Supplementary Materials For Explaining Time Series Predictions with Dynamic Masks

Jonathan Crabbé [1] **Mihaela van der Schaar** [1 2 3]

1. More details on the mathematical formulation

1.1. Proofs

In this subsection section, we prove the propositions form the main paper.

Proposition 1 (Metric properties)**.** *For all labelling sets $A, B \subset [1:T] \times [1:d_X]$, the mask information and entropy enjoy the following properties:*

Positivity*:*

$$I_{\boldsymbol{M}}(A) \geq 0 \qquad S_{\boldsymbol{M}}(A) \geq 0$$

Additivity*:*

$$I_{\boldsymbol{M}}(A \cup B) = I_{\boldsymbol{M}}(A) + I_{\boldsymbol{M}}(B) - I_{\boldsymbol{M}}(A \cap B)$$
$$S_{\boldsymbol{M}}(A \cup B) = S_{\boldsymbol{M}}(A) + S_{\boldsymbol{M}}(B) - S_{\boldsymbol{M}}(A \cap B)$$

Monotonicity *If $A \subset B$:*

$$I_{\boldsymbol{M}}(A) \leq I_{\boldsymbol{M}}(B) \qquad S_{\boldsymbol{M}}(A) \leq S_{\boldsymbol{M}}(B).$$

Proof. Let us proof all the properties one by one.

Positivity By definition, all coefficients from the mask are normalized: $m_{t,i} \in [0,1]$ for $(t,i) \in [1:T] \times [1:d_X]$. Positivity follows trivially from the properties of the logarithm function

$$\begin{aligned} I_{\mathbf{M}}(A) &= -\sum_{(t,i)\in A} \underbrace{\ln\left(1-m_{t,i}\right)}_{\leq 0} \\ &\geq 0. \end{aligned}$$

The same goes for the entropy

$$\begin{aligned} S_{\mathbf{M}}(A) &= -\sum_{(t,i)\in A} \underbrace{m_{t,i}}_{\geq 0}\underbrace{\ln m_{t,i}}_{\leq 0} + \underbrace{(1-m_{t,i})}_{\geq 0}\underbrace{\ln\left(1-m_{t,i}\right)}_{\leq 0} \\ &\geq 0. \end{aligned}$$

Additivity We first note that the proposition follows trivially if the sets are distinct $A \cap B = \varnothing$:

$$\begin{aligned} I_{\mathbf{M}}\left(A \cup B\right) = & -\sum_{(t,i)\in A\cup B} \ln\left(1-m_{t,i}\right) \\ = & -\sum_{(t,i)\in A} \ln\left(1-m_{t,i}\right) \\ & -\sum_{(t,i)\in B} \ln\left(1-m_{t,i}\right) \\ = & \; I_{\mathbf{M}}(A) + I_{\mathbf{M}}(B). \end{aligned}$$

Now consider the case where $C \subset D$, since C and $D \setminus C$ are disjoint, we can write

$$I_{\mathbf{M}}\left(D\right) = I_{\mathbf{M}}\left(C\right) + I_{\mathbf{M}}\left(D \setminus C\right) \tag{1}$$
$$\Rightarrow I_{\mathbf{M}}\left(D \setminus C\right) = I_{\mathbf{M}}\left(D\right) - I_{\mathbf{M}}\left(C\right). \tag{2}$$

We shall now prove the additivity property in general by using these two ingredients. First we note that the set $A \cup B$ can be written as the disjoint union $A \sqcup [B \setminus (A \cap B)]$. It follows that

$$\begin{aligned} I_{\mathbf{M}}(A \cup B) &= I_{\mathbf{M}}\left(A\right) + I_{\mathbf{M}}\left(B \setminus [A \cap B]\right) \\ &= I_{\mathbf{M}}\left(A\right) + I_{\mathbf{M}}\left(B\right) - I_{\mathbf{M}}\left(A \cap B\right), \end{aligned}$$

where we have used the additivity property for disjoint sets in the first equality and the fact that $(A \cap B) \subset B$ in the second equality. The same reasoning holds for the entropy.

Monotonicity To prove the monotonicity property, it is useful to note that if $A \subset B$, we can use (2) to write

$$\begin{aligned} I_{\mathbf{M}}\left(A\right) &= I_{\mathbf{M}}\left(B\right) - \underbrace{I_{\mathbf{M}}\left(B \setminus A\right)}_{\geq 0} \\ &\leq I_{\mathbf{M}}\left(B\right), \end{aligned}$$

where we have used the information positivity to produce the inequality. The same reasoning holds for the entropy. □

1.2. Normalized information and entropy

In this subsection, we introduce the normalized counterparts of our information theoretic metrics. It is important to keep in mind that all the available information for issuing

[1]DAMTP, University of Cambridge, UK [2]University of California Los Angeles, USA [3]The Alan Turing Institute, UK. Correspondence to: Jonathan Crabbé <jc2133@cam.ac.uk>, Mihaela van der Schaar <mv472@cam.ac.uk>.

Proceedings of the 38^{th} International Conference on Machine Learning, PMLR 139, 2021.

a black-box prediction is in $[1:T]\times[1:d_X]$. Therefore, the monotonicity property allows to introduce normalized counterparts of the information and the entropy.

Definition 2 (Normalized metrics)**.** The normalized mask information associated to a mask $\mathbf{M}$ and a subsequence $(x_{t,i})_{(t,i)\in A}$ of the input $\mathbf{X}$ with $A\subseteq[1:T]\times[1:d_X]$ is

$$i_{\mathbf{M}}(A)=\frac{I_{\mathbf{M}}(A)}{I_{\mathbf{M}}([1:T]\times[1:d_X])}.$$

The same goes for the related normalized mask entropy

$$s_{\mathbf{M}}(A)=\frac{S_{\mathbf{M}}(A)}{S_{\mathbf{M}}([1:T]\times[1:d_X])}.$$

Remark 3. By the monotonicity and the positivity properties, it is clear that $0\leq i_{\mathbf{M}}(A), s_{\mathbf{M}}(A)\leq 1$ for all $A\subseteq[1:T]\times[1:d_X]$. This gives a natural interpretation of these quantities as being, respectively, the fraction of the total information and entropy contained in the subsequence A according to the mask $\mathbf{M}$.

The normalized version of the metrics allow to measure what percentage of the total mask information/entropy is contained in a given subsequence.

1.3. Definition of a mask for other saliency methods

In this section, we explain how to associate a mask to any saliency method. Suppose that a given method produces a score matrix $\mathbf{R}\in\mathbb{R}^{T\times d_X}$ that assigns an importance score $r_{t,i}$ for each element $x_{t,i}$ of the input matrix $\mathbf{X}$. Then, if we normalize the coefficients of the score matrix, we obtain an associated mask:

$$\begin{aligned}\mathbf{M}&=\frac{1}{r_{max}}\left[\mathbf{R}-r_{min}\cdot(1)^{T\times d_X}\right],\\ r_{min}&=\min\left\{r_{t,i}\mid(t,i)\in[1:T]\times[1:d_X]\right\}\\ r_{max}&=\max\left\{r_{t,i}\mid(t,i)\in[1:T]\times[1:d_X]\right\}\end{aligned}$$

where $(1)^{T\times d_X}$ denotes a $T\times d_X$ matrix with all elements set to 1. This mask can subsequently be used to compute the mask information content and entropy. In our experiments, we use this correspondence to compare our method with popular saliency methods.

2. More details on the implementation

2.1. Algorithm

The mask optimization algorithm is presented in Algorithm 1. In the algorithm, we used the notation $(0.5)^{T\times d_X}$ for a $T\times d_X$ matrix with all elements set[1] to 0.5. Similarly, $(0)^{T\cdot d_X\cdot(1-a)}$ denotes a vector with $T\cdot d_X\cdot(1-a)$ components set to 0 and $(1)^{T\cdot d_X\cdot a}$ denotes a vector with $T\cdot d_X\cdot a$ components set to 1. The symbol $\oplus$ denotes the direct sum between two vector spaces, which is equivalent to the concatenation in Algorithm 1. The error part of the loss $\mathcal{L}_e$ depends on the task (regression or classification), as explained in Section 3 of the paper. The momentum and the learning rate are typically set to 1, the number of epoch is typically 1000. We also use the clamp function, which is defined component by component as

$$\left[\text{clamp}_{[0,1]}\left(\mathbf{M}\right)\right]_{t,i}=\min\left[\max\left(m_{t,i},0\right),1\right].$$

Finally, we note that the mask size regularization coefficient λ_a grows exponentially during the optimization to reach a maximum value of $\delta\cdot\lambda_0$ at the end of the optimization. In practice, it is initialized to a small value (typically $\lambda_0=0.1$) and dilated by several order of magnitude during the optimization (typically $\delta=1000$). In this way, the optimization procedure works in two times. At the beginning, the loss is dominated by the error $\mathcal{L}_e(\mathbf{M})$ so that the mask increases the mask coefficients of salient features. As the regulation coefficient λ_a increases, the regulation term becomes more and more important so that the mask coefficients are attracted to 0 and 1. At the end of the optimization, the mask is almost binary.

Algorithm 1 Dynamask

Input: input sequence $\mathbf{X}\in\mathbb{R}^{T\times d_X}$, black-box f, perturbation operator Π, mask area $a\in[0,1]$, learning rate $\eta\in\mathbb{R}^+$, momentum $\alpha\in\mathbb{R}^+$, initial size regulator $\lambda_0\in\mathbb{R}^+$, regulator dilation $\delta\in\mathbb{R}_{\geq 1}$, time variation regulator $\lambda_c\in\mathbb{R}^+$, number of epochs $N\in\mathbb{N}$
Output: mask $\mathbf{M}\in[0,1]^{T\times d_X}$
$\mathbf{M}\leftarrow(0.5)^{T\times d_X}$
$\mathbf{r}_a\leftarrow(0)^{T\cdot d_X\cdot(1-a)}\oplus(1)^{T\cdot d_X\cdot a}$
$\Delta\mathbf{M}\leftarrow 0$
$\lambda_a\leftarrow\lambda_0$
for $i=1$ **to** N **do**
  $\tilde{\mathbf{X}}\leftarrow\Pi_{\mathbf{M}}\left(\mathbf{X}\right)$
  Evaluate the error $\mathcal{L}_e\left(\mathbf{M}\right)$ between $f(\mathbf{X})$ and $f(\tilde{\mathbf{X}})$
  $\mathcal{L}_a\left(\mathbf{M}\right)\leftarrow\|\text{vecsort}(\mathbf{M})-\mathbf{r}_a\|^2$
  $\mathcal{L}_c\left(\mathbf{M}\right)\leftarrow\sum_{i=1}^{d_X}\sum_{t=1}^{T-1}|m_{t+1,i}-m_{t,i}|$
  $\Delta\mathbf{M}\leftarrow\eta\cdot\nabla_{\mathbf{M}}\left[\mathcal{L}_e+\lambda_a\mathcal{L}_a+\lambda_c\mathcal{L}_c\right]+\alpha\cdot\Delta\mathbf{M}$
  $\mathbf{M}\leftarrow\mathbf{M}+\Delta\mathbf{M}$
  $\mathbf{M}\leftarrow\text{clamp}_{[0,1]}\left(\mathbf{M}\right)$
  $\lambda_a\leftarrow\lambda_a\times\exp\left(\log\delta/N\right)$
end for

2.2. Deletion variant

We notice that Algorithm 1 produces a mask that highlights the features that allow to reproduce the black-box prediction by keeping the error part of the loss $\mathcal{L}_e(\mathbf{M})$ to be small.

[1] By setting all the initial coefficients of the mask $\mathbf{M}$ to 0.5, we make no prior assumption on the saliency of each feature.

However, it is possible to highlight important features in another way. For instance, we could try to find the features that maximizes the prediction shift when perturbed. In this alternative formulation, the mask is obtained by solving the following optimization problem:

$$\tilde{\mathbf{M}}_a^* = \underset{\mathbf{M}\in[0,1]^{T\times d_X}}{\arg\min} -\mathcal{L}_e\left(1-\mathbf{M}\right) + \lambda\cdot\mathcal{L}_a\left(\mathbf{M}\right).$$

Note that, in this case, the sign of the error part is flipped in order to maximizes the shift in the prediction. Moreover, the error is now evaluated for $1-\mathbf{M}$ rather than $\mathbf{M}$. This is because important features are maximally perturbed in this case. In this way, a salient feature $x_{t,i}$ can keep a mask coefficient $m_{t,i}$ close to 1 while being maximally perturbed. The regulator stays the same in this deletion variant, as the mask area still corresponds to the number of mask coefficients set to 1. We use the deletion variant to obtain the masks in the experiment with clinical data in the main paper.

3. More details on the experiments

3.1. Metrics

We give the precise definition of each metric that appears in the experiments. Let us start with the metrics that are defined when the true importance is known.

Definition 4 (AUP,AUR)**.** Let $\mathbf{Q} = (q_{t,i})_{(t,i)\in[1:T]\times[1:d_X]}$ be a matrix in $\{0,1\}^{T\times d_X}$ whose elements indicate the true saliency of the inputs contained in $\mathbf{X}\in\mathbb{R}^{T\times d_X}$. By definition, $q_{t,i}=1$ if the feature $x_{t,i}$ is salient and 0 otherwise. Let $\mathbf{M} = (m_{t,i})_{(t,i)\in[1:T]\times[1:d_X]}$ be a mask in $[0,1]^{T\times d_X}$ obtained with a saliency method. Let $\tau\in(0,1)$ be the detection threshold for $m_{t,i}$ to indicate that the feature $x_{t,i}$ is salient. This allows to convert the mask into an estimator $\hat{\mathbf{Q}}(\tau) = (\hat{q}_{t,i}(\tau))_{(t,i)\in[1:T]\times[1:d_X]}$ for $\mathbf{Q}$ via

$$\hat{q}_{t,i}(\tau) = \begin{cases} 1 & \text{if } m_{t,i}\geq\tau \\ 0 & \text{else.}\end{cases}$$

Consider the sets of truly salient indexes and the set of indexes selected by the saliency method

$$A = \{(t,i)\in[1:T]\times[1:d_X] \mid q_{t,i}=1\}$$
$$\hat{A}(\tau) = \{(t,i)\in[1:T]\times[1:d_X] \mid \hat{q}_{t,i}(\tau)=1\}.$$

We define the precision and recall curves that map each threshold to a precision and recall score:

$$\mathrm{P}:(0,1)\longrightarrow[0,1]:\tau\longmapsto\frac{|A\cap\hat{A}(\tau)|}{|\hat{A}(\tau)|}$$
$$\mathrm{R}:(0,1)\longrightarrow[0,1]:\tau\longmapsto\frac{|A\cap\hat{A}(\tau)|}{|A|}.$$

The AUP and AUR scores are the area under these curves

$$\mathrm{AUP} = \int_0^1 \mathrm{P}(\tau)d\tau$$
$$\mathrm{AUR} = \int_0^1 \mathrm{R}(\tau)d\tau.$$

Remark 5. Roughly speaking, we consider the identification of salient features as a binary classification task. Each saliency method can thus be seen as a binary classifier for which we compute the AUP and the AUR.

Remark 6. Integrating over several detection thresholds allows to evaluate a saliency method with several levels of tolerance on what is considered as a salient feature.

In our experiment with MIMIC-III, since the ground true feature importance is unknown, we use the following metrics defined for a binary classification problem[2].

Definition 7 (CE, ACC)**.** Consider a classifier f that maps the input $\mathbf{X}$ to a probability $f(\mathbf{X})\in[0,1]$. Let $\tilde{\mathbf{X}}$ be a perturbed input produced by a saliency method[3]. We define the function that converts a probability into a class

$$\mathrm{class}(p) = \begin{cases} 0 & \text{if } p<0.5 \\ 1 & \text{else.}\end{cases}$$

To measure the shift in the classifier's prediction caused by the perturbation of the input for several test examples $\{\mathbf{X}_k \mid k\in[1:K]\}$, we use the binary cross-entropy (or log-loss)

$$\mathrm{CE} = -\frac{1}{K}\sum_{k=1}^{K}\mathrm{class}\left[f(\mathbf{X}_k)\right]\cdot\log f\left(\tilde{\mathbf{X}}_k\right) + \left(1-\mathrm{class}\left[f(\mathbf{X}_k)\right]\right)\cdot\log\left[1-f\left(\tilde{\mathbf{X}}_k\right)\right].$$

To measure the number of prediction flipped by the perturbation, we use the accuracy

$$\mathrm{ACC} = \frac{\left|\left\{k\in[1:K] : \mathrm{class}\left[f(\mathbf{X}_k)\right] = \mathrm{class}\left[f(\tilde{\mathbf{X}}_k)\right]\right\}\right|}{K}.$$

We reproduce our experiment several times to get an average and a standard deviation for all of these metrics.

3.2. Computing infrastructure

All our experiments have been performed on a machine with Intel(R) Core(TM) i5-8600K CPU @ 3.60GHz [6 cores] and Nvidia GeForce RTX 2080 Ti GPU.

[2] In our experiment, each input corresponds to a patient. Class 0 indicates that the patient survives and class 1 indicates that the patient dies.

[3] In our experiment, we replace the most important features by the time average of the corresponding feature.

3.3. Details on the rare experiment

Data generation Since this experiment relies on a white-box, we only have to generate the input sequences. As we explain in the main paper, each feature sequence is generated with an ARMA process:

$$x_{t,i} = \varphi_1 \cdot x_{t-1,i} + \varphi_2 \cdot x_{t-2,i} + \varphi_3 \cdot x_{t-3,i} + \epsilon_t,$$

with $\varphi_1 = 0.25$, $\varphi_2 = 0.1$, $\varphi_3 = 0.05$ and $\epsilon_t \sim \mathcal{N}(0, 1)$. We generate one such sequence with $t \in [1 : 50]$ for each feature $i \in [1 : 50]$ by using the Python statsmodels library.

In the rare feature experiment, 5 features are selected as salient. Their indices are contained in A_X and drawn uniformly without replacement from $[1 : 50]$. The salient times are defined as $A_T = [13 : 38]$.

In the rare time experiment, 5 time steps are selected as salient. The initial salient time is drawn uniformly $t^* \sim \mathrm{U}([1 : 46])$. The salient times are then defined as $A_T = [t^* : t^* + 4]$. The salient features are defined as $A_X = [13 : 38]$.

Mask fitting For each time series, we fit a mask by using the temporal Gaussian blur π^g as a perturbation operator with $\sigma_{max} = 1$ and by using the squared error loss. A mask is fitted for each value of $a \in \left\{(n+1) \cdot 10^{-3} \mid n \in [0 : 49]\right\}$. The mask $\mathbf{M}_a^*$ with the lowest squared error $\mathcal{L}_e(\mathbf{M}_a^*)$ is selected. The hyperparameters for this optimization procedure are $\eta = 1, \alpha = 1, \lambda_0 = 1, \delta = 1000, \lambda_c = 0, N = 1000$.

In our experiments, we don't consider $a > 0.05$. This is because we found experimentally that the error $\mathcal{L}_e(\mathbf{M}_a^*)$ generally reaches a plateau as a gets closer to 0.05, as illustrated in the examples from Figures 1 & 2. This is consistent with the fraction of inputs that are truly salient since

$$\frac{|A|}{|[1:50] \times [1:50]|} = \frac{25 \cdot 5}{50 \cdot 50} = 0.05.$$

Runtime For rare time, finding the best mask takes on average 15.7s. For rare feature, finding the best mask takes on average 20.7s.

Illustrations To illustrate the results of our experiments, we show the saliency masks produced by various methods for the rare feature experiment in Figure 3 & 4 and for the rare time experiment in Figure 5 & 6. For all of these examples, we notice that Dynamask identifies a bigger portion of the truly salient inputs, which illustrates the bigger AUR reported in the main paper.

3.4. Details on the state experiment

Data generation The data generation is governed by a Hidden Markov Model (HMM). The initial distribution vector for this HMM is given by $\boldsymbol{\pi} = (0.5, 0.5)$ and its transition matrix is

$$\mathbf{C} = \begin{pmatrix} 0.1 & 0.9 \\ 0.1 & 0.9 \end{pmatrix}.$$

At each time, the input feature vector has three components ($d_X = 3$) and is generated according to the current state via $\mathbf{x}_t \sim \mathcal{N}\left(\boldsymbol{\mu}_{s_t}, \boldsymbol{\Sigma}_{s_t}\right)$ with mean vectors depending on the state: $\boldsymbol{\mu}_1 = (0.1, 1.6, 0.5)$ or $\boldsymbol{\mu}_2 = (-0.1, -0.4, -1.5)$. When it comes to the covariance matrices, only the off-diagonal terms differ from one state to another:

$$\boldsymbol{\Sigma}_1 = \begin{pmatrix} 0.8 & 0 & 0 \\ 0 & 0.8 & 0.01 \\ 0 & 0.01 & 0.8 \end{pmatrix}$$
$$\boldsymbol{\Sigma}_2 = \begin{pmatrix} 0.8 & 0.01 & 0 \\ 0.01 & 0.8 & 0 \\ 0 & 0 & 0.8 \end{pmatrix}.$$

To each of these input vectors is associated a binary label $y_t \in \{0, 1\}$. This binary label is conditioned by one of the three component of the feature vector, based on the state:

$$p_t = \begin{cases} (1 + \exp\left[-x_{2,t}\right])^{-1} & \text{if } s_t = 0 \\ (1 + \exp\left[-x_{3,t}\right])^{-1} & \text{if } s_t = 1 \end{cases}.$$

The length of each time series is fixed to 200 (T = 200). We generate 1000 such time series, 800 are used for model training and 200 for testing.

Model training We train a RNN with one layer made of 200 GRU cells trained using the Adam optimizer for 80

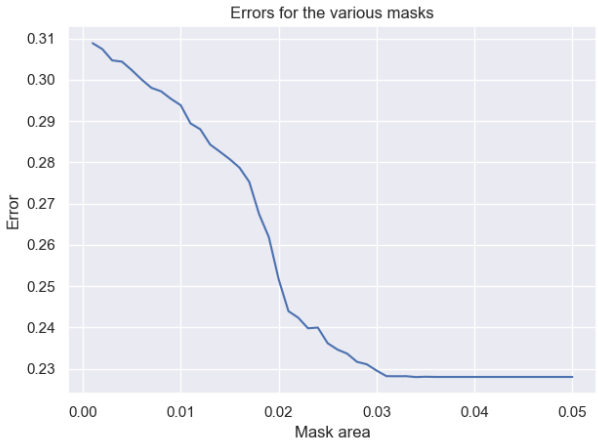

Figure 1. The error $\mathcal{L}_e(\mathbf{M}_a^*)$ as a function of a. We clearly see that the error stops decreasing when a gets close to 0.05. This group of masks are fitted on a time series from the rare feature experiment.

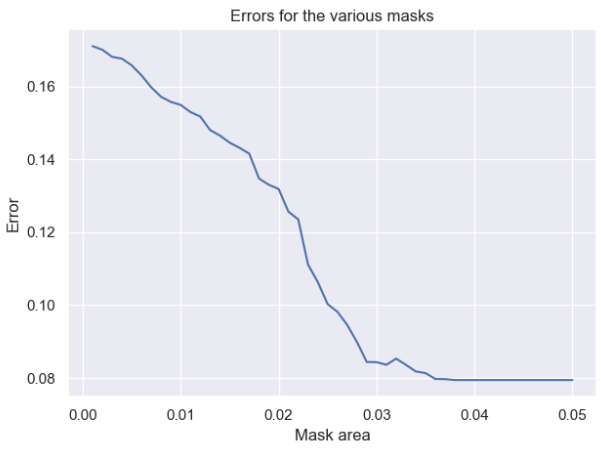

Figure 2. The error $\mathcal{L}_e(\mathbf{M}_a^*)$ as a function of a. We clearly see that the error stops decreasing when a gets close to 0.05. This group of masks are fitted on a time series from the rare time experiment.

epochs ($\text{lr} = 0.001, \beta_1 = 0.9, \beta_2 = 0.999$ and no weight decay).

Mask fitting For each test time series, we fit a mask by using the temporal Gaussian blur π^g as a perturbation operator with $\sigma_{max} = 1$. A mask is optimized for each value of $a \in \left\{0.15 + 2n \cdot 10^{-2} \mid n \in [0:10]\right\}$. We keep the extremal mask for a threshold set to $\varepsilon = 0.9 \cdot \mathcal{L}_e\left(\mathbf{M} = (1)^{T \times d_X}\right)$. The hyperparameters for this optimization procedure are $\eta = 1, \alpha = 1, \lambda_0 = 0.1, \delta = 100, \lambda_c = 1, N = 1000$.

Runtime Finding the extremal mask for a given input takes 49.8s on average.

Illustrations To illustrate the results of our experiments, we show the saliency masks produced by various methods on Figure 7 & 8. By inspecting these figures, we notice that only Dynamask and Integrated Gradients seem to produce saliency maps where the imprint of the true saliency can be distinguished. One advantage of Dynamask is the contrast put between these salient inputs and the rest. In the case of Integrated Gradients, we see that many irrelevant inputs are assigned an important saliency score, although smaller than the truly salient inputs. This is because gradients are computed individually, without the goal of achieving parsimonious feature selection.
In addition, we have reported the mask entropy for each of the methods. As claimed in the main paper, Dynamask produces mask that have significantly lower entropy. Among the methods that produce masks with high entropy, we notice two trends. Some methods, such as RETAIN, produce masks where a significant portion of the inputs are assigned a mask entropy close the 0.5. As discussed in the main paper, these significance of the saliency scores is limited in this situation, since no clear contrast can be drawn between the saliency of different inputs. On the other hand, some methods like FIT produce masks with many different masks coefficients, which renders the saliency map somewhat fuzzy. In both cases, the high entropy detects these obstructions for legibility.

3.5. Details on the mimic experiment

Data preprocessing The data preprocessing used here is precisely the same as the one described in (Tonekaboni et al., 2020), we summarize it here for completeness. We use the adult ICU admission data from the MIMIC-III dataset (Johnson et al., 2016). For each patient, we use the features Age, Gender, Ethnicity, First Admission to the ICU, LACTATE, MAGNESIUM, PHOSPHATE, PLATELET, POTASSIUM, PTT, INR, PR, SODIUM, BUN, WBC, HeartRate, DiasBP, SysBP, RespRate, SpO2, Glucose, Temp (in total, $d_X = 31$). The time series data is converted in 48 hour blocks ($T = 48$) by averaging all the measurements over each hour block. The patients with all 48 hour blocks missing for a specific features are excluded, this results in 22,9888 ICU admissions. Mean imputation is used when HeartRate, DiasBP, SysBP, RespRate, SpO2, Glucose, Temp are missing. Forward imputation is used when LACTATE, MAGNESIUM, PHOSPHATE, PLATELET, POTASSIUM, PTT, INR, PR, SODIUM, BUN, WBC are missing. All features are standardized and the label is a mortality probability score in [0, 1]. The resulting dataset is split into a training set (65%), a validation set (15%) and a test set (20%).

Model training The model that we train is a RNN with a single layer made of 200 GRU cells. It is trained for 80 epochs with an Adam optimizer ($\text{lr} = 0.001, \beta_1 = 0.9, \beta_2 = 0.999$ and no weight decay).

Mask fitting For each test patient, we simply fit a mask with $a = 0.1$ by maximizing the cross-entropy loss in the deletion variant formulation of Dynamask. We use the fade-to-moving average perturbation π^m with $W = 48$. The hyperparameters for this optimization procedure are $\eta = 1, \alpha = 1, \lambda_0 = 0.1, \lambda_c = 0, \delta = 1000, N = 1000$.

Runtime Fitting a mask for a given patient takes 3.58 s on average.

Illustrations To illustrate the results of our experiments, we show the 10% most important features for patients that are predicted to die on Figure 9 & 10 and for patients that are predicted to survive on Figure 11 & 12. In each case, we indicate the cross-entropy between the unperturbed and the perturbed prediction, as defined in Definition 7. We note that Dynamask identifies the features that create the biggest shift. Qualitatively, we note that Dynamask seems to focus

Table 1. Influence of perturbation operator.

Acc	π^g	π^m	π^p
π^g	1	.85	.80
π^m	.85	1	.80
π^p	.80	.80	1
AUROC	.90	.90	.86

much more on the input that appear at latter time. This is consistent with the observations in (Ismail et al., 2019): these inputs are the most important for the black-box, since it is trained to predict the mortality after the 48 hours and RNNs have short term memory.

3.6. Influence of the perturbation operator

To study the effect of the perturbation operator choice, we have performed the following experiment: in the set-up of the state experiment, we optimize 100 masks on distinct examples by using a Gaussian blur perturbation $(\pi^g, \sigma_{max} = 1)$ and a fade-to-moving average perturbation $(\pi^m, W = 3)$. We do the same with a fade-to-past average perturbation that only uses past values of the features: $(\pi^p, W = 6)$. For each pair of perturbation operators, we compute the average accuracy between the associated masks (i.e. the fraction of inputs where both masks agree). For each method, we report the AUROC for the identification of true salient inputs. The results are reported in Table 1. We observe that π^g, π^m, π^p generally agree and offer similar performances.

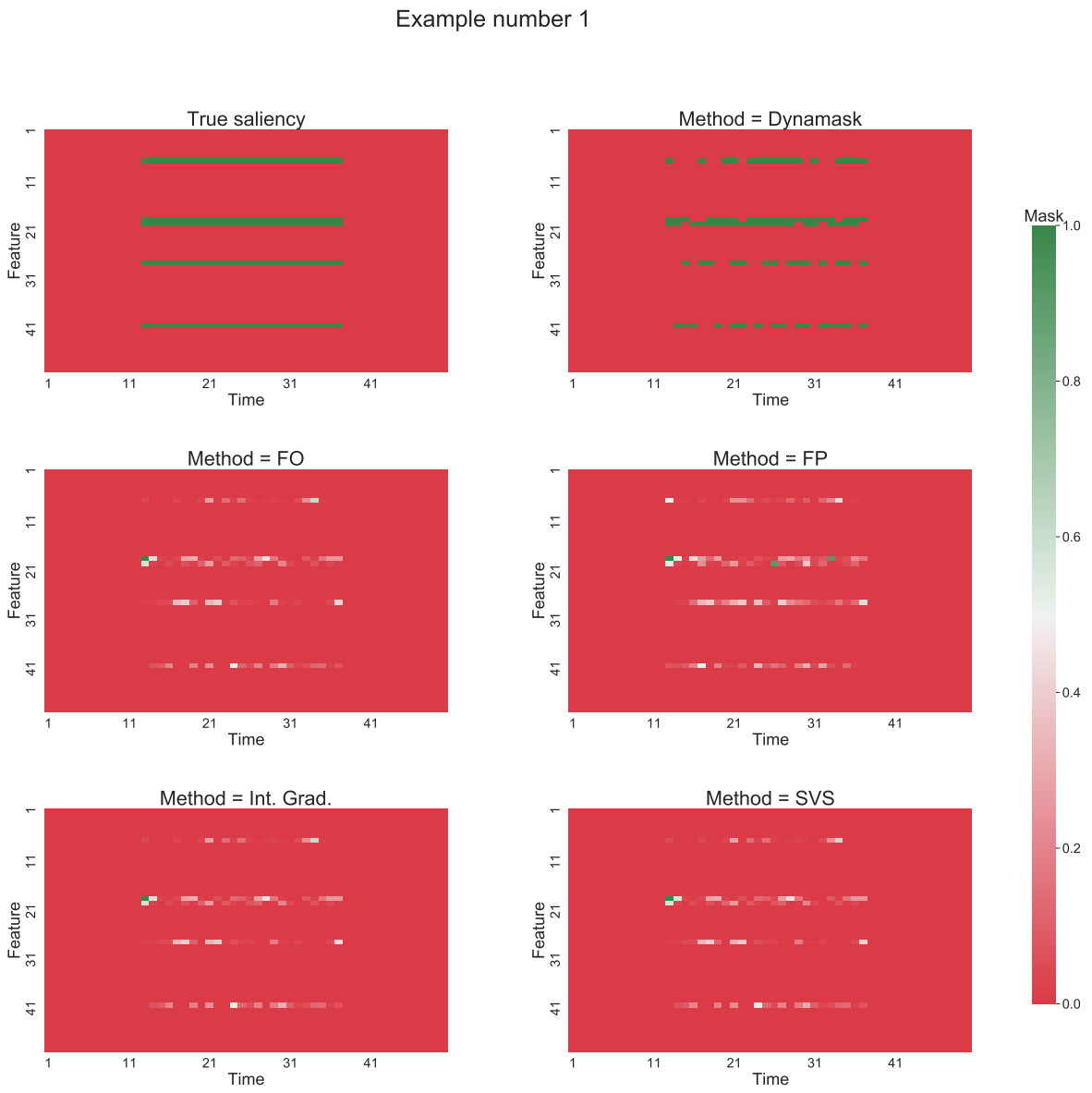

Figure 3. Saliency masks produced by various methods for the test example 1 of the rare feature experiment.

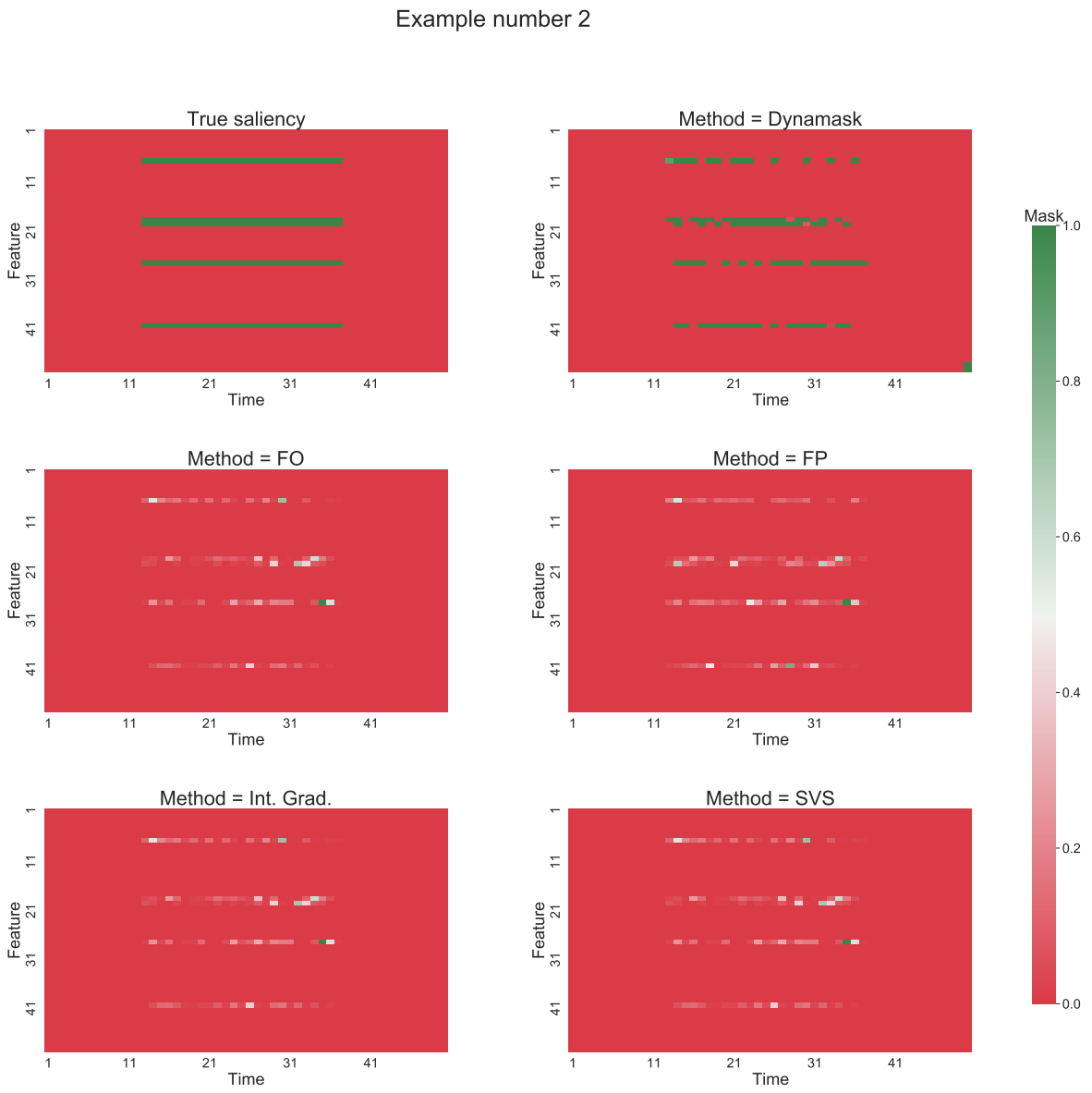

Figure 4. Saliency masks produced by various methods for the test example 2 of the rare feature experiment.

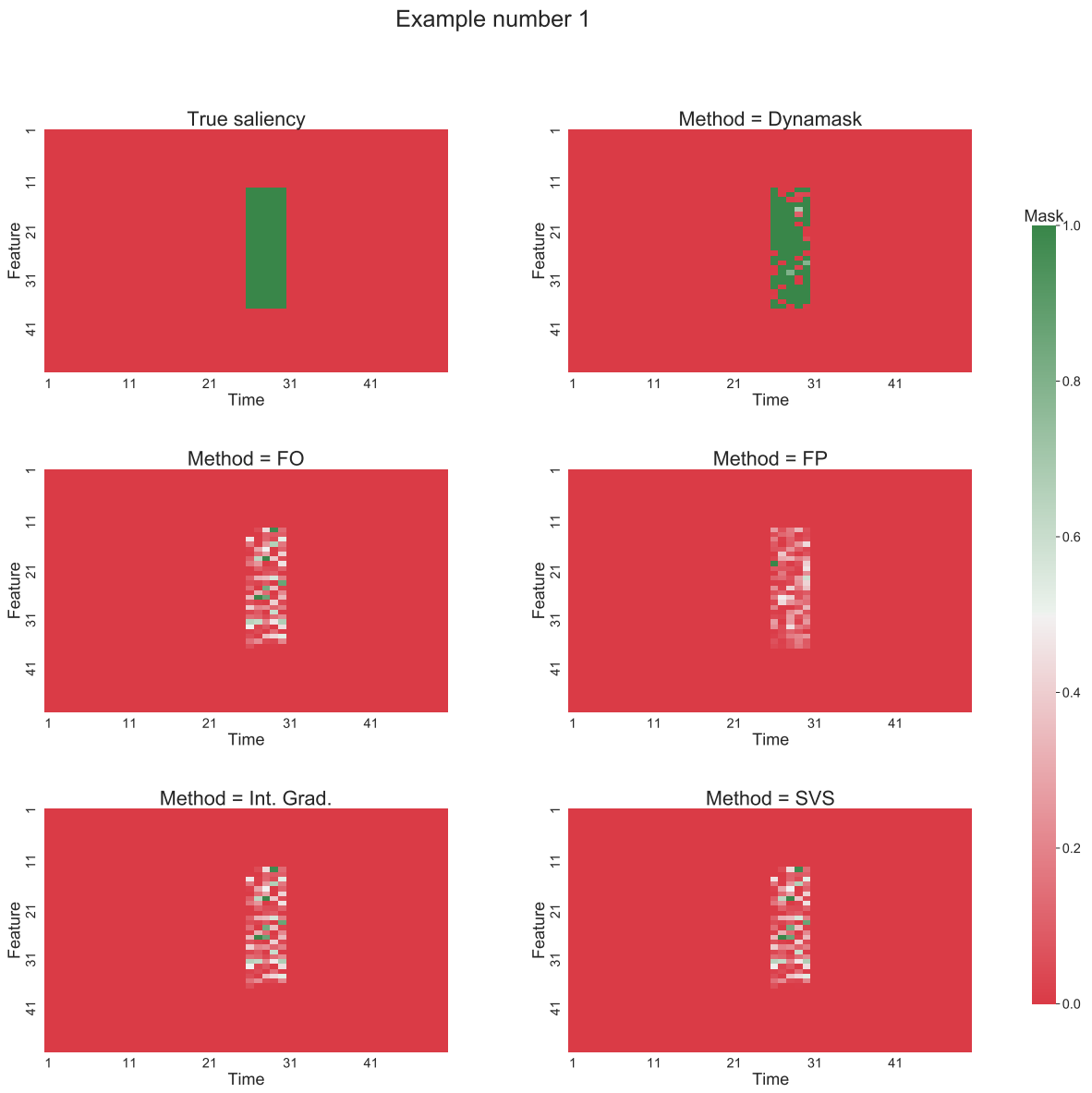

Figure 5. Saliency masks produced by various methods for the test example 1 of the rare time experiment.

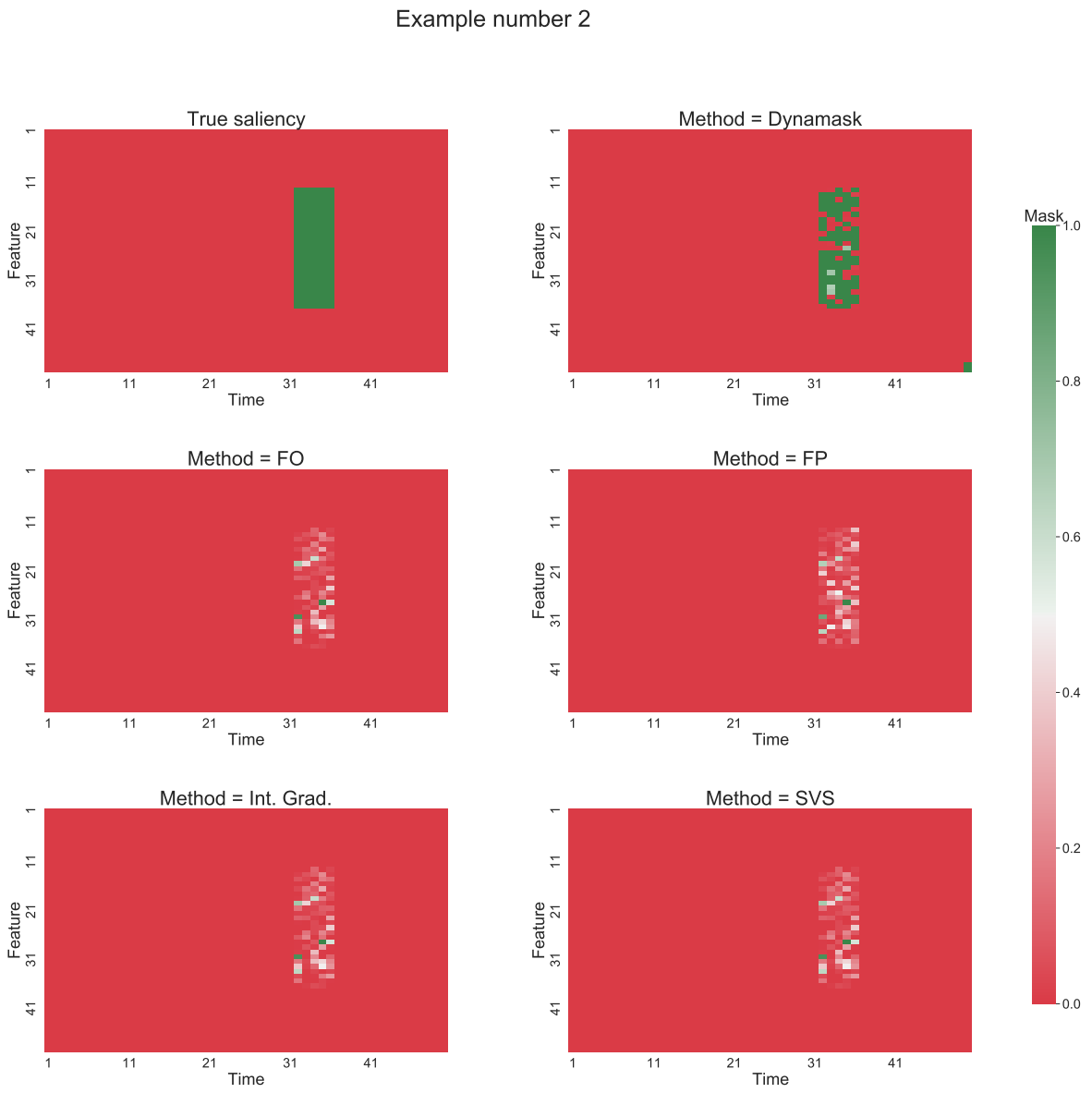

Figure 6. Saliency masks produced by various methods for the test example 2 of the rare time experiment.

Figure 7. Saliency masks produced by various methods for the test example 5 of the state experiment. For each method, the global entropy of the mask $S_{\mathbf{M}}([1:100] \times [1:3])$ is reported.

Figure 8. Saliency masks produced by various methods for the test example 21 of the state experiment. For each method, the global entropy of the mask $S_{\mathbf{M}}([1:100] \times [1:3])$ is reported.

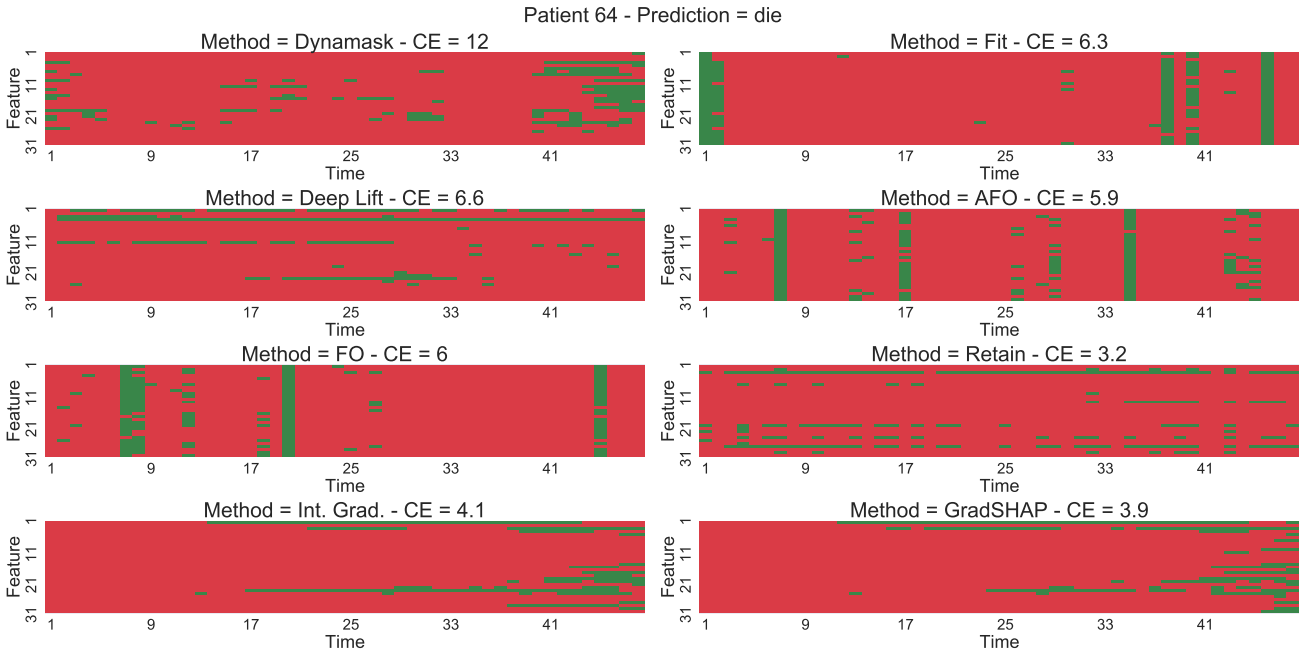

Figure 9. Most important inputs for patient 64. For each saliency method, the 10% most important inputs are represented in green. The black-box predicts that this patient will die. In each case, the cross entropy (CE) between the unperturbed and the perturbed prediction is reported.

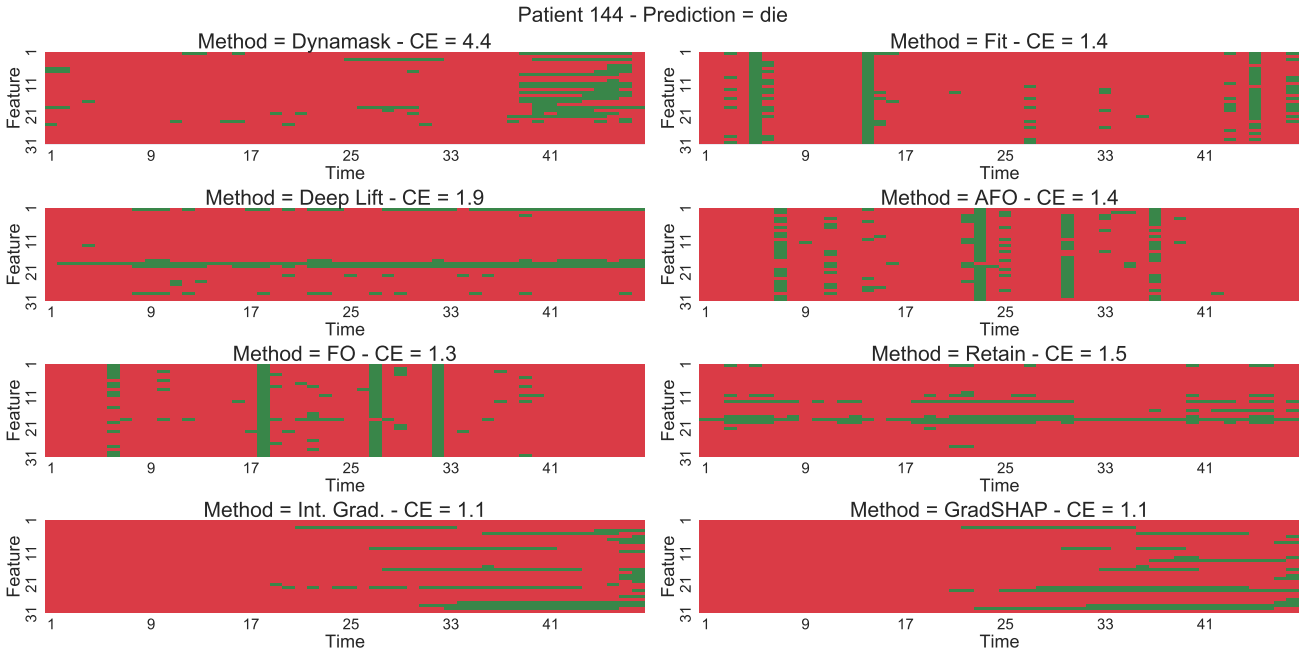

Figure 10. Most important inputs for patient 144. For each saliency method, the 10% most important inputs are represented in green. The black-box predicts that this patient will die. In each case, the cross entropy (CE) between the unperturbed and the perturbed prediction is reported.

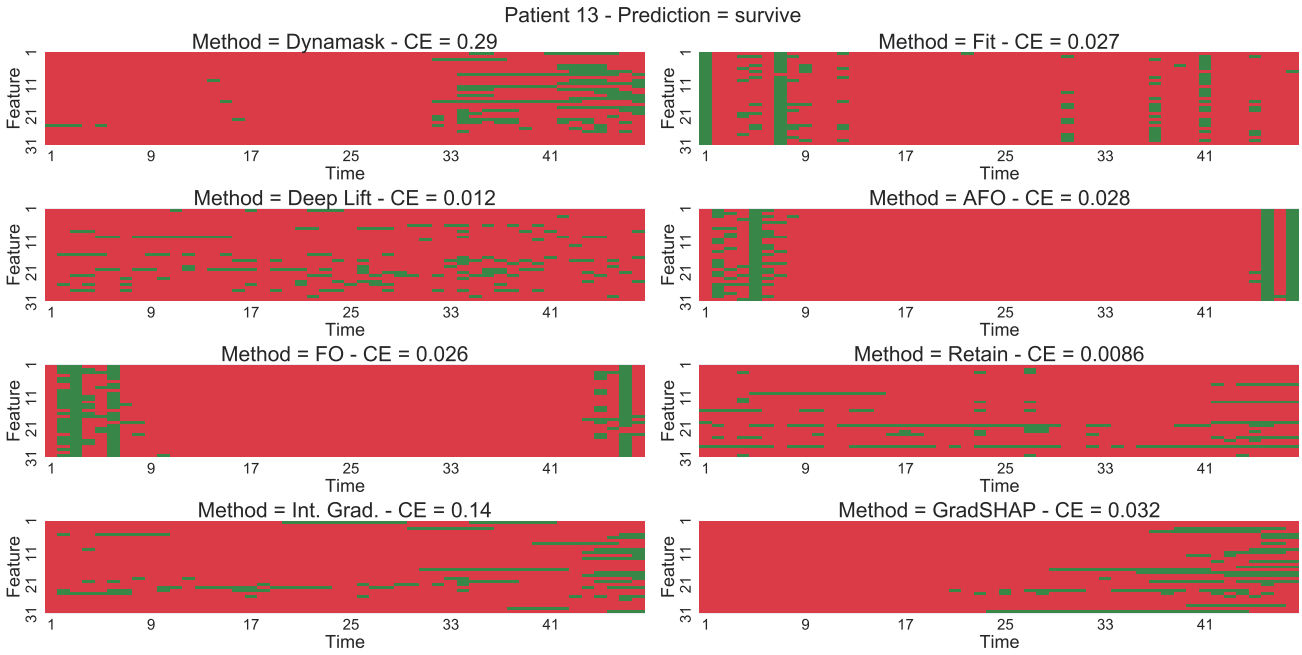

Figure 11. Most important inputs for patient 13. For each saliency method, the 10% most important inputs are represented in green. The black-box predicts that this patient will survive. In each case, the cross entropy (CE) between the unperturbed and the perturbed prediction is reported.

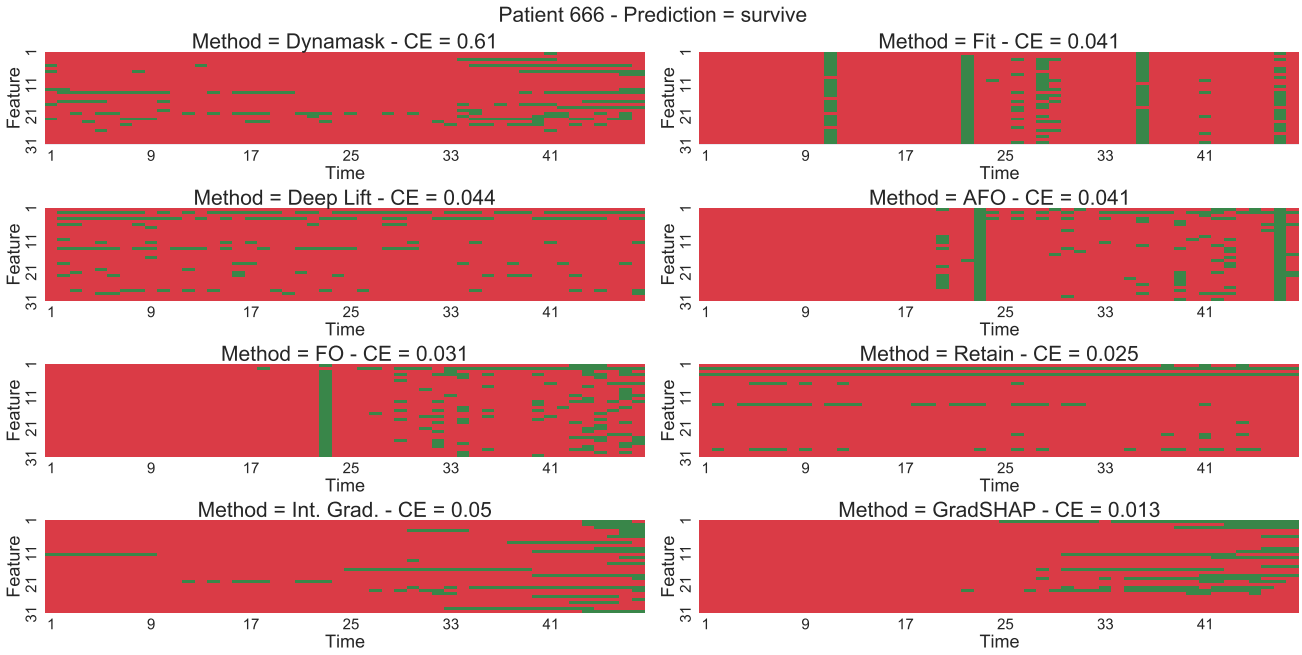

Figure 12. Most important inputs for patient 666. For each saliency method, the 10% most important inputs are represented in green. The black-box predicts that this patient will survive. In each case, the cross entropy (CE) between the unperturbed and the perturbed prediction is reported.

References

Adebayo, J., Gilmer, J., Muelly, M., Goodfellow, I., Hardt, M., and Kim, B. Sanity Checks for Saliency Maps. *Advances in Neural Information Processing Systems*, 2018-December:9505–9515, oct 2018.

Alvarez-Melis, D. and Jaakkola, T. S. On the Robustness of Interpretability Methods. *arXiv*, jun 2018.

Baehrens, D., Harmeling, S., Kawanabe, M., Hansen Khansen, K., and Edward Rasmussen, C. How to Explain Individual Classification Decisions. *Journal of Machine Learning Research*, 11(61):1803–1831, 2010. ISSN 1533-7928.

Barredo Arrieta, A., Díaz-Rodríguez, N., Del Ser, J., Bennetot, A., Tabik, S., Barbado, A., Garcia, S., Gil-Lopez, S., Molina, D., Benjamins, R., Chatila, R., and Herrera, F. Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. *Information Fusion*, 58(December 2019):82–115, 2020. ISSN 15662535. doi: 10.1016/j.inffus.2019.12.012.

Caruana, R., Lou, Y., Gehrke, J., Koch, P., Sturm, M., and Elhadad, N. Intelligible models for healthcare: Predicting pneumonia risk and hospital 30-day readmission. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, volume 2015-Augus, pp. 1721–1730, New York, NY, USA, aug 2015. Association for Computing Machinery. ISBN 9781450336642. doi: 10.1145/2783258.2788613.

Castro, J., Gómez, D., and Tejada, J. Polynomial calculation of the Shapley value based on sampling. *Computers and Operations Research*, 36(5):1726–1730, may 2009. ISSN 03050548. doi: 10.1016/j.cor.2008.04.004.

Chen, J., Song, L., Wainwright, M. J., and Jordan, M. I. Learning to Explain: An Information-Theoretic Perspective on Model Interpretation. *35th International Conference on Machine Learning, ICML 2018*, 2:1386–1418, feb 2018.

Ching, T., Himmelstein, D. S., Beaulieu-Jones, B. K., Kalinin, A. A., Do, B. T., Way, G. P., Ferrero, E., Agapow, P.-M., Zietz, M., Hoffman, M. M., Xie, W., Rosen, G. L., Lengerich, B. J., Israeli, J., Lanchantin, J., Woloszynek, S., Carpenter, A. E., Shrikumar, A., Xu, J., Cofer, E. M., Lavender, C. A., Turaga, S. C., Alexandari, A. M., Lu, Z., Harris, D. J., DeCaprio, D., Qi, Y., Kundaje, A., Peng, Y., Wiley, L. K., Segler, M. H. S., Boca, S. M., Swamidass, S. J., Huang, A., Gitter, A., and Greene, C. S. Opportunities and obstacles for deep learning in biology and medicine. *Journal of The Royal Society Interface*, 15(141):20170387, apr 2018. ISSN 1742-5689. doi: 10.1098/rsif.2017.0387.

Choi, E., Bahadori, M. T., Kulas, J. A., Schuetz, A., Stewart, W. F., and Sun, J. RETAIN: An Interpretable Predictive Model for Healthcare using Reverse Time Attention Mechanism. *Advances in Neural Information Processing Systems*, pp. 3512–3520, 2016. ISSN 10495258.

Cover, T. M. and Thomas, J. A. *Elements of Information Theory*. Wiley, 2005. ISBN 9780471241959. doi: 10.1002/047174882X.

Das, A. and Rad, P. Opportunities and Challenges in Explainable Artificial Intelligence (XAI): A Survey. *arXiv*, jun 2020.

Fong, R., Patrick, M., and Vedaldi, A. Understanding deep networks via extremal perturbations and smooth masks. *Proceedings of the IEEE International Conference on Computer Vision*, 2019-Octob:2950–2958, 2019. ISSN 15505499. doi: 10.1109/ICCV.2019.00304.

Fong, R. C. and Vedaldi, A. Interpretable Explanations of Black Boxes by Meaningful Perturbation. *Proceedings of the IEEE International Conference on Computer Vision*, 2017-Octob:3449–3457, 2017. ISSN 15505499. doi: 10.1109/ICCV.2017.371.

Gimenez, J. R. and Zou, J. Discovering Conditionally Salient Features with Statistical Guarantees. *36th International Conference on Machine Learning, ICML 2019*, 2019-June:4140–4152, may 2019.

Guo, T., Lin, T., and Antulov-Fantulin, N. Exploring Interpretable LSTM Neural Networks over Multi-Variable Data. *36th International Conference on Machine Learning, ICML 2019*, 2019-June:4424–4440, may 2019.

Ho, L. V., Aczon, M. D., Ledbetter, D., and Wetzel, R. Interpreting a Recurrent Neural Network's Predictions of ICU Mortality Risk. *Journal of Biomedical Informatics*, pp. 103672, may 2019. doi: 10.1016/j.jbi.2021.103672.

Ismail, A. A., Gunady, M., Pessoa, L., Bravo, H. C., and Feizi, S. Input-Cell Attention Reduces Vanishing Saliency of Recurrent Neural Networks. *Advances in Neural Information Processing Systems*, 32, oct 2019.

Ismail, A. A., Gunady, M., Bravo, H. C., and Feizi, S. Benchmarking Deep Learning Interpretability in Time Series Predictions. *Advances in Neural Information Processing Systems*, 2020.

Jain, S. and Wallace, B. C. Attention is not explanation. In *NAACL-HLT*, 2019.

John, G. H., Kohavi, R., and Pfleger, K. Irrelevant Features and the Subset Selection Problem. In *Machine Learning Proceedings 1994*, pp. 121–129. Elsevier, jan 1994. doi: 10.1016/b978-1-55860-335-6.50023-4.

Johnson, A. E., Pollard, T. J., Shen, L., Lehman, L. W. H., Feng, M., Ghassemi, M., Moody, B., Szolovits, P., Anthony Celi, L., and Mark, R. G. MIMIC-III, a freely accessible critical care database. *Scientific Data*, 3(1):1–9, may 2016. ISSN 20524463. doi: 10.1038/sdata.2016.35.

Karpathy, A., Johnson, J., and Fei-fei, L. Visualizing and Understanding Recurrent Networks. In *ICLR*, 2016.

Kindermans, P.-J., Hooker, S., Adebayo, J., Alber, M., Schütt, K. T., Dähne, S., Erhan, D., and Kim, B. The (Un)reliability of saliency methods. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 11700 LNCS:267–280, nov 2017.

Kwon, B. C., Choi, M.-J., Kim, J. T., Choi, E., Kim, Y. B., Kwon, S., Sun, J., and Choo, J. RetainVis: Visual Analytics with Interpretable and Interactive Recurrent Neural Networks on Electronic Medical Records. *IEEE Transactions on Visualization and Computer Graphics*, 25(1): 299–309, may 2018. doi: 10.1109/TVCG.2018.2865027.

Lipton, Z. C. The Mythos of Model Interpretability. *Communications of the ACM*, 61(10):35–43, jun 2016.

Lundberg, S. and Lee, S.-I. A Unified Approach to Interpreting Model Predictions. *Advances in Neural Information Processing Systems*, pp. 4766–4775, 2017.

Lundberg, S. M., Nair, B., Vavilala, M. S., Horibe, M., Eisses, M. J., Adams, T., Liston, D. E., Low, D. K. W., Newman, S. F., Kim, J., and Lee, S. I. Explainable machine-learning predictions for the prevention of hypoxaemia during surgery. *Nature Biomedical Engineering*, 2(10):749–760, oct 2018. ISSN 2157846X. doi: 10.1038/s41551-018-0304-0.

MacKay, D. J. C. *Information Theory, Inference and Learning Algorithms*, volume 13. Press, Cambridge University, 2003. ISBN 0521642981.

Moraffah, R., Karami, M., Guo, R., Raglin, A., and Liu, H. Causal Interpretability for Machine Learning - Problems, Methods and Evaluation. *ACM SIGKDD Explorations Newsletter*, 22(1):18–33, 2020. ISSN 1931-0145. doi: 10.1145/3400051.3400058.

Phillips, L., Goh, G., and Hodas, N. Explanatory Masks for Neural Network Interpretability. In *IJCAI Workshop on Explainable Artificial Intelligence (XAI)*, pp. 1–4, 2018.

Ribeiro, M. T., Singh, S., and Guestrin, C. "why should I trust you?": Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016*, pp. 1135–1144, 2016a.

Ribeiro, M. T., Singh, S., and Guestrin, C. Model-agnostic interpretability of machine learning. *ArXiv*, abs/1606.05386, 2016b.

Shannon, C. E. A Mathematical Theory of Communication. *Bell System Technical Journal*, 27(4):623–656, 1948. ISSN 00058580. doi: 10.1002/j.1538-7305.1948.tb00917.x.

Shrikumar, A., Greenside, P., and Kundaje, A. Learning Important Features Through Propagating Activation Differences. *34th International Conference on Machine Learning, ICML 2017*, 7:4844–4866, apr 2017.

Siddiqui, S. A., Mercier, D., Munir, M., Dengel, A., and Ahmed, S. TSViz: Demystification of Deep Learning Models for Time-Series Analysis. *IEEE Access*, 7:67027–67040, feb 2019. ISSN 21693536. doi: 10.1109/ACCESS.2019.2912823.

Simonyan, K., Vedaldi, A., and Zisserman, A. Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps. *2nd International Conference on Learning Representations, ICLR 2014 - Workshop Track Proceedings*, dec 2013.

Song, H., Rajan, D., Thiagarajan, J. J., and Spanias, A. Attend and Diagnose: Clinical Time Series Analysis using Attention Models. *32nd AAAI Conference on Artificial Intelligence, AAAI 2018*, pp. 4091–4098, nov 2017.

Sundararajan, M., Taly, A., and Yan, Q. Axiomatic Attribution for Deep Networks. *34th International Conference on Machine Learning, ICML 2017*, 7:5109–5118, mar 2017.

Suresh, H., Hunt, N., Johnson, A., Celi, L. A., Szolovits, P., and Ghassemi, M. Clinical Intervention Prediction and Understanding using Deep Networks. *arXiv*, 2017.

Tjoa, E. and Guan, C. A Survey on Explainable Artificial Intelligence (XAI): Toward Medical XAI. *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–21, 2020. ISSN 2162-237X. doi: 10.1109/tnnls.2020.3027314.

Tonekaboni, S., Joshi, S., Campbell, K., Duvenaud, D., and Goldenberg, A. What went wrong and when? Instance-wise Feature Importance for Time-series Models. In *Advances in Neural Information Processing Systems*, 2020.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 2017-December, pp. 5999–6009. Neural information processing systems foundation, jun 2017.

Xu, Y., Biswal, S., Deshpande, S. R., Maher, K. O., and Sun, J. RAIM: Recurrent Attentive and Intensive Model of Multimodal Patient Monitoring Data. *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 18:2565–2573, jul 2018.

Yang, Y., Tresp, V., Wunderle, M., and Fasching, P. A. Explaining therapy predictions with layer-wise relevance propagation in neural networks. In *Proceedings - 2018 IEEE International Conference on Healthcare Informatics, ICHI 2018*, pp. 152–162. Institute of Electrical and Electronics Engineers Inc., jul 2018. ISBN 9781538653777. doi: 10.1109/ICHI.2018.00025.

Yoon, J., Jordon, J., and Van Der Schaar, M. INVASE: Instance-wise Variable Selection using Neural Networks. In *ICLR*, 2019a.

Yoon, J., Jordon, J., and van der Schaar, M. ASAC: Active Sensing using Actor-Critic models. *Machine Learning for Healthcare Conference*, pp. 1–18, jun 2019b.

Zhang, Q. and Zhu, S.-C. Visual Interpretability for Deep Learning: a Survey. *Frontiers of Information Technology and Electronic Engineering*, 19(1):27–39, feb 2018.