Counterfactual Explanations via Locally-guided Sequential Algorithmic Recourse

Authors: Edward A. Small, Jeffrey N. Clark, Christopher J. McWilliams, Michael Ambler, Kacper Sokol, Jeffrey Chan, Flora D. Salim, Raul Santos-Rodriguez

ACM Transactions on Intelligent Systems and Technology, 2025

Saman Khamesian

08/26/2026

What problem are they solving?

A counterfactual explanation starts with:

- **Factual:** the current input x and its model prediction
- **Counterfactual:** a modified input x' receiving the desired prediction

For example:

$$f(x) = \text{Not Ready for Discharge}$$

but:

$$f(x') = \text{Ready for Discharge.}$$

The problem is that many counterfactual methods only return the final difference:

$$z = x' - x.$$

What problem are they solving?

This can be unrealistic. The final counterfactual may require many variables to change simultaneously, ignore dependencies between variables, or provide no indication of the order in which changes should occur.

LocalFACE instead constructs:

$$x \to x_1 \to x_2 \to \cdots \to x',$$

where each transition represents an intermediate step through a region supported by observed data.

Their argument is:

> A good final counterfactual is not enough; the path used to reach it should also be feasible.

What problem are they solving?

Simple example

A model might say that an ICU patient becomes discharge-ready when:

- Fever resolves
- Respiratory condition improves
- Mechanical ventilation is removed

But directly jumping from the current state to that final state says nothing about whether the combination is realistic or in which order the physiological changes normally occur.

LocalFACE tries to identify a sequence resembling patterns found among real patients.

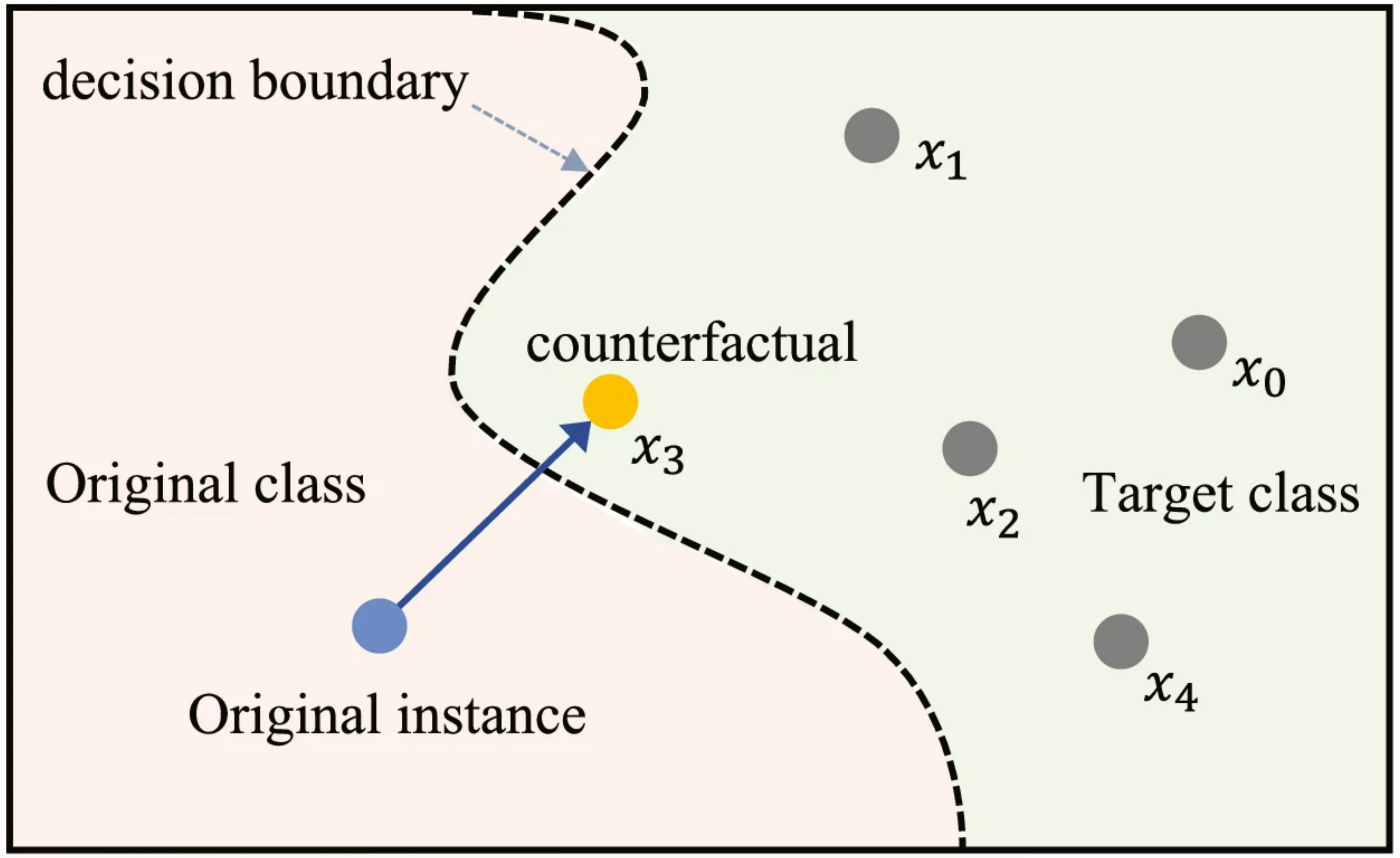
Related Work and Limitations

Why not use existing counterfactual methods?

The paper discusses two broad categories.

Gradient-based methods

These optimize the input until the classifier changes its output.



Their advantages:

- Often find a close counterfactual
- Do not necessarily require access to the training dataset

Their weakness:

- The modified point may fall outside realistic data regions
- Intermediate changes may be impossible
- They may exploit strange classifier behavior

Related Work and Limitations

Data-based methods such as FACE

FACE constructs a graph using observed training samples and searches for a path through high-density data regions.

Its strength:

- The path is better supported by observed data

Its weaknesses, according to the authors:

- It requires constructing a graph from the broader dataset
- It assumes the desired counterfactual is already known
- Fixed distance thresholds can exclude individuals in sparse regions
- It may raise privacy and model-secrecy concerns by exposing more data than necessary

LocalFACE is designed as an extension of the core FACE idea.

Main Contributions

1. **Finding the counterfactual and its path together:**
FACE primarily finds a path to a supplied counterfactual. LocalFACE first searches for a potentially reachable counterfactual and then constructs its path.

2. **Sequential rather than one-step recourse:**
It provides multiple intermediate changes rather than one large feature-vector modification.

3. **Data-supported feasibility:**
It uses local density as a proxy for whether the suggested states and transitions are realistic.

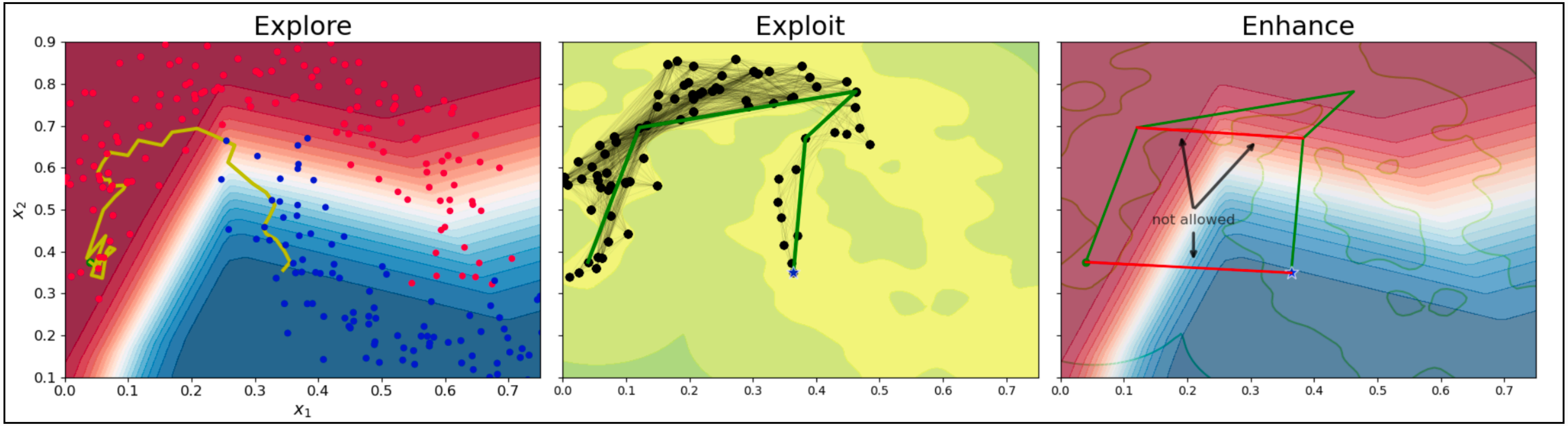
4. **Local information retrieval:**
It attempts to access only the subset of data relevant to the current explanation.

Methodology

The LocalFACE pipeline

The entire contribution is organized into three steps:

1. **Explore:** Find a reachable counterfactual.
2. **Exploit:** Retrieve only the local data needed to connect the factual and counterfactual.
3. **Enhance:** Optimize the final path through that local graph.



Step 1: Explore

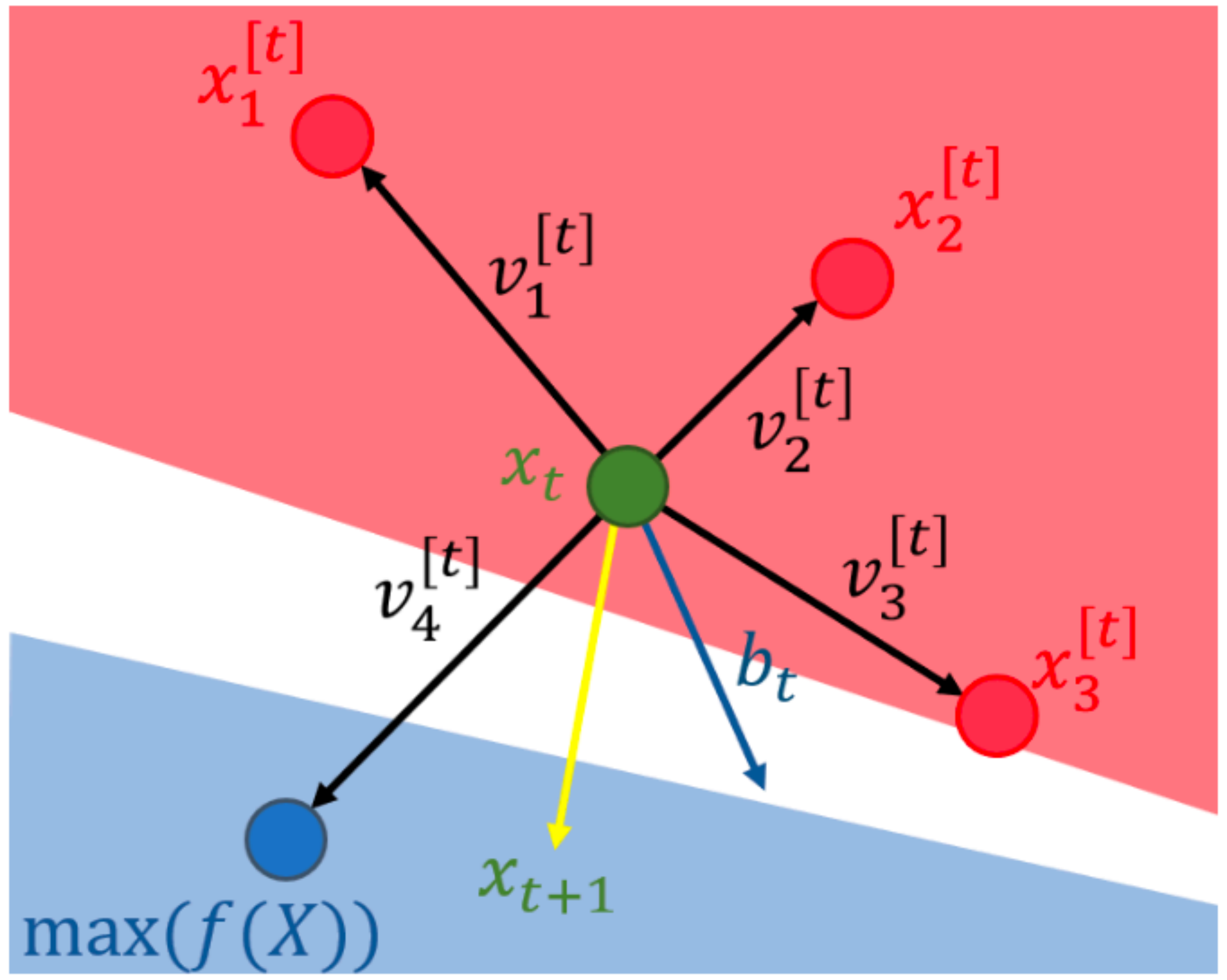


Figure 2: An example for finding x_{t+1} for 4 nearest neighbours using momentum b_t.If $f(x_t) \geq T_f$ then $x' = x_t$.

Algorithm 1: Find viable counterfactual x'.

Require: starting point x_0; decision function f; KD-Tree of data set T_X; decision threshold T_f; m history for momentum; k for nearest neighbours.
Ensure: best counterfactual candidate x'.

- $t \leftarrow 0$
- $x_t \leftarrow x_0$
- **while** $f(x_t) < T_f$ **do**
  - $G_t \leftarrow T_X(x_t)$ {get k-NN by querying T_X}
  - **if** $m > t > 0$ **then**
    - $b_t \leftarrow \frac{1}{t}\sum_{i=1}^{t}(Z_i - Z_{i-1})$
  - **else**
    - $b_t \leftarrow$ Equation 3
  - **end if**
  - $x_{t+1} \leftarrow$ Equation 2
  - $Z_{t+1} \leftarrow x_{t+1} - x_t$
  - $T_X \leftarrow T_X - x_{t+1}$ {update tree}
  - $t \leftarrow t + 1$
- **end while**
- $x' \leftarrow x_t$

At any iteration of our algorithm – denoted by a point x_t – we seek a candidate point that is likely to be in the direction of the chosen counterfactual class by finding the k nearest neighbours $x_i \in G_t \subset X$ of x_t and considering

$$x_{t+1} = x_t + \frac{\max_i\left(\frac{f(x_i)}{1+\|u_i\|}\right) - x_t + b_t}{2} \quad \forall x_i \in G_t \qquad (2)$$

where $u_i = x_i - x_t$ and b_t is the average direction of the last m steps, which can be understood here as the *momentum*:

$$b_t = \frac{1}{m}\sum_{i=(t-m)}^{t} z_i \,. \qquad (3)$$

The algorithm begins at the factual point x. At each iteration, it:

1. Queries the k nearest observed data points.
2. Evaluates which neighbor moves toward a higher probability of the desired class.
3. Considers distance so it does not make an unnecessarily large jump.
4. Uses momentum from previous steps to maintain a consistent direction.
5. Repeats until it crosses the classifier's decision boundary.

Step 2: Exploit

Now that the factual x and counterfactual x′ are known, the algorithm retrieves relevant observations between them and constructs a small local graph.

Each node represents a data-supported intermediate state. The search is guided by:

- Direction toward the counterfactual
- Local data density
- Feasibility of the region between nodes

Algorithm 2: Generate local graph.

Require: starting point x_0; counterfactual x'; maximum distance ϵ or k-NN; KD-Tree of data set T_X; f_p probability density function.

Ensure: Local graph G.

$t \leftarrow 0$
$x_t \leftarrow x_0$
$Z_t \leftarrow x_t$
$V_t \leftarrow x_t$
while $x_t \neq x'$ **do**
    $G_t \leftarrow T_X(x_t, \epsilon; k)$ {get all points near x_t}
    $x_{t+1} \leftarrow$ Equation 4
    $T_X \leftarrow T_X - x_{t+1}$ {update tree}
    $v_{t+1} \leftarrow x_{t+1}$
    $t \leftarrow t + 1$
    for $i = 0; i > t; i = i + 1$ **do**
        if $f_p(x) > T_p$ **then**
            $w_{i,t} \leftarrow$ {Equation 5}
        else
            $w_{i,t} \leftarrow 0$
        end if
    end for
end while

Step 3: Enhance

Once the local graph is constructed, LocalFACE uses Dijkstra's shortest-path algorithm to identify the lowest-cost path from the factual to the counterfactual.

The output is a sequence:

$$Z = [z_1, z_2, \ldots, z_k],$$

where the cumulative changes transform x into x':

$$x' = x + \sum_{i=1}^{k} z_i.$$

This sequence constitutes the final counterfactual explanation and algorithmic recourse.

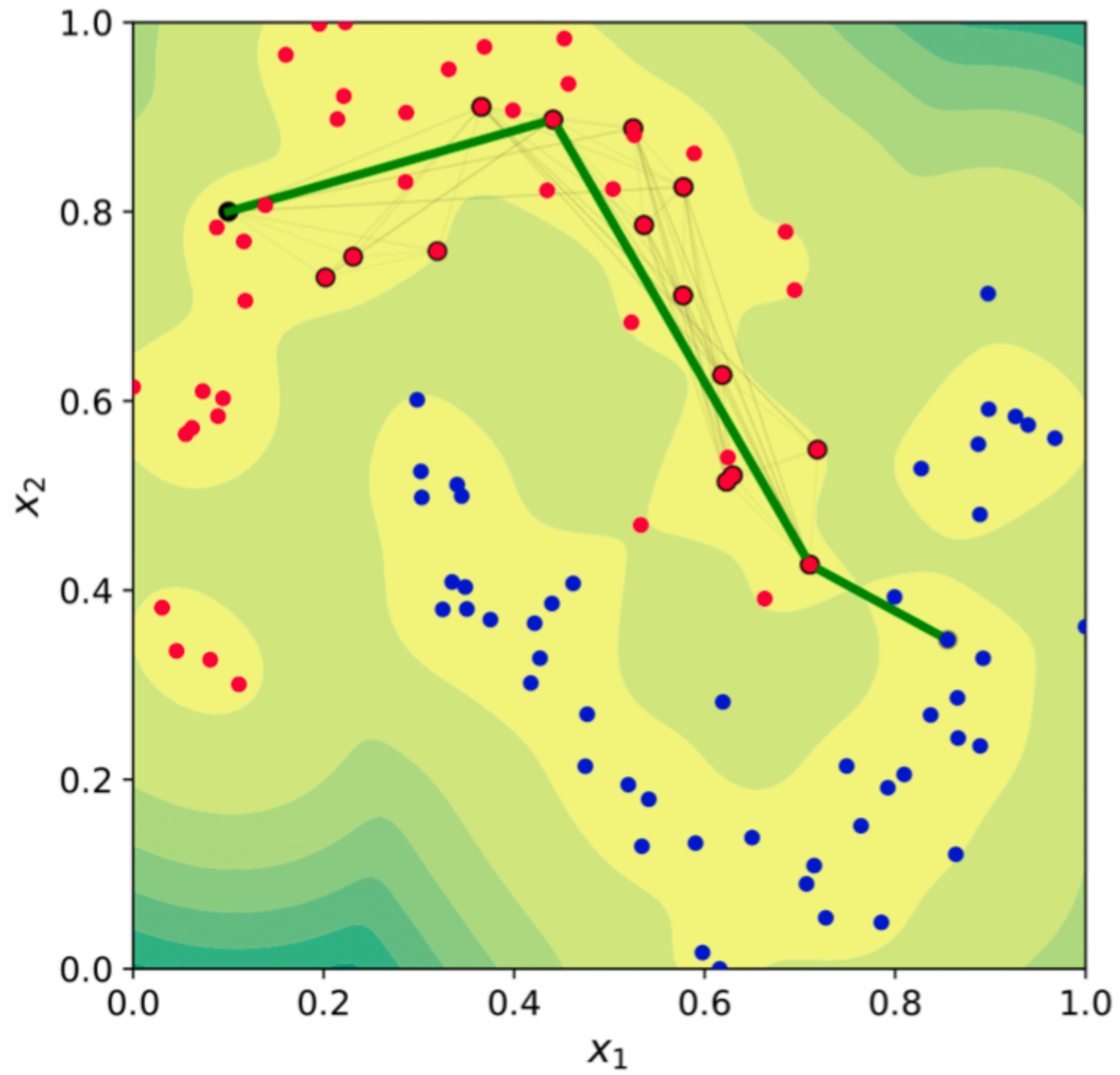


Figure 4: A local graph G (black from the exploit stage) where $V \subset X$. G captures the local knowledge required to find an optimal path (dark green from the enhance stage) from x to x' through high density data regions (yellow shading).

Case Study: Patient Readiness for Discharge from the ICU

They train classifiers to distinguish:

- Ready for discharge
- Not ready for discharge

The inputs include physiological and clinical variables such as:

- Temperature
- Oxygen saturation
- Respiratory rate
- Mechanical airway status
- Serum bicarbonate
- Creatinine
- Glasgow Coma Scale
- Length of stay

For a patient classified as not ready for discharge, LocalFACE generates a sequence of states that eventually causes the classifier to produce a high probability of readiness for discharge.

Evaluation

They examine three cases:

- **True negative:** Both clinician and model say the patient is not ready.
- **False negative:** The model says not ready, but the clinician discharged the patient successfully.
- **False positive:** The model says ready, but the clinician says not ready.

The purpose is not merely to “correct” the classification. The counterfactual path helps clinicians inspect what the model believes must change.

For example:

> The model appears to require resolution of fever, improved respiratory measurements, and removal of mechanical ventilation before predicting readiness for discharge.

That can help clinicians evaluate whether the model’s reasoning is clinically reasonable.

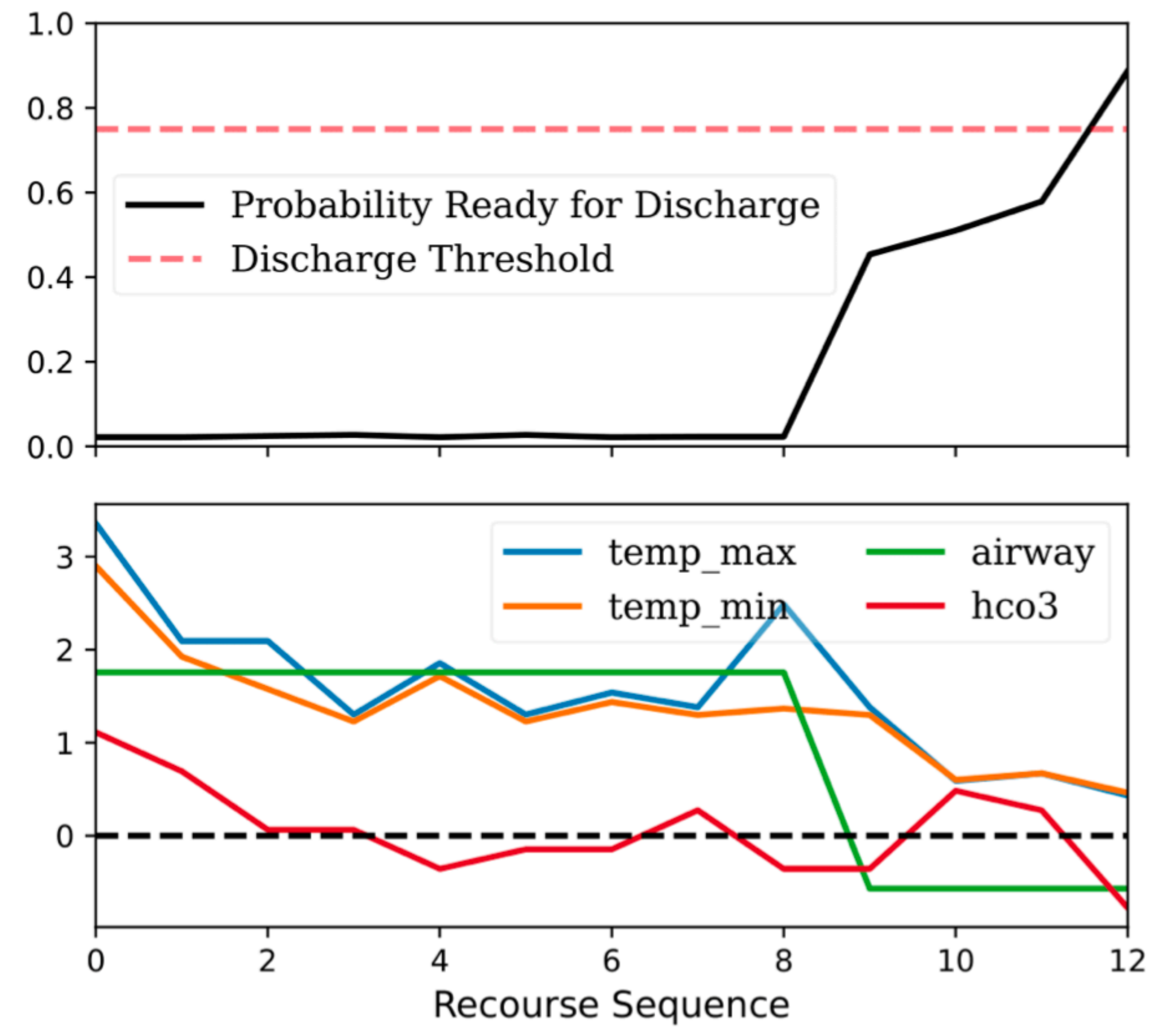
Quantitative Results

Model	AUROC	Brier loss	F1-score	Accuracy	Sensitivity
Random forest	0.8814	0.1442	0.8162	0.7978	0.8966
Logistic regression	0.8726	0.1436	0.8099	0.7920	0.8851

LocalFACE was applied to all 2,806 test patients and found a viable recourse path for every evaluated instance. The complete Explore–Exploit–Enhance procedure required an average of 0.453 seconds per patient.

True Negative – Actionable Recourse to Discharge

- The clinician and the classifier both determined that the patient was not ready for discharge.
- LocalFACE generated a sequence describing how the patient could progress toward a discharge-ready state.
- The path involved:
  - Resolving the fever
  - Improving oxygen saturation
  - Improving serum bicarbonate
  - Eventually removing mechanical ventilation
- The clinical team considered this sequence a realistic physiological recovery.
- Both classifiers produced closely corresponding paths.

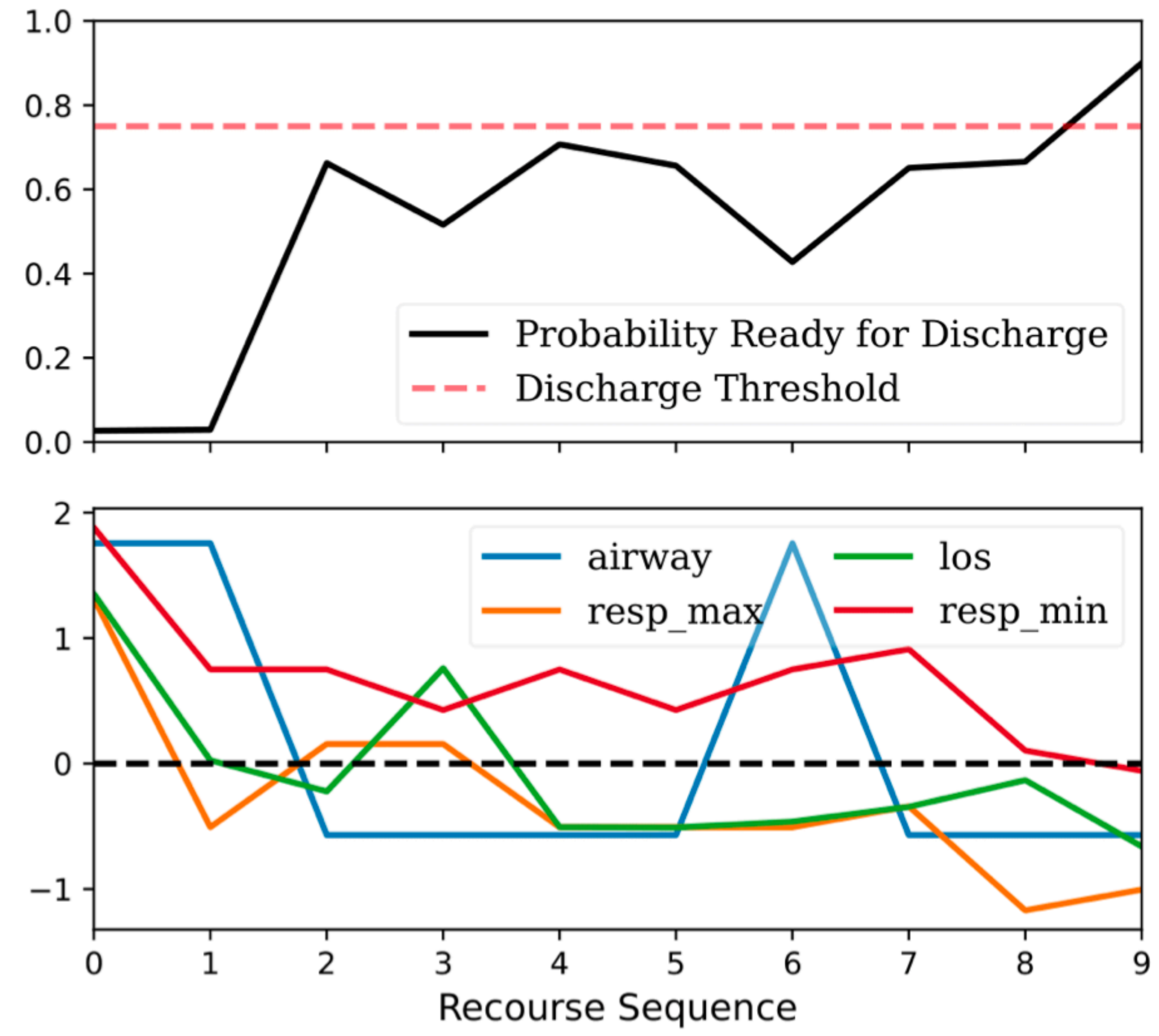


(a) True negative case.

False Negative – Hypothetical Recourse to Discharge

- The classifier predicted that the patient was not ready for discharge, although the clinician discharged the patient with a successful outcome.
- LocalFACE generated a hypothetical sequence showing what the model believed still needed to change before discharge.
- The path was mainly driven by respiratory concerns and involved:
  - Removing mechanical ventilation
  - Reducing the maximum respiratory rate
  - Reducing the minimum respiratory rate
  - Adjusting the length of stay
- The recourse helps the clinician understand why the model disagreed with the successful discharge decision.
- It can also help determine whether the model relied on clinically meaningful evidence or missed information available to the clinician.

FN Rate: 3.8% of the test set — 107 patients

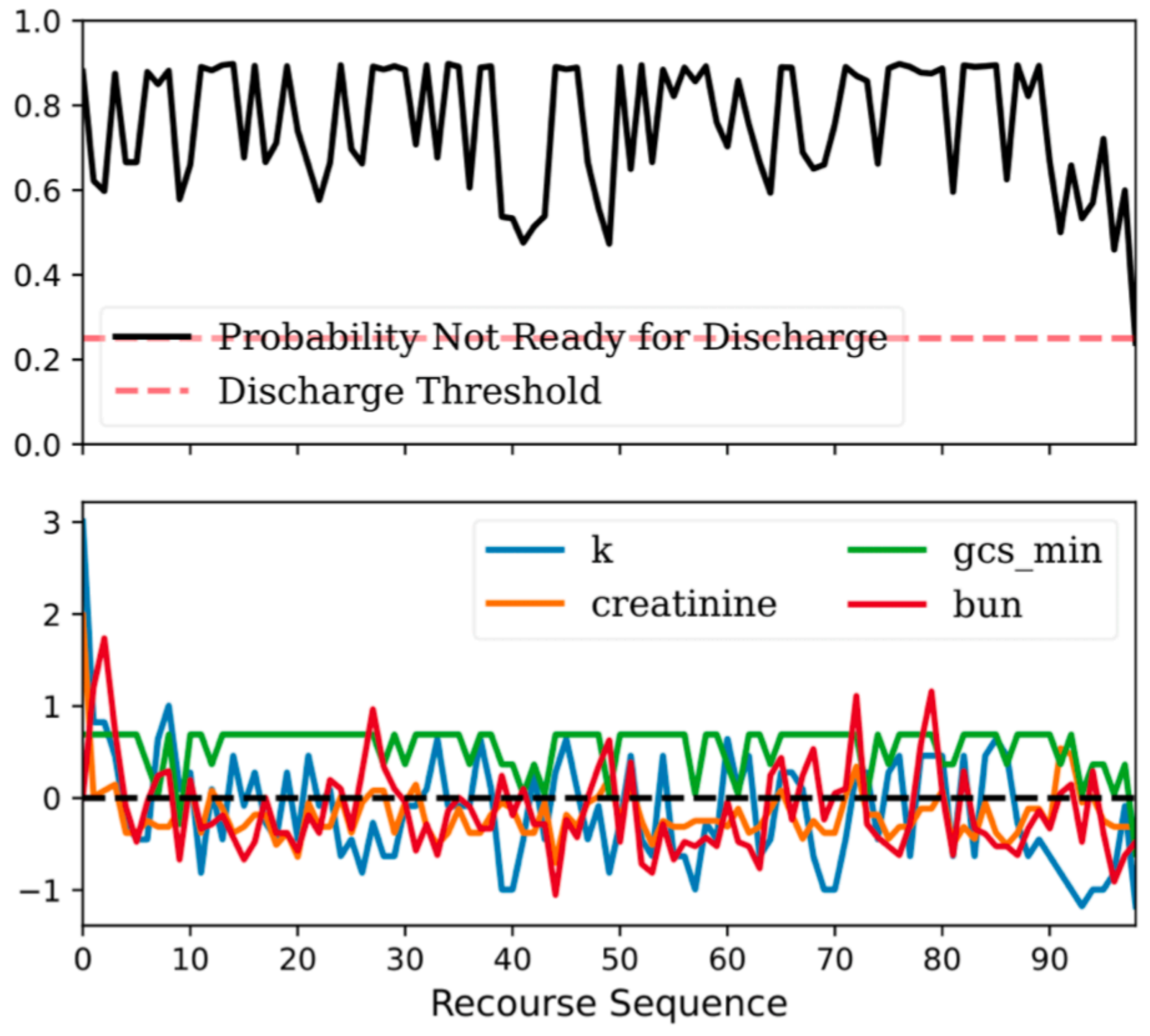


(b) False negative case.

False Positive – Reconciling Differing Beliefs

- The classifier predicted that the patient was ready for discharge, while the clinician determined that the patient was not ready.
- LocalFACE reversed the explanation direction and generated a path toward a not-ready-for-discharge prediction.
- The path primarily involved changes in:
    - Consciousness level
    - Potassium level
    - Creatinine
    - Serum bicarbonate
- The long recourse sequence suggested that the patient was located in a dense region of discharge-ready cases.
- This disagreement may indicate that information unavailable to the classifier influenced the clinician's decision to keep the patient in intensive care.

FP Rate: 15.4% of the test set — 433 patients



(c) False positive case.

Limitations

1. Data support does not establish a causal recovery path.
2. Some features, such as length of stay, are not directly actionable.
3. Data density is only a proxy for clinical feasibility.
4. The paths were evaluated through selected cases, not a formal clinician study.
5. Important treatment and intervention information was unavailable.

Conclusion

1. LocalFACE explains both the counterfactual and the path to reach it.

2. It combines local data, estimated density, and graph-based search.

3. Sequential paths can make model decisions easier to inspect.

4. The method was fast and found recourse for every evaluated patient.

5. Its paths support interpretation—not causal treatment recommendations.

Thank you for your attention