0.6)$ for computing PV.
2. The second scenario is more constrained. We set $\alpha = 0.5, \kappa = 0.1|\mathcal{D}|$ to reflect more stringent clinical priorities for high precision and low total alarms. Recent economic estimates (Rogers et al., 2023) suggest a missed clinical deterioration may have roughly 20-40 times the cost of a false alarm, so we use $C_{\text{FP:FN}} \sim \text{Unif}(\frac{1}{40}, \frac{1}{20})$. On MIMIC-IV data, this implies a non-uniform density for $t \in (.18, .31)$.

Evaluation Plan. We compare 4 possible selection strategies: max pAUROC and max recall in the feasible region (α, κ-aware; t-unaware), max VOROS (α, κ-unaware; t-aware), and max PV (aware of both constraints and costs). For each strategy, we search a grid of over 140 ROC curves each from a different classifier-hyperparameter combination. For each possible strategy, we select one curve, and then a threshold τ to use at each t, always using the validation set for selection decisions (details in App. E). We then evaluate and report the test set performance of each selected threshold. In this way, we examine how binary alert systems behave on new data, avoiding unrealistic post-hoc threshold selection on the test set.

Results. For MIMIC-IV, Tab. 1 reports the test cost of each strategy across the two scenarios. These are the costs *in expectation* over a uniform distribution of cost ratios $C_{\text{FN:FP}}$. eICU cost results are in Tab. 2. Further results are in App. E, including visuals of selected ROC curves, expanded tables, and details of the calculation of expected cost.

	scenario 1 $\alpha = 0.15, \kappa = 0.5\|\mathcal{D}\|$ $\frac{C_{\text{FP}}}{C_{\text{FN}}} \sim \text{Unif}(\frac{1}{9}, \frac{1}{6})$ $t \in [.5, .6]$	scenario 2 $\alpha = 0.5, \kappa = 0.1\|\mathcal{D}\|$ $\frac{C_{\text{FP}}}{C_{\text{FN}}} \sim \text{Unif}(\frac{1}{40}, \frac{1}{20})$ $t \in [.18, .31]$
Strategy	Avg. Cost	Avg. Cost
max VOROS	**0.261**	0.640
max pAUROC	0.303	**0.538**
max recall	0.302	**0.538**
max PV (ours)	**0.261**	**0.538**

Table 1: **MIMIC-IV results for model selection.** We report average costs on test (lower is better). All models satisfy the α, κ constraints on both validation and test. Further results in Tab. E.1.

	scenario 1 $\alpha = 0.15, \kappa = 0.5\|\mathcal{D}\|$ $\frac{C_{\text{FP}}}{C_{\text{FN}}} \sim \text{Unif}(\frac{1}{9}, \frac{1}{6})$			scenario 2 $\alpha = 0.5, \kappa = 0.1\|\mathcal{D}\|$ $\frac{C_{\text{FP}}}{C_{\text{FN}}} \sim \text{Unif}(\frac{1}{40}, \frac{1}{20})$		
		Prec $\geq \alpha$? Cap $\leq \kappa$?			Prec $\geq \alpha$? Cap $\leq \kappa$?	
Strategy	Avg. Cost on test	val.	test	Avg. Cost on test	val.	test
max VOROS	**0.319**	Y	Y	0.772	Y	Y
max pAUROC	0.336	Y	N	0.707	Y	N
max recall	0.336	Y	N	0.707	Y	N
max PV (ours)	**0.319**	Y	Y	0.707	Y	N

Table 2: **eICU results for model selection.** We report average costs on test (lower is better). Strategies and scenarios are the same as for MIMIC-IV in Tab. 1. All models satisfy constraints on validation, but some do not satisfy on test; further results in Tab. E.2.

Analysis. In the more relaxed scenario 1, our max partial VOROS (PV) strategy yields better (lower) normalized cost on test data than 3 alternatives on MIMIC-IV. It also ties for best cost on eICU. This highlights how using cost-unaware strategies like maximizing recall can be inadequate, even if they incorporate "hard cutoffs" for the constraints of interest.

The more stringent scenario 2 favors the lower left of ROC curves where there is less crossing; we find 3 of the 4 strategies (including PV) sensibly choose the same ROC curve and thus have similar costs. Only Ratigan and Cowen's original VOROS (α, κ-unaware) yields a model with worse cost performance in this scenario on both datasets, highlighting the inadequacy of *ignoring constraints* in model selection.

On MIMIC-IV, all selected models satisfy the α, κ constraints on test data. However, on eICU several models that satisfy α, κ on validation fall slightly short of satisfying the same constraints on test (e.g. precision of 0.44 instead of 0.5), due to statistical variation across subsets of the same size. These are marked with red highlighting in Tab. 2; detailed results in Tab. E.2.

7 RELATED WORK

For an overview of performance metrics see Hand (2012); for a focus on visuals, see (Prati et al., 2011). Steyerberg and Vergouwe (2014) provides advice for healthcare focused modeling, where beyond just assessing discrimination via ROC, calibration is also valuable.

A common summary of the ROC is the area under the curve (Bradley, 1997; Hand, 2012), known as AUROC or the concordance statistic. Many works recommend a *partial area* under the ROC by only integrating over some false positive rates (McClish, 1989; Jiang et al., 1996; Robin et al., 2011). Our partial VOROS can be seen as a 3D extension of the partial AUROC that focuses on a desired cost range and obeys both capacity and precision constraints. Other alternative partial area metrics seek better correspondence to concordance (Carrington et al., 2020) but do not account for capacity or precision.

Shao et al. (2024) extend ROC curves to cost-sensitive learning, seeking to make weighted area-under-curve training robust to train-to-test shifts in cost and class distribution. Their work does not address precision or capacity constraints, unlike our partial VOROS.

Particularly for hospital alert systems, some works recommend *precision-recall* curves instead of (or in addition to) ROC curves (Saito and Rehmsmeier, 2015; Romero-Brufau et al., 2015; Martin et al., 2025). However, claims that the PR curve or the area under it (AUPRC) is somehow superior to the ROC/AUROC for imbalanced data have been recently refuted (Richardson et al., 2024; McDermott et al., 2024). For more on PR curves, see Flach and Kull (2015).

Synergistically with our work on a new performance metric, other work has sought to train models directly to perform well on certain metrics (Tsoi et al., 2022; Eban et al., 2017). Some of these directly optimize for recall at a precision constraint (Rath and Hughes, 2022; Fathony and Kolter, 2020; Peng et al., 2025) or the area under the PR curve (Ramzi et al., 2021). It may be possible to optimize for multi-objective criteria, such as maximizing recall and precision while minimizing capacity (and possibly other objectives) using a weighted sum of losses. This might be disfavored because it presents additional challenges like how to choose the weights on each objective term. Our approach that enforces hard constraints on precision or capacity better reflects the realities of intended applications: as long as the precision is high enough, cost alone rather than "cost + precision" should be the driving objective.

8 CONCLUSION

We have developed the **partial volume over the ROC surface**, aka the partial VOROS or PV metric, as a performance measure for binary classifiers that accounts for cost-imbalance, precision constraints, and capacity constraints. Our work represents careful geometric reasoning about tradeoffs between recall, precision, and false alarm rates. Our experiments on real health records show how partial VOROS can help evaluate models and make model selection decisions that would meet necessary operational constraints and reduce costs when deployed.

Limitations. We focus on binary classification; future work would be needed to transfer these ideas to a many-class or multi-label setting. Computing this metric requires specific constraints α, κ to be specified as well as a distribution over costs, either explicitly via a distribution over fractional cost parameter t or implicitly via a distribution on the ratio $C_{\text{FP:FN}}$. It is likely challenging to elicit values for technical constraints from non-technical stakeholders, though some work suggests routes forward (Wu et al., 2008).

The PV metric is specialized to a particular class balance. Numerical PV values are not comparable across datasets with different prevalence. Naturally, due to the relationship between precision and prevalence, any performance metric that requires enforcing a precision constraint could not escape a sensitivity to prevalence.

Real applications often face *drift* in class balance over time (e.g. due to emerging diseases, patient population shifts, or changing care practices). Stakeholder needs can evolve too, requiring different α or κ values or different cost distributions over time. Care is needed to wield PV effectively in such cases.

Even without drift, ensuring constraint satisfaction on several datasets from the same source with the same prevalence can be challenging due to statistical variation, as Tab. 2 indicates. In practice, enforcing a slightly higher α or lower κ than strictly needed may better ensure generalization. Future work could examine *statistical guarantees* for constraint satisfaction.

Recommendations. To help practitioners compute our PV metric on their data, we provide a User Guide in App. A and an open source release of Python code (see GitHub link on page 1). We intend that PV can be used in at least two ways: (i) as a standalone diagnostic metric for model assessment by human inspection, and (ii) a metric for model selection. We provide recommendations for each use case in App. A.

Acknowledgements

The authors acknowledge support from the U.S. National Science Foundation (NSF) via two awards: IIS #2338962 and the DIAMONDS REU #2149871. A part of author MH's effort was also supported by the U.S. National Institutes of Health under award number R01HL180937. This paper's content is solely the responsibility of the authors and does not necessarily represent the official views of the NSF or NIH.

References

Abella Álvarez, A., Torrejón Pérez, I., Enciso Calderón, V., Hermosa Gelbard, C., Sicilia Urban, J. J., Ruiz Grinspan, M., García Ureña, M. Á., Salinas Gabiña, I., Mozo Martín, T., Calvo Herranz, E., Díaz Blázquez, M., and Gordo Vidal, F. (2013). ICU without walls project. Effect of the early detection of patients at risk. *Medicina Intensiva*, 37(1):12–18.

Albanowski, K., Burdick, K. J., Bonafide, C. P., Kleinpell, R., and Schlesinger, J. J. (2023). Ten Years Later, Alarm Fatigue Is Still a Safety Concern. *AACN Advanced Critical Care*, 34(3):189–197.

Bradley, A. P. (1997). The use of the area under the ROC curve in the evaluation of machine learning algorithms. *Pattern Recognition*, 30(7).

Carrington, A. M., Fieguth, P. W., Qazi, H., Holzinger, A., Chen, H. H., Mayr, F., and Manuel, D. G. (2020). A new concordant partial AUC and partial c statistic for imbalanced data in the evaluation of machine learning algorithms. *BMC Medical Informatics and Decision Making*, 20(1):4.

Chan, T. M. (1996). Optimal output-sensitive convex hull algorithms in two and three dimensions. *Discrete & computational geometry*, 16(4):361–368.

Cvach, M. (2012). Monitor Alarm Fatigue: An Integrative Review. *Biomedical Instrumentation & Technology*, 46(4):268–277.

Drummond, C. and Holte, R. C. (2000). Explicitly representing expected cost: An alternative to ROC representation. In *Proceedings of the sixth ACM SIGKDD International Conference on Knowledge Discovery and Data mining*, pages 198–207.

Eban, E., Schain, M., Mackey, A., Gordon, A., Rifkin, R., and Elidan, G. (2017). Scalable Learning of Non-Decomposable Objectives. In *Artificial Intelligence and Statistics*.

Edelson, D. P., Churpek, M. M., Carey, K. A., Lin, Z., Huang, C., Siner, J. M., Johnson, J., Krumholz, H. M., and Rhodes, D. J. (2024). Early Warning Scores With and Without Artificial Intelligence. *JAMA Network Open*, 7(10):e2438986.

Fathony, R. and Kolter, Z. (2020). AP-perf: Incorporating generic performance metrics in differentiable learning. In *International Conference on Artificial Intelligence and Statistics*. PMLR.

Flach, P. and Kull, M. (2015). Precision-Recall-Gain Curves: PR Analysis Done Right. In *Advances in Neural Information Processing Systems*.

Goldberger, A. L., Amaral, L., Glass, L., Hausdorff, J. M., Ivanov, P. C., Mark, R. G., Mietus, J. E., Moody, G. B., Peng, C.-K., and Stanley, H. E. (2000). Physiobank, physiotoolkit, and physionet: components of a new research resource for complex physiologic signals. *Circulation*, 101(23).

Guo, C., Pleiss, G., Sun, Y., and Weinberger, K. Q. (2017). On Calibration of Modern Neural Networks. In *International Conference on Machine Learning*.

Gupta, M., Gallamoza, B., Cutrona, N., Dhakal, P., Poulain, R., and Beheshti, R. (2022). An extensive data processing pipeline for MIMIC-IV. In *Machine Learning for Health (ML4H)*. PMLR.

Hand, D. J. (2012). Assessing the performance of classification methods. *International Statistical Review*, 80(3):400–414.

Hardt, M., Price, E., and Srebro, N. (2016). Equality of Opportunity in Supervised Learning. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Harrison, A. M., Herasevich, V., and Gajic, O. (2015). Automated Sepsis Detection, Alert, and Clinical Decision Support: Act on It or Silence the Alarm? *Critical Care Medicine*, 43(8):1776.

Hyland, S. L., Faltys, M., Hüser, M., Lyu, X., Gumbsch, T., Esteban, C., Bock, C., Horn, M., Moor, M., Rieck, B., Zimmermann, M., Bodenham, D., Borgwardt, K., Rätsch, G., and Merz, T. M. (2020). Early prediction of circulatory failure in the intensive care unit using machine learning. *Nature Medicine*, 26(3):364–373.

Jiang, Y., Metz, C. E., and Nishikawa, R. M. (1996). A receiver operating characteristic partial area index for highly sensitive diagnostic tests. *Radiology*, 201(3):745–750.

Johnson, A. E., Bulgarelli, L., Shen, L., Gayles, A., Shammout, A., Horng, S., Pollard, T. J., Hao, S., Moody, B., Gow, B., et al. (2023). MIMIC-IV, a freely accessible electronic health record dataset. *Scientific data*, 10(1):1.

Kahn, J. M., Yabes, J. G., Bukowski, L. A., and Davis, B. S. (2023). Intensivist physician-to-patient ratios and mortality in the intensive care unit. *Intensive Care Medicine*, 49(5):545–553.

Lee, Y. and Lim, W. (2017). Shoelace Formula: Connecting the Area of a Polygon and the Vector Cross Product. *The Mathematics Teacher*, 110(8):631–636.

Martin, B., Bennett, T. D., DeWitt, P. E., Russell, S., and Sanchez-Pinto, L. N. (2025). Use of the Area Under the Precision-Recall Curve to Evaluate Prediction Models of Rare Critical Illness Events. *Pediatric Critical Care Medicine*, 26(6):e855–e859.

McClish, D. K. (1989). Analyzing a portion of the ROC curve. *Medical Decision Making: An International Journal of the Society for Medical Decision Making*, 9(3):190–195.

McDermott, M. B., Zhang, H., Hansen, L. H., Angelotti, G., and Gallifant, J. (2024). A closer look at AUROC and AUPRC under class imbalance. In *Advances in Neural Information Processing Systems*.

Morasca, S. and Lavazza, L. (2020). On the assessment of software defect prediction models via ROC curves. *Empirical Software Engineering*, 25(5).

Muralitharan, S., Nelson, W., Di, S., McGillion, M., Devereaux, P. J., Barr, N. G., and Petch, J. (2021). Machine Learning–Based Early Warning Systems for Clinical Deterioration: Systematic Scoping Review. *Journal of Medical Internet Research*, 23(2):e25187.

Park, S.-H., Kang, J., Kim, T. J., Lee, H. Y., Cho, H.-J., and Lee, S.-M. (2022). Current status of the rapid response system and early warning score: A survey-based analysis. *Acute and Critical Care*, 37(4):687–689.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, É. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Peng, L., Travadi, Y., He, C., Cui, Y., and Sun, J. (2025). Exact Reformulation and Optimization for Direct Metric Optimization in Binary Imbalanced Classification. *Preprint*, arXiv:2507.15240.

Pessach, D. and Shmueli, E. (2022). A review on fairness in machine learning. *ACM Comput. Surv.*, 55(3).

Pollard, T., Johnson, A., Raffa, J., Celi, L. A., Badawi, O., and Mark, R. (2019). eICU Collaborative Research Database. *PhysioNet*. Version 2.0.

Pollard, T. J., Johnson, A. E. W., Raffa, J. D., Celi, L. A., Mark, R. G., and Badawi, O. (2018). The eICU Collaborative Research Database, a freely available multi-center database for critical care research. *Scientific Data*, 5(1).

Prati, R. C., Batista, G. E. A. P. A., and Monard, M. C. (2011). A Survey on Graphical Methods for Classification Predictive Performance Evaluation. *IEEE Transactions on Knowledge and Data Engineering*, 23(11):1601–1618.

Provost, F. and Fawcett, T. (1997). Analysis and visualization of classifier performance with nonuniform class and cost distributions. In *AAAI-97 Workshop on AI Approaches to Fraud Detection & Risk Management*, pages 57–63.

Provost, F. and Fawcett, T. (2001). Robust classification for imprecise environments. *Machine learning*, 42:203–231.

Ramzi, E., Thome, N., Rambour, C., Audebert, N., and Bitot, X. (2021). Robust and decomposable average precision for image retrieval. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Rath, P. and Hughes, M. (2022). Optimizing early warning classifiers to control false alarms via a minimum precision constraint. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*.

Ratigan, C. and Cowen, L. (2025). The VOROS: Lifting ROC curves to 3D to summarize unbalanced classifier performance. In *AAAI Conference on Artificial Intelligence*.

Richardson, E., Trevizani, R., Greenbaum, J. A., Carter, H., Nielsen, M., and Peters, B. (2024). The receiver operating characteristic curve accurately assesses imbalanced datasets. *Patterns*, 5(6).

Robin, X., Turck, N., Hainard, A., Tiberti, N., Lisacek, F., Sanchez, J.-C., and Müller, M. (2011). pROC: An open-source package for R and S+ to analyze and compare ROC curves. *BMC bioinformatics*, 12:77.

Rogers, P., Boussina, A. E., Shashikumar, S. P., Wardi, G., Longhurst, C. A., and Nemati, S. (2023). Optimizing the Implementation of Clinical Predictive Models to Minimize National Costs: Sepsis Case Study. *Journal of Medical Internet Research*, 25:e43486.

Romero-Brufau, S., Huddleston, J. M., Escobar, G. J., and Liebow, M. (2015). Why the C-statistic is not informative to evaluate early warning scores and what metrics to use. *Critical Care*, 19(1).

Rothman, M. J., Rothman, S. I., and Beals, J. (2013). Development and validation of a continuous measure of patient condition using the Electronic Medical Record. *Journal of Biomedical Informatics*, 46(5):837–848.

Royal College of Physicians (2012). National Early Warning Score (NEWS): Standardising the assess-

ment of acute-illness severity in the NHS. Report of a working party, RCP, London.

Royal College of Physicians (2017). National Early Warning Score (NEWS) 2: Standardising the assessment of acute-illness severity in the NHS. Updated report of a working party, RCP, London.

Saito, T. and Rehmsmeier, M. (2015). The Precision-Recall Plot Is More Informative than the ROC Plot When Evaluating Binary Classifiers on Imbalanced Datasets. *PLoS ONE*, 10(3).

Sendak, M. P., Ratliff, W., Sarro, D., Alderton, E., Futoma, J., Gao, M., Nichols, M., Revoir, M., Yashar, F., Miller, C., Kester, K., Sandhu, S., Corey, K., Brajer, N., Tan, C., Lin, A., Brown, T., Engelbosch, S., Anstrom, K., Elish, M. C., Heller, K., Donohoe, R., Theiling, J., Poon, E., Balu, S., Bedoya, A., and O'Brien, C. (2020). Real-World Integration of a Sepsis Deep Learning Technology Into Routine Clinical Care: Implementation Study. *JMIR Medical Informatics*, 8(7).

Sendelbach, S. and Funk, M. (2013). Alarm fatigue: A patient safety concern. *AACN advanced critical care*, 24(4):378–386.

Shao, H., Xu, Q., Yang, Z., Wen, P., Peifeng, G., and Huang, Q. (2024). Weighted ROC curve in cost space: Extending AUC to cost-sensitive learning. In *Advances in Neural Information Processing Systems*.

Steyerberg, E. W. and Vergouwe, Y. (2014). Towards better clinical prediction models: Seven steps for development and an ABCD for validation. *European Heart Journal*, 35(29):1925–1931.

Tang, S., Davarmanesh, P., Song, Y., Koutra, D., Sjoding, M., and Wiens, J. (2021). MIMIC-III and eICU-CRD: Feature Representation by FIDDLE Preprocessing. *PhysioNet*. Version 1.0.0.

Tang, S., Davarmanesh, P., Song, Y., Koutra, D., Sjoding, M. W., and Wiens, J. (2020). Democratizing EHR analyses with FIDDLE: a flexible data-driven preprocessing pipeline for structured clinical data. *Journal of the American Medical Informatics Association*, 27(12):1921–1934.

Tsoi, N., Candon, K., Li, D., Milkessa, Y., and Vázquez, M. (2022). Bridging the gap: Unifying the training and evaluation of neural network binary classifiers. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Vaicenavicius, J., Widmann, D., Andersson, C., and Lindsten, F. (2019). Evaluating model calibration in classification. In *Artificial Intelligence and Statistics (AISTATS)*.

Wang, S., McDermott, M. B. A., Chauhan, G., Ghassemi, M., Hughes, M. C., and Naumann, T. (2020). MIMIC-Extract: A data extraction, preprocessing, and representation pipeline for MIMIC-III. In *Proceedings of the ACM Conference on Health, Inference, and Learning (CHIL)*.

Wu, T., Huang, H., Du, G., and Sun, Y. (2008). A novel partial area index of receiver operating characteristic (ROC) curve. In *Medical Imaging 2008: Image Perception, Observer Performance, and Technology Assessment*, volume 6917, pages 82–89. SPIE.

Zwillinger, D. (2018). Ch. 4.7: Polygons. In *CRC Standard Mathematical Tables and Formulas*, Advances in Applied Mathematics, page 212. CRC Press, 33 edition.

Checklist

1. For all models and algorithms presented, check if you include:
   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. **Yes**
   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. **Yes**
   (c) (Optional) Source code, with specification of all dependencies, including external libraries. **Yes**, see link on page 1.
2. For any theoretical claim, check if you include:
   (a) Statements of the full set of assumptions of all theoretical results. **Yes, see Def. 11.**
   (b) Complete proofs of all theoretical results. **Yes**
   (c) Clear explanations of any assumptions. **Yes**, see text around Def. 11.
3. For all figures and tables that present empirical results, check if you include:
   (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). **Yes**, see App. E and released code.
   (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). **Yes**, see App. E.4.
   (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). **Yes**.
   (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). **Yes**, see App. E.4.
4. If you are using existing assets (e.g., code, data,

models) or curating/releasing new assets, check if you include:

(a) Citations of the creator If your work uses existing assets. **Yes.**

(b) The license information of the assets, if applicable. **Yes, see App. E.**

(c) New assets either in the supplemental material or as a URL, if applicable. **No new data is used. Code is shared via Github link on page 1.**

(d) Information about consent from data providers/curators. **N/A. Our use of datasets is consistent with their licenses.**

(e) Discussion of sensitive content if applicable, e.g., personally identifiable information or offensive content. **Not applicable, all of MIMIC-IV and eICU is deidentified.**

5. If you used crowdsourcing or conducted research with human subjects, check if you include:

(a) The full text of instructions given to participants and screenshots. **Not Applicable.**

(b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. **Not Applicable.**

(c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. **Not Applicable.**

A User Guide for Partial VOROS

In this section, we give a guide to effectively using the partial VOROS to evaluate classifier performance. The high-level outline is

- Step 1: Define scenario-specific settings: data settings $|\mathcal{P}|$, $|\mathcal{N}|$, constraints α, κ, and costs t.
- Step 2: Evaluate the partial VOROS.

A.1 Step 0: Identify goals for model development and evaluation.

Backing up, before calculating PV, we hope developers have a clear intention of what problem they want to solve. We intend that PV can be used in at least two ways: (i) as a standalone diagnostic metric for model assessment by humans, and (ii) a metric for model selection.

First, as a standalone diagnostic, we hope that a human model developer could look at the relative PV values of two models and gain understanding about which one is delivering better binary classifications under the assumed constraints about α, κ and cost distribution $p(t)$. PV has a well-defined numerical range and directionality (perfect models get 1.0, higher is better). In terms of single number metrics that summarize the possible confusion matrices of a classifier, we think PV should stand alone. We would suggest it be reported instead of alternative metrics like AUROC or AUPRC that assess discriminative quality via transformations of possible confusion matrices, but don't reflect cost or constraints. For example, showing PV and AUROC side-by-side doesn't help answer one question, but answers distinct questions with conflicting assumptions: AUROC does not reflect specific constraints or cost ranges, but PV does. That said, PV may still be complementary to visualizations of the ROC curve. Showing these side-by-side can still be illuminating, especially when the constraint lines are overlaid. PV would also still be complementary to other metrics that assess classifier performance beyond the confusion matrix, such as metrics for calibration (Guo et al., 2017; Vaicenavicius et al., 2019), group fairness (Hardt et al., 2016; Pessach and Shmueli, 2022), or net benefit (Steyerberg and Vergouwe, 2014).

Second, we do recommend that PV can be valuable in model selection. This requires several key settings to be known in advance, such as the deployment setting's constraints α, κ as well as a plausible cost range or distribution. We intend that Sec. 6 provides a demonstration of how the PV performance metric can inform model and threshold selection, ultimately leading to lower cost predictions. It is critical to ensure all selection decisions align with the intended deployment or "test" scenario. In particular, the validation set must match deployment's expected prevalence and data distribution as much as possible. Even under well-matched circumstances, a selected model that barely meets constraints on validation may not meet them on test due to statistical noise. It may be wise to set α slightly higher and κ slightly lower than strictly needed. Future work could examine generalization bounds for these constraints.

A.2 Step 1: Define Scenario-specific Settings

The first step is to define the values or distributions of various scalar quantities used to calculate the partial VOROS. These include:

1. **Define $\mathcal{P}$ and $\mathcal{N}$.** Determine the relative sizes of the positive-labeled set $\mathcal{P}$ and the negative-labeled set $\mathcal{N}$ which together comprise the overall dataset $\mathcal{D}$. These can be defined in absolute terms as $|\mathcal{P}|$ and $|\mathcal{N}|$, or in relative terms by knowing the total size $|\mathcal{D}|$ and the prevalence of the positive class $p = \frac{|\mathcal{P}|}{|\mathcal{D}|}$. It is important to remember that partial VOROS values are **only comparable on datasets of the same prevalence**.
2. **Define α.** The minimum precision constraint value α must be determined in consultation with appropriate stakeholders. This is a rate between 0.0 and 1.0. It represents the minimum allowable fraction of all alarms that must be correct for the system to be useful.
3. **Define κ.** The maximum capacity constraint value κ can be specified as an absolute value taking any non-negative number on the scale of 0.0 to $|\mathcal{D}|$. We could also specify a fractional capacity $\tilde{\kappa}$ between 0.0 and 1.0, such that $\kappa = \tilde{\kappa}|\mathcal{D}|$, so the fraction represents capacity relative to total data size. In some cases, it may be more natural to specify κ in absolute terms, such as when the clear limit is the total number of available staff. A hospital's capacity may be more readily interpreted as maximum number of alarms handled per day (or other unit of time), since a single hospital with finite staff might know exactly the number of alarms they can handle. However, different use cases may favor the fractional constraint. For example, an entire

hospital system might prefer to specify a ratio instead.

4. **Define cost.** The last two settings which must be specified, again in consultation with stakeholders, are the cost of a false positive $C_{\text{FP}} > 0$ and the cost of a false negative $C_{\text{FN}} > 0$. For the hospital alert use case, cost values should be specified relative to the usual standard of care. So the cost of a false negative is the cost of not giving additional immediate attention and treatment to a patient in need (but still visiting them as planned in rounds). Similarly, the cost of a false positive should cover both the cost of treatment when none was needed and the opportunity cost of additional time spent caring for the patient beyond standard of care.

   Cost as a Ratio. The absolute values of these costs are not important, instead what matters is the ratio: $C_{\text{FP:FN}} = \frac{C_{\text{FP}}}{C_{\text{FN}}}$. All absolute cost settings with the same ratio yield the same fractional cost t via Def. 2.

   Uncertain costs. A key strength of our PV approach is the ability to *average over uncertainty* about the fractional cost $t \in [0, 1]$ or the cost ratio $C_{\text{FP:FN}} > 0$, rather than force a specific value. Users can specify a valid distribution over t or directly over $C_{\text{FP:FN}}$. Using the one-to-one mapping between t and $C_{\text{FP}}, C_{\text{FN}}$ values in Def. 2, we can write the expected value definition of PV in terms of whichever distribution is convenient, as shown below.

$$
\begin{aligned}
PV(\mathcal{F}) &= \mathbb{E}_{p(t)}[\max_{\tau} A^*_t(\mathcal{F}_\tau)] \qquad (1)\\
&= \mathbb{E}_{p(C_{\text{FP:FN}})}[\max_{\tau} A^*_{t(C_{\text{FP:FN}})}(\mathcal{F}_\tau)], \qquad t(C_{\text{FP:FN}}) := \frac{C_{\text{FP:FN}}|\mathcal{N}|}{C_{\text{FP:FN}}|\mathcal{N}| + |\mathcal{P}|},
\end{aligned}
$$

In practice, a direct distribution over the fractional cost parameter t is complicated to select because it requires beliefs about $C_{\text{FP:FN}}$ as well as specific values related to the relative sizes of $\mathcal{P}$ and $\mathcal{N}$ in the dataset. Unless the composition of validation and test sets are exactly matched, the desired $p(t)$ will need to *change* even across validation and test sets from the same data source.

We thus recommend directly specifying a distribution over $C_{\text{FP:FN}}$ as both more natural to stakeholders and transportable across datasets with different prevalence. See our analysis of Scenario 1 and Scenario 2 in Sec. 6 for two practical versions of a $C_{\text{FP:FN}}$ distributions in our hospital alert application.

A.3 Step 2: Evaluate the Partial VOROS

Once all of the settings from Step 1 are appropriately specified, we can evaluate the partial VOROS metric.

Exact evaluation of the Partial VOROS is only possible when the chosen distribution over $C_{\text{FP:FN}}$ (or t) has a PDF we can readily evaluate. We use simple uniform distributions throughout our experiments. Alternatively, if sampling is possible then Monte Carlo estimation of the integral could be done.

To evaluate PV in practice, we first use the constants α, κ, $|\mathcal{P}|$, and $|N|$ to define the feasible region as a subset of ROC space (this is the intersection of six half spaces).

Then, we obtain a dense grid of $C_{\text{FP:FN}}$ values and map each element to its corresponding t value via Def. 2, yield a grid of t values $\{T_1, ..., T_N\}$. For each, T_i, pick the optimal allowed point on the ROC curve for $\mathcal{F}$. Note, such an optimal point can be found readily by looking at the slope of the observed ROC curve to the left, $m_{\text{left}}(x, f(x))$ and right $m_{\text{right}}(f, f(x))$ of any point $(x, f(x))$ on the curve in the feasible region, the optimal point will be the one where either $m_{\text{right}} \leq \frac{T_i}{1-T_i} \leq m_{\text{left}}$ or, if the optimal point on the curve lies outside the feasible region, the optimal point will be the rightmost feasible point on the curve. Each such point is associated a threshold τ_i for our classifier $\mathcal{F}$. When using the partial VOROS for model development and selection, the choice of this τ_i for the given T_i should be determined on the validation set and then that same τ_i used on the test set.

Once we have a threshold τ_i associated to each T_i, we can calculate the partial area of lesser classifiers at that point. This is the area of the polygon below the isoperformance line associated to τ_i in the feasible region. Our Python code performs the required geometry and uses the shoelace formula to evaluate the area.

We now have the required ingredients to evaluate the integral that defines PV. Our experiments use a numerical approximation via the trapezoid method with 1000 evenly-space points covering the support of the uniform distribution over $C_{\text{FP:FN}}$. This yields the partial VOROS for the given α, κ, $|\mathcal{P}|$ and $|\mathcal{N}|$.

Practitioners may consider task-specific ranges of the limit parameters α, κ. The experiments in Fig. 4 provide a reasonable guide. Remember that PV values can only be fairly compared for datasets with equal prevalence p.

B Cases for Feasible Region Polygons

In this section, we provide a detailed analysis of the feasible region as well as explicit formulas for the vertices and areas of the feasible region.

Using the notation of Definition 14, we have the following formulas for the potential vertices of the feasible region.

Lemma B.1 (Coordinates of Vertices)**.** *If $i, j \in \{0, 1\}$, then $v_{ij} = (i, j)$, also if $i = 0$, then $v_{i\alpha} = v_{\alpha i} = (0, 0)$. Otherwise, we have*

- $v_{\alpha\kappa} = \left(\frac{(1-\alpha)\kappa}{|\mathcal{N}|}, \frac{\alpha\kappa}{|\mathcal{P}|}\right)$
- $v_{\alpha 1} = \left(\frac{(1-\alpha)|\mathcal{P}|}{\alpha|\mathcal{N}|}, 1\right)$
- $v_{\kappa 1} = \left(\frac{\kappa - |\mathcal{P}|}{|\mathcal{N}|}, 1\right)$
- $v_{0\kappa} = \left(0, \frac{\kappa}{|\mathcal{P}|}\right)$
- $v_{\kappa 0} = \left(\frac{\kappa}{|\mathcal{N}|}, 0\right)$
- $v_{1\kappa} = \left(1, \frac{\kappa - |\mathcal{N}|}{|\mathcal{P}|}\right)$
- $v_{1\alpha} = \left(1, \frac{\alpha|\mathcal{N}|}{(1-\alpha)|\mathcal{P}|}\right)$

This lemma follows directly from simple algebra using the definition.

In this section, we explain how to handle cases that don't fit into Definition 11.

Note: since ℓ_α passes through the origin, we have $v_{00} = v_{\alpha 0} = v_{0\alpha} = (0, 0)$ for all α, so there are really only 7 points of interest for our analysis.

With a precision $\alpha \geq 0$ and maximum capacity $\kappa \geq 0$, there are 13 cases for our feasible region, we break these into 8 degenerate and 5 nondegenerate cases, where the 3 cases that satisfy Definition 11 are 3 of the 5 non-degenerate cases, and the other two non-degenerate cases violate Definition 11 because $\alpha < p$. Let ℓ_α denote the precision line and ℓ_κ denote the capacity line.

B.1 Degenerate Cases

First, note that if $\alpha = 1$ or $\kappa = 0$, then the feasible region is merely the point $(0, 1)$ or $(0, 0)$ respectively. Also, the feasible region is empty if $\alpha = 1$ and $\kappa \leq |\mathcal{D}|$.

If $\alpha = 0$ or $\kappa \geq |\mathcal{D}|$, we consider the feasible region to be degenerate as only one of the two bounds intersects the interior of ROC space.

The degenerate cases depend on how ℓ_α or ℓ_κ intersect ROC space. If all we have is a minimum precision bound, there are two possibilities. $v_{\alpha 1}$ borders ROC space, or $v_{1\alpha}$ borders ROC space. Note the first case is equivalent to nondegenerate case 1 below. Similarly, if all we have is a capacity bound, there are four cases depending on which pair of $v_{\kappa 0}, v_{\kappa 1}, v_{0\kappa}$ and $v_{1\kappa}$ border ROC space (note $v_{0\kappa}$ and $v_{\kappa 1}$ cannot both border ROC space since ℓ_κ has nonpositive slope).

B.2 Nondegenerate cases

Assume that $\alpha \in (0, 1)$ and $\kappa \in (0, |\mathcal{D}|)$, then we have five cases.

Case 1: $v_{\alpha\kappa}$ lies inside of ROC space.

This also splits into 2 subcases depending on whether v_{01} is feasible according to ℓ_κ.

Case 1A: If v_{01} is not feasible then the feasible region is the triangle $\triangle v_{00}v_{\alpha\kappa}v_{0\kappa}$. This is Case 1 in the main paper.

Case 1B: If v_{01} is feasible, then the feasible region is the quadrilateral $v_{00}v_{\alpha\kappa}v_{\kappa 1}v_{01}$. This is Case 2 in the main paper.

Case 2: $v_{\alpha\kappa}$ lies above $y = 1$. (This is Case 3 in the main paper)

In this case, the feasible region is simply the triangle $\triangle v_{00}v_{\alpha 1}v_{01}$.

Case 3: $v_{\alpha\kappa}$ lies to the right of $x = 1$. (Not in the main paper, violates Definition 11 since $\alpha < p$)

This splits into 2 subcases depending on whether v_{01} is feasible according to ℓ_κ.

Case 3A: if v_{01} is feasible, then the feasible region is the pentagon $v_{00}, v_{1\alpha}v_{1\kappa}v_{\kappa 1}v_{01}$

Case 3B: if v_{01} is not feasible, then the feasible region is the trapezoid $v_{00}v_{1\alpha}v_{1\kappa}v_{0\kappa}$.

B.3 Geometry of Polygon Cases Satisfying Definition 11

In this section we detail the form and area of the feasible region using the notation of Definition 14 subject to the cases in Section 3.4 (and also denoted Cases A, C1 and C2 above):

Lemma B.2 (Vertices for Feasible Regions)**.** *Counterclockwise from the origin, the bounding vertices of the Case 1 Triangle are*

- $v_{00} = (0, 0)$
- $v_{\alpha\kappa} = (\frac{(1-\alpha)\kappa}{|\mathcal{N}|}, \frac{\alpha\kappa}{|\mathcal{P}|})$
- $v_{0\kappa} = (0, \frac{\kappa}{|\mathcal{P}|})$

Counterclockwise from the origin, the bounding vertices of the Case 2 Quadrilateral are

1. $v_{00} = (0, 0)$
2. $v_{\alpha\kappa} = \left(\dfrac{(1-\alpha)\kappa}{|\mathcal{N}|}, \dfrac{\alpha\kappa}{|\mathcal{P}|}\right)$
3. $v_{\kappa 1} = \left(\dfrac{\kappa - |\mathcal{P}|}{|\mathcal{N}|}, 1\right)$
4. $v_{01} = (0, 1)$

Counterclockwise from the origin, the bounding vertices of the Case 3 triangle are

1. $v_{00} = (0, 0)$
2. $v_{\alpha 1} = \left(\dfrac{(1-\alpha)|\mathcal{P}|}{\alpha|\mathcal{N}|}, 1\right)$
3. $v_{01} = (0, 1)$

Each of these vertices can be found by simple algebra.

In order to decide which of the 3 cases to use in the area formula given by C.4, we can use the following lemma.

Lemma B.3 (Area of Feasible Region)**.** *In general, the area of the feasible region in ROC space is*

$$A^* = \begin{cases} \dfrac{(1-\alpha)\kappa^2}{2|\mathcal{N}||\mathcal{P}|} & \textit{case 1} \\ \dfrac{2\kappa|\mathcal{P}| - \alpha\kappa^2 - |\mathcal{P}|^2}{2|\mathcal{N}|\mathcal{P}|} & \textit{case 2} \\ \dfrac{(1-\alpha)|\mathcal{P}|}{\alpha|\mathcal{N}|} & \textit{case 3} \end{cases}$$

Proof. This follows by applying the well-known "shoelace" formula (Lee and Lim, 2017; Zwillinger, 2018) for the area of a polygon applied to the boundary vertices defined above. □

This lemma shows that the area of the feasible region can be calculated precisely from the constraints defining the problem and dataset. Alternatively, since the feasible region consists of the convex hull of 6 lines in the plane, software can calculate the area of the convex hull of the collection of 6 lines $x = 0$, $x = 1$, $y = 0$, $y = 1$, ℓ_κ, and ℓ_α, quite quickly (see Section D below).

C Cases for Partial Areas

Given that we are in one of the three cases from Sec. 3.4, there are four cases for the region of lesser classifiers.

Assuming the practical assumptions in Definition 11 we have the following Lemma.

Lemma C.1. *The isoperformance line through the baseline of never alarming does not intersect the interior of the feasible region iff* $t \leq \dfrac{\alpha|\mathcal{N}|}{\alpha|\mathcal{N}| + (1-\alpha)|\mathcal{P}|}$.

Proof. The isoperformance line through $(0,0)$ is $y = \dfrac{t}{1-t}x$ which lies below ℓ_α in the first quadrant precisely when $t < \dfrac{\alpha|\mathcal{N}|}{\alpha|\mathcal{N}| + (1-\alpha)|\mathcal{P}|}$. If equality holds, the lines coincide, but ℓ_t still misses the interior. □

This lemma means that the baseline of never alarming has the highest cost of any point in our feasible region, it also ensures that the partial VOROS of this baseline is 0.

Using the notation of definition 14 we have

Lemma C.2 (Coordinates for Iso-performance Vertices.)**.** *We have the following*

- $v_{0t} = (0, k - \frac{t}{1-t}h)$.
- $v_{t1} = (\frac{(1-t)(1-k)}{t} + h, 1)$
- $v_{\alpha t}$ *has* x*-coordinate given by* $A' + \dfrac{B'}{C't - D'}$ *where*

$$\begin{aligned} A' &= \frac{(1-\alpha)|\mathcal{P}|(h+k)}{\alpha|\mathcal{N}| + (1-\alpha)|\mathcal{P}|} \\ B' &= (1-\alpha)|\mathcal{P}|\left(\frac{\alpha|\mathcal{N}|h - (1-\alpha)|\mathcal{P}|k}{\alpha|\mathcal{N}| + (1-\alpha)|\mathcal{P}|}\right) \\ C' &= \alpha|\mathcal{N}| + (1-\alpha)|\mathcal{P}| \\ D' &= \alpha|\mathcal{N}| \end{aligned}$$

- $v_{\kappa t}$ *has* x*-coordinate* $A + \dfrac{B}{Ct + D}$ *where*

$$\begin{aligned} A &= \frac{|\mathcal{P}|(h+k) - \kappa}{|\mathcal{P}| - |\mathcal{N}|} \\ B &= \kappa - k|\mathcal{P}| - \frac{|\mathcal{N}|(|\mathcal{P}|(h+k) - \kappa)}{|\mathcal{P} - |\mathcal{N}|} \\ C &= |\mathcal{P}| - |\mathcal{N}| \\ D &= |\mathcal{N}| \end{aligned}$$

Proof. The proof follows from taking the intersection of the lines ℓ_t, ℓ_α, ℓ_κ, $y = 1$ and $x = 0$ as needed. The rational linear t-terms in the x-coordinates of $v_{\alpha t}$ and $v_{\kappa t}$ follow from the fact that the coefficient of x in ℓ_t is rational in t. □

Lemma C.3. *The region of lesser classifiers for a feasible* (h, k) *is always one of:*

- *The triangle* $v_{00}v_{\alpha t}v_{0t}$.
- *The quadrilateral* $v_{00}v_{\alpha\kappa}v_{\kappa t}v_{0t}$

- *The pentagon* $v_{00}v_{\alpha\kappa}v_{\kappa 1}v_{t1}v_{0t}$.
- *The quadrilateral* $v_{00}v_{\alpha 1}v_{t1}v_{0t}$.

Where, the triangle $v_{00}v_{\alpha t}v_{0t}$ *can apply to any of the three cases in section 3., so long as* ℓ_t *intersects* ℓ_α *below* ℓ_κ *and below* $y = 1$. *The quadrilateral* $v_{00}v_{\alpha\kappa}v_{\kappa t}v_{0t}$, *can only apply to cases 1 and 2 from Sec. 3.4 since* ℓ_t *needs to intersect* ℓ_κ *on the border of the feasible region. The pentagon only applies when* ℓ_t *intersects* $y = 1$ *on the border of the feasible region in case* 2 *from Sec. 3.4. Finally, the quadrilateral* $v_{00}v_{\alpha 1}v_{t1}v_{0t}$ *only applies when* ℓ_t *intersects* $y = 1$ *on the border of the feasible region in case* 3.

Proof. Since we are in one of the three cases from Sec. 3.4, note that the isoperformance line will always leave the feasible region on the left through the y-axis by the last part of Definition 11, so v_{0t} is a vertex of all the regions.

The remaining possible regions depend on how the line leaves to the right and which points are included in the polygon. Specifically,

1. if we are in Case 1, the region of lesser classifiers is the triangle $v_{00}v_{\alpha t}v_{0t}$ if ℓ_t lies on or below $v_{\alpha\kappa}$, and is the quadrilateral $v_{00}v_{\alpha\kappa}v_{\kappa t}v_{0t}$ otherwise.
2. in Case 2, the region of lesser classifiers is the triangle $v_{00}v_{\alpha t}v_{0t}$ if ℓ_t lies on or below $v_{\alpha\kappa}$, the quadrilateral $v_{00}v_{\alpha\kappa}v_{\kappa t}v_{0t}$ if ℓ_t lies between $v_{\alpha\kappa}$ and $v_{\kappa 1}$, or the pentagon $v_{00}v_{\alpha\kappa}v_{\kappa 1}v_{t1}v_{0t}$ if ℓ_t lies above $v_{\kappa 1}$.
3. in Case 3, the region of lesser classifiers is the triangle $v_{00}v_{\alpha t}v_{0t}$ if ℓ_t lies on or below $v_{\alpha 1}$, or the quadrilateral $v_{00}v_{\alpha 1}v_{t1}v_{0t}$ otherwise.

□

It is worth noting that whether ℓ_t lies above or below any of these vertices can be readily checked by comparing the cost of (h, k) to that of the boundary point (higher cost points lie below lower costs ones), so this combined with Lemma B.2 yields a simple algorithm for determining the form of the partial area.

Lemma C.4 (Calculating Partial Area)**.** *Let* $t \in [0, 1]$ *be fixed, and let* (h, k) *be the ROC coordinates of an allowed classifier, then the (non-normalized) partial area of lesser classifiers is given by the formula in Figure 3.*

The formula for the first case relies on the x *coordinate of* $v_{\alpha t}$, $x = A' + \dfrac{B'}{C't - D'}$, *where we define*

$$\begin{aligned}
A' &= \frac{(1-\alpha)|\mathcal{P}|(h+k)}{\alpha|\mathcal{N}| + (1-\alpha)|\mathcal{P}|} \\
B' &= (1-\alpha)|\mathcal{P}|\left(\frac{(\alpha|\mathcal{N}|h - (1-\alpha)|\mathcal{P}|k}{\alpha|\mathcal{N}| + (1-\alpha)|\mathcal{P}|}\right) \\
C' &= \alpha|\mathcal{N}| + (1-\alpha)|\mathcal{P}| \\
D' &= \alpha|\mathcal{N}|.
\end{aligned}$$

The formula of the second case relies on the x *coordinate of* $v_{\kappa t}$, $x = A + \dfrac{B}{Ct + D}$, *where*

$$\begin{aligned}
A &= \frac{|\mathcal{P}|(h+k) - \kappa}{|\mathcal{P}| - |\mathcal{N}|} \\
B &= \kappa - k|\mathcal{P}| - \frac{(|\mathcal{P}|(h+k) - \kappa)|\mathcal{N}|}{|\mathcal{P}| - |\mathcal{N}|} \\
C &= |\mathcal{P}| - |\mathcal{N}| \\
D &= |\mathcal{N}|
\end{aligned}$$

Proof. The proof follows directly from applying the well-known determinant formula for a polygon. Each formula corresponds to a distinct polygon from the preceding lemma. Specifically:

- In the first formula, we are calculating the area of triangle $v_{00}v_{\alpha t}v_{0t}$.
- In the second formula, we are calculating the area of the quadrilateral $v_{00}v_{\alpha\kappa}v_{\kappa t}v_{0t}$.

- In the third formula, we are calculating the area of the pentagon, $v_{00}v_{\alpha\kappa}v_{\kappa 1}v_{t1}v_{0t}$.
- In the fourth formula we are calculating the area of the quadrilateral $v_{00}v_{\alpha 1}v_{t1}v_{0t}$

□

D Computational Complexity

Given a classifier $\mathcal{F}$, we can produce an ROC curve by taking all possible binarized classifiers $\mathcal{F}_\tau$ and plotting them in ROC space. To distinguish which $\mathcal{F}_\tau$ are potentially useful, we can take the convex hull of the curve together with the point $(1, 0)$ in $O(n\log(h))$ time, where h is the number of vertices in the Convex Hull using Chan's Algorithm (Chan, 1996).

Then, given the convex hull, we can associate to each feasible point a range of values of t for which that point will have the lowest cost on the curve and hence the highest area of lesser classifiers.

Let $\{(x_i, y_i)\}$ the points of the convex hull oriented clockwise from $(0, 0)$ to $(1, 1)$
if (x_{i+1}, y_{i+1}) is feasible **then**
  if $x_i = 0$ **then**
    if $x_{i+1} = 0$ **then**
      Skip.
    else
      (x_i, y_i) is optimal for $t \in [\frac{y_{i+1}-y_i}{x_{i+1}-x_i+y_{i+1}-y_i}, 1]$
    end if
  else
    if $y_i = 1$ **then**
      (x_i, y_i) is optimal for t in $\left[0, \frac{y_i-y_{i-1}}{x_i-x_{i+1}+y_i-y_{i+1}}\right]$; Break
    else
      (x_i, y_i) is optimal for t in $\left[\frac{y_{i+1}-y_i}{x_{i-1}-x_i+y_{i-1}-y_i}, \frac{y_i-y_{i-1}}{x_i-x_{i-1}+y_i-y_{i-1}}\right]$
    end if
  end if
else
  if (x_i, y_i) is feasible. **then**
    (x_i, y_i) is optimal for $t \in \left[0, \frac{y_i-y_{i-1}}{x_i-x_{i-1}+y_i-y_{i-1}}\right]$
    Break
  end if
end if

The only computations in this algorithm are the values of $\frac{y_{i+1}-y_i}{x_{i+1}-x_i+y_{i+1}-y_i}$ for all but the last feasible (x_i, y_i) and checks whether each (x_i, y_i) is feasible. Together, this takes order $n_{\alpha\kappa}$ time where $n_{\alpha\kappa}$ is the number of feasible points on the convex hull.

From here, we can use the formula for partial area in Figure 3 to calculate the areas and average them in $O(n_{\alpha\kappa})$ time.

Since the formulas in Figure 3 are linear combinations of quotients of linear functions of t, we can also integrate them directly getting a formula for the Volume associated to each t-range involving a logarithm. Computing this is again $O(n_{\alpha\kappa})$, since for a given t-range $[a, b]$, we can partition $[a, b]$ into subintervals $[a_i, b_i]$ such that the optimal feasible point is $(x_i, f(x_i))$ on the interval $[a_i, b_i]$. Thus the integral can be broken up into $n_{\alpha\kappa}$ integrals each of which can be evaluated in $O(1)$ time. The association of x_i's to $[a_i, b_i]$ is also $O(n_{\alpha\kappa})$.

This yields an overall computational complexity of $O(n_{\alpha\kappa} \log n_{\alpha\kappa}))$ if we need to compute the ROC convex hull. The partial ROC convex hull takes only $O(n_{\alpha\kappa} \log n_{\alpha\kappa})$ or $O(n_{\alpha\kappa})$ time since we can start with $(0, 0)$ and proceed right along the ROC curve until we hit the left boundary of the feasible region. If we already have the convex hull to begin with, we need only consider the points of the hull in the feasible region a determination that takes only $O(n_{\alpha\kappa})$ steps and from which the partial VOROS can be computed exactly in $O(n_{\alpha\kappa})$ steps.

Alternatively, we can approximate the partial VOROS by drawing a monte carlo sample of $p(t)$ and evaluating the area of lesser classifiers of our curve in $O(n_{\alpha\kappa})$, since we can find the optimal feasible point in an $O(\log(n_{\alpha\kappa})$ step binary search, and then for each sample we can calculate the area of lesser classifiers in $O(1)$ steps. These partial areas can then be averaged to give a good approximation of the partial VOROS.

VITALS

Vital Sign	`feat_name` in code
Blood pressure (diastolic)	bp_diastolic_mmHg
Blood pressure (systolic)	bp_systolic_mmHg
Heart rate	heart_rate
Oxygen saturation	oxygen_saturation
Respiratory Rate	resp_rate
Temperature	temp

LAB MEASUREMENTS

Lab Measurement	`feat_name` in code
Cholesterol	cholesterol
Glucose	glucose
Hemoglobin	hemoglobin
Lactic Acid	lactic_acid
pH	pH
platelets	platelets
white blood cell count	white_blood_cell_count

Table E.1: Summary of 6 vitals and 7 labs used in MIMIC-IV experiments

E Details on Experiments

E.1 License and availability

Our experiments in Sec. 6 use two open-access datasets available via PhysioNet (Goldberger et al., 2000) at https://physionet.org/. First, the MIMIC-IV dataset (Johnson et al., 2023) is freely available to qualified researchers subject to the PhysioNet Credentialed Health Data License 1.5.0. Second, we used version 2.0 of the eICU Collaborative Research Database (Pollard et al., 2018, 2019), which is also available under the PhysioNet Credentialed Health Data License 1.5.0.

E.2 MIMIC-IV Mortality Prediction: Features and Task Setup

For the mortality prediction experiments on MIMIC-IV data, we examine 6 vital signs, collected via bedside monitors and extracted from health records via the CHARTEVENTS table in MIMIC-IV. We further examine 7 laboratory measurements from extracted blood and other fluids, again from the CHARTEVENTS table in MIMIC-IV. See listing in Tab. E.1

These vitals and labs are extracted via best practices in *clinical grouping* of conceptually similar variables that have distinct ITEMID codes in the EHR. We use the groupings provided in Wang et al. (2020), given explicitly in Tab. E.2 for our variables of interest. Note that we extracted weight from the charts over time, but treated it as static (not dynamic) due to the limited 48 hour window.

itemid	label	unitname	feat_name
224643	Manual Blood Pressure Diastolic Left	mmHg	bp_diastolic_mmHg
225310	ART BP Diastolic	mmHg	bp_diastolic_mmHg
220180	Non Invasive Blood Pressure diastolic	mmHg	bp_diastolic_mmHg
220051	Arterial Blood Pressure diastolic	mmHg	bp_diastolic_mmHg
227243	Manual Blood Pressure Systolic Right	mmHg	bp_systolic_mmHg
224167	Manual Blood Pressure Systolic Left	mmHg	bp_systolic_mmHg
220179	Non Invasive Blood Pressure systolic	mmHg	bp_systolic_mmHg
225309	ART BP Systolic	mmHg	bp_systolic_mmHg
220050	Arterial Blood Pressure systolic	mmHg	bp_systolic_mmHg
220603	Cholesterol		cholesterol
220621	Glucose (serum)		glucose
226537	Glucose (whole blood)		glucose
220045	Heart Rate	bpm	heart_rate
226730	Height (cm)	cm	height
226707	Height	Inch	height
220228	Hemoglobin	g/dl	hemoglobin
225668	Lactic Acid		lactic_acid
220277	O2 saturation pulseoxymetry	%	oxygen_saturation
220227	Arterial O2 Saturation	%	oxygen_saturation
223830	PH (Arterial)		pH
220274	PH (Venous)		pH
227457	Platelet Count		platelets
224422	Spont RR	bpm	resp_rate
220210	Respiratory Rate	insp/min	resp_rate
224689	Respiratory Rate (spontaneous)	insp/min	resp_rate
224690	Respiratory Rate (Total)	insp/min	resp_rate
223762	Temperature Celsius	°C	temp
223761	Temperature Fahrenheit	°F	temp
224639	Daily Weight	kg	weight
226512	Admission Weight (Kg)	kg	weight
220546	WBC		white_blood_cell_count

Table E.2: Exact ITEMIDs extracted from CHARTEVENTS table in MIMIC-IV

E.3 eICU Mortality Prediction: Features and Task Setup

Task description. For each ICU-stay in the eICU dataset (Pollard et al., 2019), we collect static information upon admission and time-dependent health data for the first 48 hours. The classification task is to predict a patient's mortality during their stay—if the patient will survive (class 0) or die (class 1) after the first 48 hours. We use the preprocessed eICU dataset extracted by FIDDLE consisting of binarized time-invariant and time-dependent features for each hour (Tang et al., 2020, 2021). We reformatted the extract features to represent values for 2-hour buckets.

The train/valid./test data sets contain 38532/19267/19267 ICU-stays with prevalence p of 0.116/0.114/0.114 (fraction of death cases).

Static Features. For static features, we chose 5 time-invariant variables, similar to those used in the MIMIC-IV experiment, which were extracted to 24 discretized and/or one-hot encoded features (Table E.3).

Variable Name	Value Ranges
admissionheight	$(-0.001, 160.0]$
	$(160.0, 167.0]$
	$(167.0, 172.7]$
	$(172.7, 180.0]$
	$(180.0, 612.6]$
admissionweight	$(-0.001, 63.5]$
	$(63.5, 74.8]$
	$(74.8, 86.2]$
	$(86.2, 102.5]$
	$(102.5, 953.0]$
age	$(-0.001, 51.0]$
	$(51.0, 61.0]$
	$(61.0, 69.0]$
	$(69.0, 78.0]$
	$(78.0, 89.0]$
	> 89
ethnicity	African American
	Asian
	Caucasian
	Hispanic
	Other / Unknown
gender	Female
	Male

Table E.3: Static Features and Associated Value Ranges used in eICU experiments

Time-varying features. We apply 6 summary functions (mean, variance, min, median, max, slope) to each of the 85 FIDDLE features, comprising of 17 unique time-varying variables which were discretized and one-hot encoded (Table E.4), over 3 windows (0-48 hours, 24-48 hours, 32-48 hours). We chose variables that also had associated minimum and maximum features in the complete extracted features. This meant that during extraction, these were deemed "frequent" variables. For example, a feature that was included indicates if temperature measured was between the range $(36.4, 37.0]$. We removed the "was ever measured" and "time since last measurement" summary functions used with MIMIC-IV, as the extracted features from FIDDLE already removed missing values. A future step could be to incorporate the "mask" variable (which likely represents missingness) found in the extracted features from FIDDLE post-application of summary functions, as we do not want to apply the other 6 summary functions to "mask."

Evaluation plan. Given the similarity of the eICU mortality prediction task with the task from MIMIC-IV, we use the same scenarios for setting $\alpha, \kappa, C_{\text{FP:FN}}$, the same list of possible classifiers and hyperparameters, and the same selection strategies.

Variable Name	Value Range
Vital Signs|Heart Rate|Heart Rate	$(-0.001, 70.0]$, $(70.0, 80.0]$, $(80.0, 90.0]$, $(90.0, 102.0]$, $(102.0, 7734.0]$
Vital Signs|O2 Saturation|O2 Saturation	$(-0.001, 95.0]$, $(95.0, 96.0]$, $(96.0, 98.0]$, $(98.0, 100.0]$
Vital Signs|Respiratory Rate|Respiratory Rate	$(-0.001, 15.0]$, $(15.0, 18.0]$, $(18.0, 20.0]$, $(20.0, 24.0]$, $(24.0, 330.0]$
cvp	$(-95.001, 6.0]$, $(6.0, 10.0]$, $(10.0, 14.0]$, $(14.0, 21.0]$, $(21.0, 400.0]$
heartrate	$(-0.001, 70.0]$, $(70.0, 80.0]$, $(80.0, 90.0]$, $(90.0, 102.0]$, $(102.0, 300.0]$
noninvasivediastolic	$(-0.001, 53.0]$, $(53.0, 60.0]$, $(60.0, 67.0]$, $(67.0, 77.0]$, $(77.0, 263.0]$
noninvasivemean	$(-0.001, 67.0]$, $(67.0, 75.0]$, $(75.0, 84.0]$, $(84.0, 94.0]$, $(94.0, 285.0]$
noninvasivesystolic	$(-0.001, 101.0]$, $(101.0, 113.0]$, $(113.0, 125.0]$, $(125.0, 140.0]$, $(140.0, 287.0]$
respiration	$(-0.001, 15.0]$, $(15.0, 18.0]$, $(18.0, 21.0]$, $(21.0, 24.0]$, $(24.0, 194.0]$
sao2	$(-0.001, 94.0]$, $(94.0, 96.0]$, $(96.0, 98.0]$, $(98.0, 100.0]$
st1	$(-25.001, -0.2]$, $(-0.2, -0.01]$, $(-0.01, 0.0]$, $(0.0, 0.2]$, $(0.2, 1090.0]$
st2	$(-16.551, -0.4]$, $(-0.4, -0.09]$, $(-0.09, 0.1]$, $(0.1, 0.48]$, $(0.48, 1170.0]$
st3	$(-23.401, -0.3]$, $(-0.3, -0.08]$, $(-0.08, 0.1]$, $(0.1, 0.4]$, $(0.4, 1170.0]$
systemicdiastolic	$(-75.001, 48.0]$, $(48.0, 55.0]$, $(55.0, 62.0]$, $(62.0, 71.0]$, $(71.0, 395.0]$
systemicmean	$(-72.001, 66.0]$, $(66.0, 74.0]$, $(74.0, 81.0]$, $(81.0, 93.0]$, $(93.0, 397.0]$
systemicsystolic	$(-94.001, 98.0]$, $(98.0, 112.0]$, $(112.0, 124.0]$, $(124.0, 141.0]$, $(141.0, 398.0]$
temperature	$(-2622.401, 36.4]$, $(36.4, 37.0]$, $(37.0, 37.4]$, $(37.4, 37.8]$, $(37.8, 105.8]$

Table E.4: Dynamic Features and Associated Value Ranges used in eICU experiments

In Scenario 1 (more permissive), on eICU data the chosen cost ratio distribution $C_{\text{FP:FN}} \sim \text{Unif}(\frac{1}{9}, \frac{1}{6})$ maps to a non-uniform distribution over the fractional cost parameter $t \in [0.46, 0.56]$.

In Scenario 2 (more constrained), on eICU data the chosen cost ratio distribution $C_{\text{FP:FN}} \sim \text{Unif}(\frac{1}{40}, \frac{1}{20})$ maps to a non-uniform distribution over $t \in [0.16, 0.28]$.

E.4 Model Development, Hyperparameter Grid, and Computational Hardware

Given a specific dataset for the mortality prediction task (either MIMIC-IV or eICU), we train many versions of 3 different model families: logistic regression (LR), multi-layer perceptron (MLP), and random forest (RF) using implementations from sklearn (Pedregosa et al., 2011). For each model family m in LR, RF, and MLP, we train a set $\mathcal{H}_m$ of different candidates across a spectrum of hyperparameters designed to span under-fitting and over-fitting, hopefully including some well-fitting model instances.

The full hyperparameter grid is provided below in Table E.5. The logistic regression model was allowed a larger hyperparameter grid to compensate for its reduced parameter size, attempting to give each model family roughly equal computation runtime.

Hardware. All experiments were run using 4 CPU cores and 16GB of memory on a high-performance computing cluster. Only 30 minutes of wallclock time was allowed for each individual model fit. All runs completed within this limit.

Hyperparameter	Logistic Regression	MLP	Random Forest
Rare Class Weight	1, 3, 9, 27	–	1, 3, 9, 27
Max Iterations	1000, 49, 7, 1	100, 200	–
Inverse L2 Reg. Strength C	10, 100, 1000, 10000, 100000	–	–
L2 Regularization Strength	–	0.0001, 0.001	–
Hidden Layer Sizes	–	(64,), (64,64), (128,64)	–
Learning Rate	–	0.001, 0.0005	–
Max Depth	–	–	4, 16, 64
Minimum Examples in Leaf	–	–	4, 16, 64

Table E.5: Hyperparameter grid used in model selection experiments on MIMIC-IV and eICU.

E.5 Procedure for model selection and deployment-aware testing

For each of Scenario 1 and Scenario 2 described in the main paper, we define at the outset some desired constraints via specific values of α, κ, as well as a desired non-uniform density over the fractional cost-parameter $p(t)$, via a chosen scenario-specific distribution over the cost ratio $C_{\text{FP:FN}}$.

We then perform the model selection and threshold selection phases described below for each scenario and dataset (MIMIC-IV and eICU). Each time, we begin with all ROC curves from the full hyperparameter grid described above in Tab. E.5.

Model selection phase. Denote the union of all trained model-hyperparameter configurations from Tab. E.5 as $\mathcal{H} = \mathcal{H}_{LR} \bigcup \mathcal{H}_{RF} \bigcup \mathcal{H}_{MLP}$. Each element in $\mathcal{H}$ results in a score-producing classifier (as defined in Sec. 2) with its own ROC curve on the validation set.

Given possible model-hyperparameter configurations $\mathcal{H}$ and their corresponding validation-set ROC curves, and a scenario-specific values for the constraints α, κ and a cost distribution $p(t)$, we use each of the following selection strategies to pick a single model-hyperparameter combination $\mathsf{h}^* \in \mathcal{H}$:

- maximizing partial VOROS: accounts for cost distribution $p(t)$ and constraints α, κ
- maximizing total VOROS: accounts for cost distribution $p(t)$ but *ignores* α, κ
- maximizing recall in feasible region: accounts for constraints α, κ but ignores costs $p(t)$
- maximizing partial AUROC in feasible region: accounts for constraints α, κ but ignores costs $p(t)$

Threshold selection phase. Using the selected model-hyperparameter h^*, we determine the binarization threshold $\tau(t)$ as follows:

- For cost-unaware strategies (max recall or max pAUROC): We use the same threshold regardless of t, so $\tau(t)$ is just a constant function. We pick the threshold value using h^*'s validation set ROC curve, searching over all valid thresholds for the one that produces maximum recall while ensuring the predictions follow the precision and capacity constraints α, κ.

- For cost-aware strategies (partial VOROS or total VOROS): At each value of cost parameter t, we find a cost-specific threshold $\tau(t)$ using h^*'s validation set ROC curve. We search over all valid thresholds for the ROC point h, k that both meets the desired precision and capacity constraints set by α, κ and also minimizes cost with respect to t (using Defn. 3).

Test phase. We wish to mimic authentic deployment of a real alert system for given limits of α, κ and costs $t \in [a, b]$ weighted by the given density $p(t)$. We will force each selected model on the test data to use its predetermined threshold $\tau(t)$ to perform alerts, thus producing binary predictions (not scores) on the test set for a given t. This avoids "tuning" the selected threshold on the test set, which is not possible in prospective deployments.

On the test set using each predetermined threshold $\tau(t)$, we record the fpr,tpr location in ROC space as $(h_{\tau(t)}, k_{\tau(t)})$. We then report the expected cost over the provided range:

$$\begin{aligned}\mathbb{E}_{t\sim p(t)}[\text{Cost}] &= \int_{t=a}^{b} p(t)\text{Cost}_t(h_{\tau(t)}, k_{\tau(t)})dt \\ &= \int_{t=a}^{b} p(t)\Big[th_{\tau(t)} + (1-t)(1-k_{\tau(t)})\Big]dt\end{aligned} \tag{2}$$

In this paper, throughout Sec. 5-6 we always assume a uniform distribution over cost ratios $C_{\text{FP:FN}}$, between minimum ratio a_c and maximum ratio b_c. We can then write the original expected cost as an integral over possible values c for this ratio:

$$\mathbb{E}_{c\sim\text{Unif}(a_c,b_c)}[\text{Cost}] = \int_{c=a_c}^{b_c} \frac{1}{b_c - a_c}\text{Cost}_{t(c)}(h_{\tau(c)}, k_{\tau(c)})dc \tag{3}$$

There are two ways to evaluate this integral in practice:

- Monte Carlo estimation. We can sample S iid ratios $c_s \sim \text{Unif}(a_c, b_c)$, map each to a fractional cost parameter $t_s = t(c_s)$, then compute

$$\approx \frac{1}{S}\sum_{s=1}^{S}\Big[t_s h_{\tau(t_s)} + (1-t_s)(1-k_{\tau(t_s)})\Big] \tag{4}$$

- Numerical integration via the trapezoid method, with a sufficiently dense grid of c values.

For all expected cost evaluations in this paper, we use the trapezoid method with a grid of 1000 evenly-spaced c values from a_c to b_c.

E.6 Results on MIMIC-IV and eICU: Comparing Model Selection Strategies

Expanded tables showing many relevant performance metrics on both validation set and test set are in Tab. E.1 and Tab. E.2.

Scenario 1 on MIMIC-IV

$\alpha = 0.15, \quad \frac{C_0}{C_1} \sim \text{Unif}(\frac{1}{9}, \frac{1}{6})$

	on Val. set: $t \in [0.5, 0.6], \kappa = 0.5\|\mathcal{D}\| = 3901$,				on Test set: $t \in [0.48, 0.58], \kappa = 0.5\|\mathcal{D}\| = 3930.5$			
Strategy	Avg. Cost	Precision	Capacity	Satisfied?	Avg. Cost	Precision	Capacity	Satisfied?
max VOROS	**0.266**	0.250	2277.5	Yes	**0.261**	0.262	2310.1	Yes
max pAUROC	0.313	0.178	3896	Yes	0.303	0.188	3919	Yes
max recall	0.304	0.181	3888	Yes	0.302	0.188	3926	Yes
max PV (ours)	**0.266**	0.250	2277.5	Yes	**0.261**	0.262	2310.1	Yes

Scenario 2 on MIMIC-IV

$\alpha = 0.5, \quad \frac{C_0}{C_1} \sim \text{Unif}(\frac{1}{40}, \frac{1}{20})$

	on Val. set: $t \in [0.18, 0.31], \kappa = 0.1\|\mathcal{D}\| = 780.2$,				on Test set: $t \in [0.17, 0.29], \kappa = 0.1\|\mathcal{D}\| = 786.1$			
Strategy	Avg. Cost	Precision	Capacity	Satisfied?	Avg. Cost	Precision	Capacity	Satisfied?
max VOROS	0.665	0.505	196	Yes	0.640	0.597	236	Yes
max pAUROC	**0.561**	0.501	429	Yes	**0.538**	0.533	486	Yes
max recall	**0.561**	0.501	429	Yes	**0.538**	0.533	486	Yes
max PV (ours)	**0.561**	0.501	429	Yes	**0.538**	0.533	486	Yes

Figure E.1: **Detailed results for selected classifiers on MIMIC-IV.** Average cost, precision, capacity on valid and test sets across two misclassification cost scenarios for maximum (full) VOROS, maximum partial AUROC, maximum recall, and maximum partial VOROS strategies. Here on MIMIC-IV, no selected models violated constraints on test.

Scenario 1 on eICU

$\alpha = 0.15, \quad \kappa = 0.5|\mathcal{D}| = 9633.5, \quad \frac{C_0}{C_1} \sim \text{Unif}(\frac{1}{9}, \frac{1}{6})$

	on Val. set: $t \in [0.46, 0.56]$				on Test set: $t \in [0.46, 0.56]$			
Strategy	Avg. Cost	Precision	Capacity	Satisfied?	Avg. Cost	Precision	Capacity	Satisfied?
max VOROS	**0.307**	0.224	6845.4	Yes	**0.319**	0.215	7037.7	Yes
max pAUROC	0.325	0.186	9631	Yes	0.336	0.182	9800	No (capacity)
max recall	0.325	0.186	9631	Yes	0.336	0.182	9800	No (capacity)
max PV (ours)	**0.307**	0.224	6845.4	Yes	**0.319**	0.215	7037.7	Yes

Scenario 2 on eICU

$\alpha = 0.5, \quad \kappa = 0.1|\mathcal{D}| = 1926.7, \quad \frac{C_0}{C_1} \sim \text{Unif}(\frac{1}{40}, \frac{1}{20})$

	on Val. set: $t \in [0.16, 0.28]$				on Test set: $t \in [0.16, 0.28]$			
Strategy	Avg. Cost	Precision	Capacity	Satisfied?	Avg. Cost	Precision	Capacity	Satisfied?
max VOROS	0.773	0.500	12	Yes	**0.772**	0.684	19	Yes
max pAUROC	**0.693**	0.501	483	Yes	0.707	0.437	474	No (precision)
max recall	**0.693**	0.501	483	Yes	0.707	0.437	474	No (precision)
max PV (ours)	**0.693**	0.501	483	Yes	0.707	0.437	474	No (precision)

Figure E.2: **Detailed results for selected classifiers on eICU.** Average cost, precision, capacity on valid and test sets across two misclassification cost scenarios for maximum (full) VOROS, maximum partial AUROC, maximum recall, and maximum partial VOROS strategies. Results that violate constraints are highlighted red.

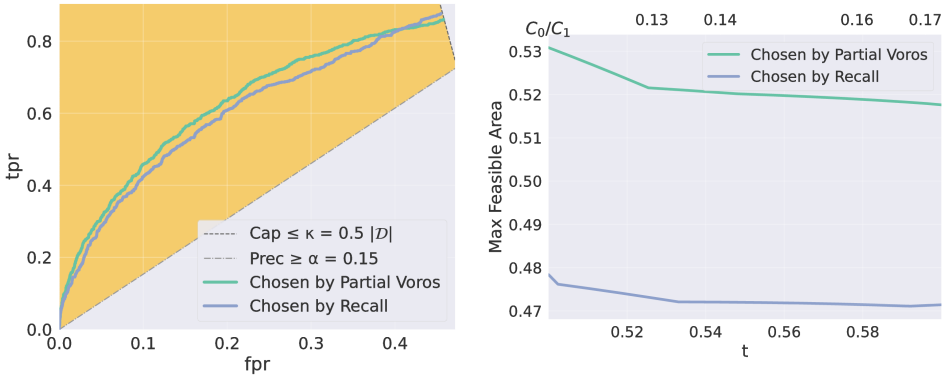

Figure E.3: **Diagnostic visuals for model selection on MIMIC-IV.** We compare our *partial VOROS* against other strategies for selecting model family, hyperparameters, and decision thresholds using validation data. We evaluate each selection's binary alerts on test data in terms of average cost over the provided distribution for cost ratio $C_{\text{FP:FN}}$. *Left:* The ROC curve on heldout data in the feasible region (gold), for the two best selection strategies in scenario 1. *Right:* The maximum feasible area of lesser classifiers across a range of cost parameter values t, with the corresponding cost ratio $C_{\text{FP:FN}}$ on the top axis.

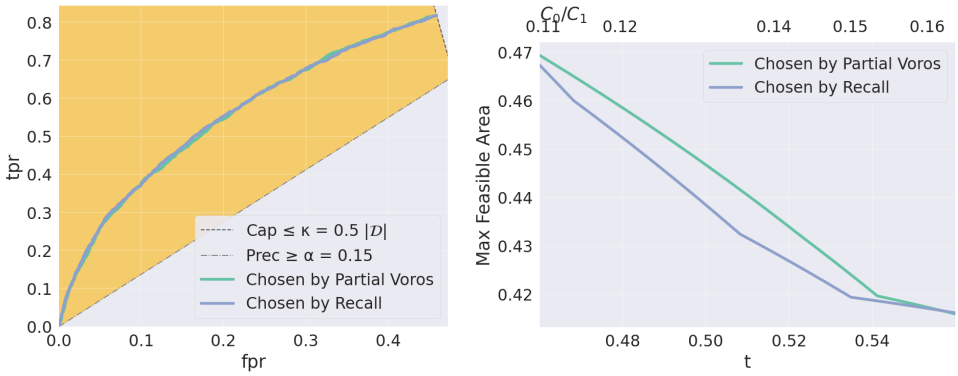

Figure E.4: **Diagnostic visuals for model selection on eICU.** We show plots for models selected by partial VOROS and recall strategies for Scenario 1. *Left:* ROC curves in the feasible region on the validation set. *Right:* Maximum feasible area of lesser classifiers for a range of cost parameter $t \in [0.46, 0.56]$ on the validation set.