Vol.2021-NL-248 No.18
Vol.2021-CVIM-226 No.18
2021/5/21

行動予測モデルの判断根拠の可視化

加藤 雅大[1,a)]　浮田 宗伯[1,b)]

概要：医療や監視カメラシステム，自動運転などへのコンピュータビジョン技術の応用に向けて，モデルがなぜそのような判断をするのかといった根拠を明確にする説明可能な AI に大きな関心が寄せられている．しかし，予測モデルや生成モデルの可視化研究は CNN モデルの可視化研究に比べて未開拓である．本研究ではその未開拓な分野に先駆け，行動予測モデルの一種である動画予測モデルの判断根拠の可視化を研究する．多くの動画予測モデルでは LSTM のような再帰型ネットワークが用いられ，その可視化手法が重要である．しかし，従来の LSTM の可視化手法では，動体の予測の判断根拠なのか，背景の予測の判断根拠なのか区別して可視化できないという問題がある．本研究では，それぞれの判断根拠を区別して可視化できるようにモデル構造を変更し，動体の予測に対する判断根拠を可視化する．これより，動画予測モデルが画像のどの領域を重要視して動体予測しているか解釈する．実験では，自作したデータセットを用いて動体予測のためにモデルはどこを重要視して予測しているかを可視化した．その結果，動画予測モデルは直感通り過去の動体領域を重要視して動体の予測を行なっていることが可視化された．

キーワード：行動予測，LSTM，説明可能な AI

1. はじめに

近年の深層学習の進歩により，医療や監視カメラシステム，自動運転などへのコンピュータビジョン技術の応用が進んでいる．このような実用的で安全性が求められる技術では，その性能を保証することに加え，出力の背後にある根拠を説明できることが重要である．例えば，自動運転では行動予測モデルの技術が応用されている．このような行動予測モデルが，過去の人の行動や物体の挙動のどこを重要視して予測を行ったかを可視化できれば，どのような根拠に基づいて予測を行っていたか解釈できる．従って行動予測モデルの判断根拠の可視化が達成できれば，自動運転車による事故が発生した場合，その自動運転車の改善箇所が明らかになるため，交通事故削減に繋がる．本研究では行動予測モデルの一種である動画予測モデルの判断根拠の可視化を研究する．

近年では，モデルの判断根拠を明確にする手法に大きな関心が寄せられている．その中でも畳み込みニューラルネットワーク（Convolutional Neural Network；CNN）モデルでは根拠を可視化する様々な手法が提案されている [1], [2], [3], [4]．これらの技術は，アテンションマップを生成することによって，分類モデルが画像中のどこを注視して，なぜその物体がそのクラスに分類されるのかを解釈可能にした．しかし，本研究が扱う予測タスク，画像生成タスクの可視化研究は，分類モデルの可視化研究に比べて，未開拓である．

本研究ではその未開拓な分野に先駆け，既存研究である長短期記憶（Long Short-Term Memory；LSTM）の判断根拠の可視化手法 [5] を，行動予測モデルへ応用する．既存研究 [5] では，従来の LSTM では内部で入力データの各情報が混合され，入力データの各情報が出力へどれほど寄与したかを正確に計算できないという問題を解決した．具体的には LSTM 内部で入力データの各情報を独立に処理し，入力空間の重要度を正確に計算可能にした．しかし，動画予測モデルの可視化には，既存手法 [5] では不十分である．なぜなら、予測画像全体を生成するための重要領域が可視化されるため、動体部分と背景部分への寄与を区別できないからである。そこで本研究では，予測画像の各領域を予測するための重要領域を可視化する．動体を含む領域の重要領域を足し合わせることで，動画予測モデルがなぜそのように動体を予測したかをアテンションマップを用いて可視化する．図 1 に動画予測モデルが時刻 t までの画像から時刻 t+1 の画像を予測する場合を考える．パターン A とパターン B では時刻 t-1 までは同じ動きをしている

[1] 豊田工業大学
Toyota Technological Institute, Nagoya, Aichi 468-8511, Japan
[a)] sd21414@toyota-ti.ac.jp
[b)] ukita@toyota-ti.ac.jp

Vol.2021-NL-248 No.18
Vol.2021-CVIM-226 No.18
2021/5/21

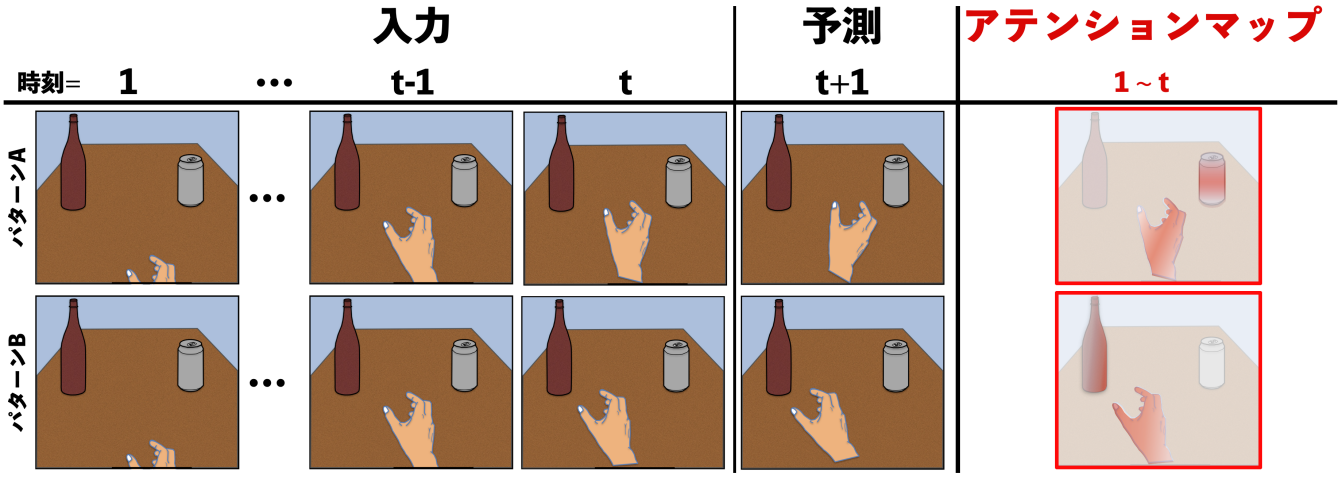

図 1 本研究の最終目標の模式図．動画予測モデルがなぜそのように動体を予測したのかをアテンションマップを用いて可視化する．

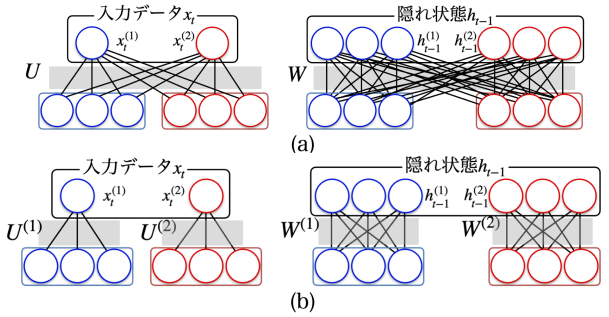

図 2 RNN 内の入力データと隠れ状態の処理方法の模式図．（a）通常の RNN の処理．（b）[5] が提案した RNN の処理．x_t は時刻 t の入力データ，h_{t-1} は時刻 $t-1$ の隠れ状態である．U と W はそれぞれ入力データ x_t と隠れ状態 h_{t-1} に対する重みである．上付き文字（1）と（2）はそれぞれが変数（1）と変数（2）に関する情報であることを表し，青色は変数（1）の情報で赤色は変数（2）の情報を表す．変数毎の情報が混合せず更新することで，各変数の正確な重要度を計算可能にした．

が時刻 t では異なっており，それに伴い動画予測モデルは別々の動きを予測した画像を生成する．パターン A では時刻 t で手が缶のある右側に回転しているため，缶の方向へ手を伸ばすと予測している．そのため，重要領域を可視化したアテンションマップでは，缶と手の周辺が重要視している．逆にパターン B では時刻 t で瓶の方向に手を伸ばしており，モデルは瓶の方向へ手を伸ばすと予測しているので，アテンションマップでは瓶と手を重要視している．このように予測画像が異なる時，アテンションマップも大きく異なることが期待される．また，本手法は行動予測モデルに限らず，LSTM を扱う他タスクのために学習されたモデルの可視化も可能である．

2. 関連研究

本節では，既存研究として，行動予測モデルとの関連が強い LSTM のような再帰型ニューラルネットワーク（Reccurent Neural Network；RNN）の可視化手法について述べる．RNN の可視化手法では，予測後の解析によって可視化する手法と，予測と同時に判断根拠を可視化する手法の二つに分類できる．

予測後の解析により可視化する手法 [6], [7], [8] では，時刻毎の隠れ状態を保存したメモリセルの情報から事後解析を行い，時間方向の重要度を求めた．しかしこれらの手法では、殆どが時間方向の重要度に焦点を当てており、入力空間のどの変数が重要であったかを解釈できない．

予測と同時に判断根拠を可視化する手法 [5], [9], [10], [11] では，隠れ状態に対して同時に求めた重要領域を掛け合わせ，重要領域が強調された隠れ状態を用いて学習する事で，予測に対する重要領域を得る．しかし，手法 [9], [10], [11] では，RNN の隠れ状態において入力変数の全ての情報が混合しているため，入力変数毎の正確な重要度を獲得することはできない．例えば，未来の気温を予測する際に，入力変数である過去の気温・天気・気圧それぞれの予測に対する重要度を可視化する場合を考える．RNN の隠れ状態では気温・天気・気圧の情報が混合するため，それぞれに対する予測への正確な寄与は計算できない．また，動画予測モデルの判断根拠の可視化においても同様の問題が発生するため，どの画素の情報が予測に寄与したかを正確に可視化できない．

そこで手法 [5] では，RNN の構造を変更することで，隠れ状態を更新する際に，各入力変数が持つ情報が混ざらないようにした．その結果，正確な入力変数毎の重要度を解釈可能にした．隠れ状態を更新する際の RNN の処理方法の模式図を図 2 に示す．従来の RNN の処理では，図 2（a）のように，変数（1）と変数（2）の情報が重み W，U を用いた行列計算により混合される．しかし [5] が提案した手法では，図 2（b）のように，変数（1）と変数（2）の情報に対してそれぞれに重み $W^{(1)}$，$U^{(1)}$ と $W^{(2)}$，$U^{(2)}$ を用いて

Vol.2021-NL-248 No.18
Vol.2021-CVIM-226 No.18
2021/5/21

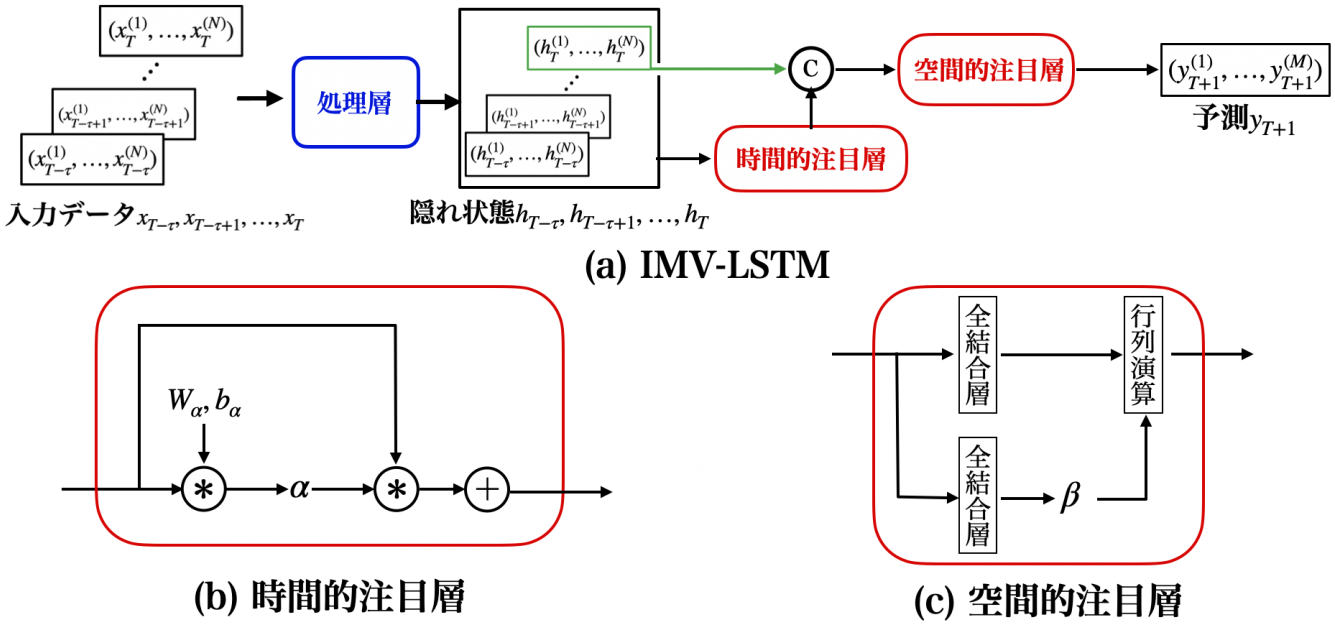

図 3 IMV-LSTM の構造．(a) IMV-LSTM の全体構造．時刻 $T-\tau$ から時刻 T の系列データを受け取り，時刻 $T+1$ のデータを予測する．また，各時刻の入力は N 個の変数を持つとする．処理層 (3.1 節) では，入力した系列データを処理し，それぞれの隠れ状態を出力する．ⓒは複数のデータを結合する処理，㊋は複数のデータの入力変数軸に沿った積を行う処理，⊕は複数のデータに対して要素和を行う処理を表す．(b) 時間的注目層 (3.2 節) の構造．時間的注目層では，系列データの時空間的な重要度を表す α を求める．W_α や b_α は，重要度 α を求める際に使われる行列パラメータである．(c) 空間的注目層 (3.3 節) の構造．空間的注目層では，入力した系列データの時間情報を総括した入力空間の重要度を表す β を求める．

計算することで，変数 (1) と変数 (2) の情報が混ざらずに隠れ状態を更新できる．また [5] では，メモリセルに対して，同時に求めた各時刻に対応する重要領域を掛け合わせ，各時刻の重要な領域が強調された隠れ状態を用いて予測した．これにより，空間的な重要度だけでなく，時間的な重要度を可視化した．またこの手法 [5] で提案された LSTM は Interpretable Multi-Variable LSTM（IMV-LSTM）と呼ばれ，本研究ではこの IMV-LSTM を動画予測モデルの可視化に応用する．

3. IMV-LSTM

IMV-LSTM とは，既存手法 [5] で提案された LSTM であり，隠れ状態が個々の入力変数の情報を別々に持つことができる内部構造を持つ．その結果，入力変数毎の正確な重要度を計算できる．IMV-LSTM では，入力した系列データを処理する処理層 (3.1 節) と，系列データの時空間的な重要度を求める時間的注目層（3.2 節），時間情報を総括した空間的な重要度を求める空間的注目層（3.3 節）からなる（図 3）．ここで，時刻 $T-\tau$ から時刻 T の系列データ $[\boldsymbol{x}_{T-\tau}, \ldots, \boldsymbol{x}_T]$ を受け取り，時刻 $T+1$ の予測値 $\boldsymbol{y}_{T+1}$ を出力する場合を考える．この際，入力データ $\boldsymbol{x}_t$ は N 次元ベクトルであり，予測値 $\boldsymbol{y}_{T+1}$ は M 次元ベクトルとする．また，τ とは，入力する系列データのフレーム数を表す．

3.1 処理層

処理層では，時刻 $T-\tau$ から時刻 T の系列データが入力され，隠れ状態を各系列それぞれに対して出力する．IMV-LSTM の処理層では，標準的な LSTM と同様にメモリセル c_t，入力ゲート i_t，出力ゲート o_t，忘却ゲート f_t を持つ．

時刻 t の入力データ $\boldsymbol{x_t}$ を構成する変数が N 個あるとき，$\boldsymbol{x_t}=[x_t^{(1)}, \ldots, x_t^{(N)}]$ の N 次元ベクトルで表現される（$\boldsymbol{x_t} \in \mathbb{R}^N$）．また，時刻 t における隠れ状態を $h_t = [\boldsymbol{h_t^{(1)}}, \ldots, \boldsymbol{h_t^{(N)}}]$，変数 n に対応する隠れ状態を $\boldsymbol{h_t^{(n)}}$ と定義する（$h_t \in \mathbb{R}^{N\times d}$，$\boldsymbol{h_t^{(n)}} \in \mathbb{R}^d$）．但し，d は隠れ状態の中で各変数が持つ次元数を表す．入力データから隠れ状態空間への写像行列を，$U=[\boldsymbol{U^{(1)}}, \ldots, \boldsymbol{U^{(N)}}]$，入力変数 n から隠れ状態空間への写像を $\boldsymbol{U^{(n)}}$ と定義する（$U \in \mathbb{R}^{N\times d}$，$\boldsymbol{U^{(n)}} \in \mathbb{R}^d$）．

式 (1) では，隠れ状態の更新に使われる j_t を求める．$j_t = [\boldsymbol{j_t^{(1)}}, \ldots, \boldsymbol{j_t^{(N)}}]$ は、隠れ状態と同じ大きさを持つ（$j_t \in \mathbb{R}^{N\times d}$）． 各要素 $\boldsymbol{j_t^{(n)}} \in \mathbb{R}^d$ は、入力変数 n の隠れ状態の更新に対応する。ここで，前時刻の隠れ状態からの遷移を表す W_j は $W_j=[W_j^{(1)}, \ldots, W_j^{(N)}]$，隠れ状態の変

数 n のデータの遷移は $W_j^{(n)}$ と定義する（$W_j \in \mathbb{R}^{N\times d\times d}$, $W_j^{(n)} \in \mathbb{R}^{d\times d}$）．従って，項 $W_j ⊛ h_{t-1}$ および $U_j ⊛ x_t$ は，それぞれ前時刻の隠れ状態と新しい入力からの更新を表す．また⊛演算とは、入力変数軸に沿った積として定義され，例えば，$W_j ⊛ h_{t-1}$ の計算は式（2）のように表現される．このような演算を行うことで，図 2（b）のように変数毎の情報が混合しない更新方法を実現している．また，バイアス b_j は入力や隠れ状態が持つ情報の遷移しやすさ，伝達しやすさを表す．

$$
\begin{aligned}
j_t &= \tanh(W_j ⊛ h_{t-1} + U_j ⊛ \boldsymbol{x}_t + b_j) \qquad (1)\\
W_j &= \left[\ W_j^{(1)}, W_j^{(2)}, \ldots, W_j^{(N)}\ \right]\\
h_j &= \left[\ \boldsymbol{h_j^{(1)}}, \boldsymbol{h_j^{(2)}}, \ldots, \boldsymbol{h_j^{(N)}}\ \right]\\
W_j ⊛ h_{t-1} &= \left[\ \boldsymbol{h_j^{(1)}} \cdot W_j^{(1)}, \ldots, \boldsymbol{h_j^{(N)}} \cdot W_j^{(N)}\ \right] \qquad (2)
\end{aligned}
$$

また，各ゲートやメモリセルに対しても式 (3)，式（4）のように更新される．⊙ は複数のデータに対して要素積を行う処理を表す．ここで i_t， f_t， o_t， c_t は隠れ状態と同じ大きさ（$\mathbb{R}^{N\times d}$）であり，遷移を表す W_i，W_f，W_o は $\mathbb{R}^{N\times d\times d}$，写像を表す U_i，U_f， U_o は $\mathbb{R}^{N\times d}$，バイアス b_i，b_f， b_o は $\mathbb{R}^{N\times d}$ である．j_t，i_t， f_t， o_t， c_t から式（5）のように時刻 t の隠れ状態 h_t が求められる．

$$
s(x) = \frac{1}{1+e^{-x}}
$$

$$
\begin{cases}
i_t = s\left(W_i ⊛ h_{t-1} + U_i ⊛ \boldsymbol{x}_t + b_i\right)\\
f_t = s\left(W_f ⊛ h_{t-1} + U_f ⊛ \boldsymbol{x}_t + b_f\right)\\
o_t = s\left(W_o ⊛ h_{t-1} + U_o ⊛ \boldsymbol{x}_t + b_o\right)
\end{cases} \qquad (3)
$$

$$
c_t = f_t ⊙ c_{t-1} + i_t ⊙ j_t \qquad (4)
$$

$$
h_t = o_t ⊙ \tanh(c_t) \qquad (5)
$$

処理層では，先述した方法で求めた時刻 $T-\tau$ から時刻 T に対応する隠れ状態 $h_{T-\tau}$，…，h_T を出力し，それを時間的注目層や空間的注目層の入力とする．時間的注目層や空間的注目層では，処理層が出力した各時刻の隠れ状態から，系列データの時空間的な重要度や，時間情報を総括した入力空間の重要度を求める．

3.2 時間的注目層

時間的注目層では，入力した系列データの時空間的な重要度を表す α を求める．時間的注目層の構造を図 3（b）に示す．α は α=[$\boldsymbol{\alpha_{T-\tau}}$，…，$\boldsymbol{\alpha_T}$] と表現でき，$\boldsymbol{\alpha_t}$ は時刻 t でどの変数が重要かを表す（$\alpha \in \mathbb{R}^{\tau\times N}$，$\boldsymbol{\alpha_t} \in \mathbb{R}^N$）．$\boldsymbol{\alpha_t}$ の各要素は時刻 t における入力空間の重要度が格納され，$\boldsymbol{\alpha_t}$ は式（6）のように求められる（$W_\alpha \in \mathbb{R}^{N\times d}$，$\boldsymbol{b_\alpha} \in \mathbb{R}^N$）．

次に，このように求めた $\boldsymbol{\alpha_t}$ と対応する隠れ状態 h_t を式（7）のように対応する変数毎に掛け合わせ，重要度が考慮された時刻 t の隠れ状態 $\dot{h}_t$ を得る（$\dot{h}_t \in \mathbb{R}^{N\times d}$）．その処理を各時刻の隠れ状態に対して行うことで，重要度が考慮された隠れ状態 $\dot{h}_{T-\tau}$，…，$\dot{h}_T$ を得る．また，重要度が考慮された隠れ状態 $\dot{h}_t$ を式（8）のように時間方向に要素和を行う．そうすることで，時空間的な重要度を総括した一つの隠れ状態 $\dot{h}$ を得る（$\dot{h} \in \mathbb{R}^{N\times d}$）．

$$
\begin{aligned}
\sigma(x_i) &= \frac{e^{x_i}}{\sum_{k=1}^{N} e^{x_k}}\\
\boldsymbol{\alpha_t} &= \sigma(W_\alpha ⊛ h_t + \boldsymbol{b}_\alpha) \qquad (6)\\
\dot{h}_t &= \boldsymbol{\alpha_t} \cdot h_t\\
&= [\alpha_t^{(1)} \cdot \boldsymbol{h_t^{(1)}}, \ldots, \alpha_t^{(N)} \cdot \boldsymbol{h_t^{(N)}}] \qquad (7)\\
\dot{h} &= \sum_{k=T-\tau}^{T} \dot{h}_k \qquad (8)
\end{aligned}
$$

時間的注目層では，時空間的な重要度を表す α を求め，その重要度を考慮した隠れ状態 $\dot{h}$ を出力する．こうして得た $\dot{h}$ と，時刻 T の隠れ状態 h_T の情報を各変数に対して結合することで，予測において一番重要だと考えられる直近のデータ（h_T）も格納する．結合して得た隠れ状態を g と定義する（$g \in \mathbb{R}^{N\times 2d}$）．そして，隠れ状態 g を空間的注目層に入力する．

3.3 空間的注目層

空間的注目層では，時間情報を総括した空間的な重要度を表す β を求め，予測した時刻 T+1 のデータ $\boldsymbol{y}_{T+1}$ を出力する．M 個の変数を予測するとき，予測データ $\boldsymbol{y}_{T+1}$ は M 次元ベクトルである（$\boldsymbol{y_{T+1}} \in \mathbb{R}^M$）．空間的注目層の構造を図 3（c）に示す．空間的注目層では，隠れ状態 g から β を求めるパスと，隠れ状態 g を出力と対応させるパスがある．隠れ状態 g から β を求めるパスでは，式（9）によって $\boldsymbol{\beta}$ を求める（$\boldsymbol{\beta} \in \mathbb{R}^N$）．この際，重要度を計算する $W_\beta \in \mathbb{R}^{N\times 2d}$，$\boldsymbol{b_\beta} \in \mathbb{R}^N$ を用いて行列の積和演算（全結合層）を用いて計算される．隠れ状態 g を出力と対応させた隠れ状態を得るパスでも同様に，全結合層を用いて，出力の大きさへ変換された隠れ状態 $\dot{g} \in \mathbb{R}^{N\times M}$ を求める．この際，隠れ状態から出力への写像を表す $W_g \in \mathbb{R}^{N\times 2d\times M}$，$b_g \in \mathbb{R}^{N\times M}$ を用いて計算される．最後に $\dot{g}$ と $\boldsymbol{\beta}$ の積から最終出力 $\boldsymbol{y}_{T+1}$ を求める（式（10））．

$$
\boldsymbol{\beta} = \mathrm{softmax}(\mathrm{g}\cdot \mathrm{W}_\beta + \boldsymbol{b_\beta}) \qquad (9)
$$

$$
\boldsymbol{y_{T+1}} = \boldsymbol{\beta} \cdot \dot{g} \qquad (10)
$$

全てのパラメータ（$W^\forall$，$U^\forall$，$b^\forall$）は，$\boldsymbol{y}_{T+1}$ が実際の値と同じになるように学習することで，最適化される．

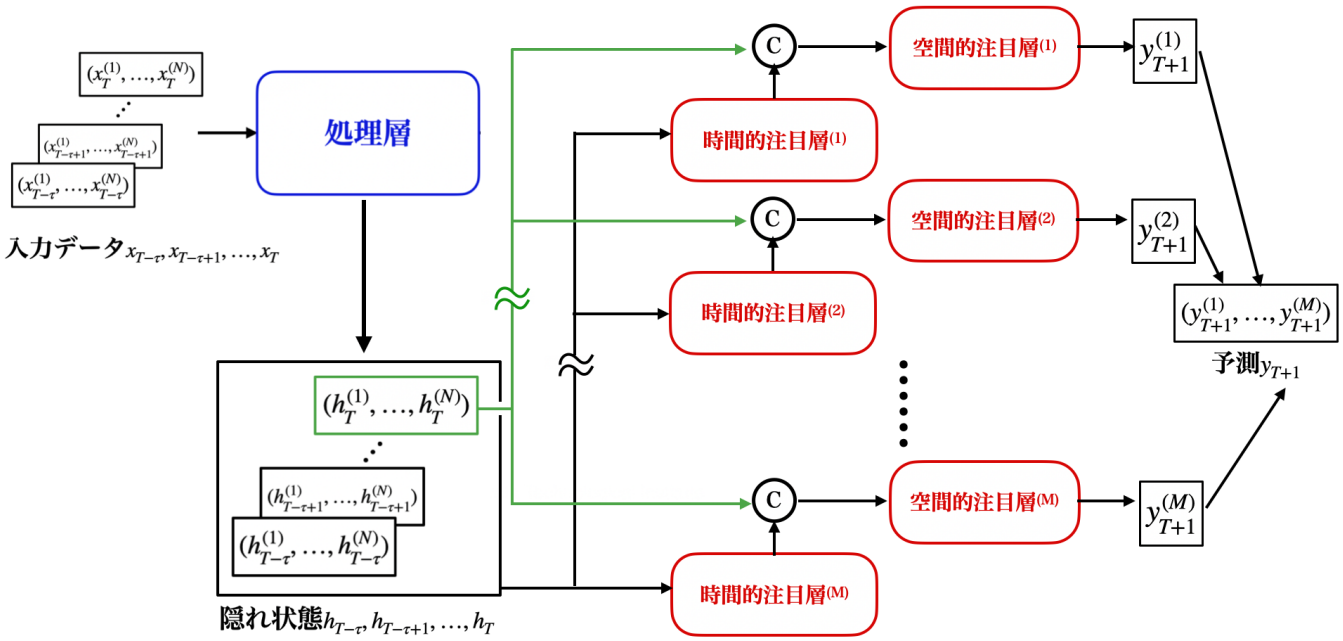

図 4 提案する可視化モデル図（IMV-LSTM+）．予測値 y_{T+1} の各変数 $y^{(1)}_{T+1}, \ldots, y^{(M)}_{T+1}$ それぞれに対応する時間的注目層と空間的注目層を用意することで，各変数を予測するための重要領域を可視化する．図の様に予測画像が大きく異なると，アテンションマップも大きく異なることが期待される．

4. 提案手法

本研究では動画予測モデルに IMV-LSTM を応用することで，動画予測モデルの判断根拠を可視化する．しかし IMV-LSTM を直接用いるのは問題がある．IMV-LSTM では，モデルがあるデータ $\boldsymbol{y}_{T+1}$ を予測するに当たって，重要視した領域（α, $\boldsymbol{\beta}$）を可視化する．しかし，$\boldsymbol{y}_{T+1}$ 全体を予測するための重要領域は可視化できるが，$\boldsymbol{y}_{T+1}$ の各変数 $y^{(1)}_{T+1}, \cdots, y^{(M)}_{T+1}$ を予測するための重要領域は可視化できない．時空間的な重要領域と空間的な重要領域共に同じことがいえる．例えば動画予測モデルが予測を行なった際，予測の判断根拠を可視化する場合を考える．IMV-LSTM では背景の予測にも動体の予測にも同じ重要領域が計算される．得られる重要領域は画像全体を予測するための重要領域を表しており，動体を予測するために重要なのか，背景を予測するために重要なのか区別できない．そのため，動体予測の根拠が正確に解釈できない．本研究では動体予測のための重要領域を適切に得るために，出力の各変数を予測するための重要領域を可視化する手法を提案する．

各変数を予測するための重要領域を可視化できるモデルとして，図 4 のモデルを提案する．図 4 のように予測変数毎に時間的注目層と空間的注目層を用意する．予測変数 m に対応する時間的注目層と空間的注目層を時間的注目層 $^{(m)}$ と空間的注目層 $^{(m)}$ と定義する．この際，時間的注目層 $^{(m)}$ と空間的注目層 $^{(m)}$ は，y_{T+1} の変数 m に対応する $y^{(m)}_{T+1}$ を出力するように学習する．そして同時に時間的注目層 $^{(m)}$ と空間的注目層 $^{(m)}$ は $y^{(m)}_{T+1}$ を出力するための重要領域 $\alpha^{(m)}$, $\boldsymbol{\beta}^{(m)}$ を学習する．従って，$y^{(m)}_{T+1}$ を予測するための重要領域 $\alpha^{(m)}$, $\boldsymbol{\beta}^{(m)}$ を可視化できる．そのため，$y^{(m)}_{T+1}$ が動体部分を予測しているならば，求めた $\alpha^{(m)}$, $\boldsymbol{\beta}^{(m)}$ は，$y^{(m)}_{T+1}$ の対応する動体部分を予測するのに重要視した領域となる．予測変数毎に時間的注目層と空間的注目層を用意する事で，対応する予測変数を予測するために重要な領域を得る．また，このように予測変数毎の重要領域の可視化を可能にした IMV-LSTM を IMV-LSTM+と定義する．

本研究で提案する IMV-LSTM+は，動画予測タスクに限らず，LSTM を用いる様々なタスクに対しても有効である．例えば，過去の気温・天気・気圧等の入力から未来の気温と天気を予測するタスクを考える．既存の IMV-LSTM の可視化手法では，気温と天気の二つを予測するための入力の重要度が可視化される．そのため，気温の予測に対して重要なのか，天気を予測するために重要なのか区別できない．しかし，本研究の提案する IMV-LSTM+の可視化手法では，気温に対する重要度と天気に対する重要度を別々に可視化できる．従って，IMV-LSTM+の可視化手法は LSTM を用いた多変数データを出力する様々なタスクに対しても有効である．

5. 実験結果

本研究の有効性を確かめるため，図 6 のようなデータセットを作成した．データセットは，学習用の 200 動画，テスト用の 50 動画の計 250 動画からなる．各動画は 25 フ

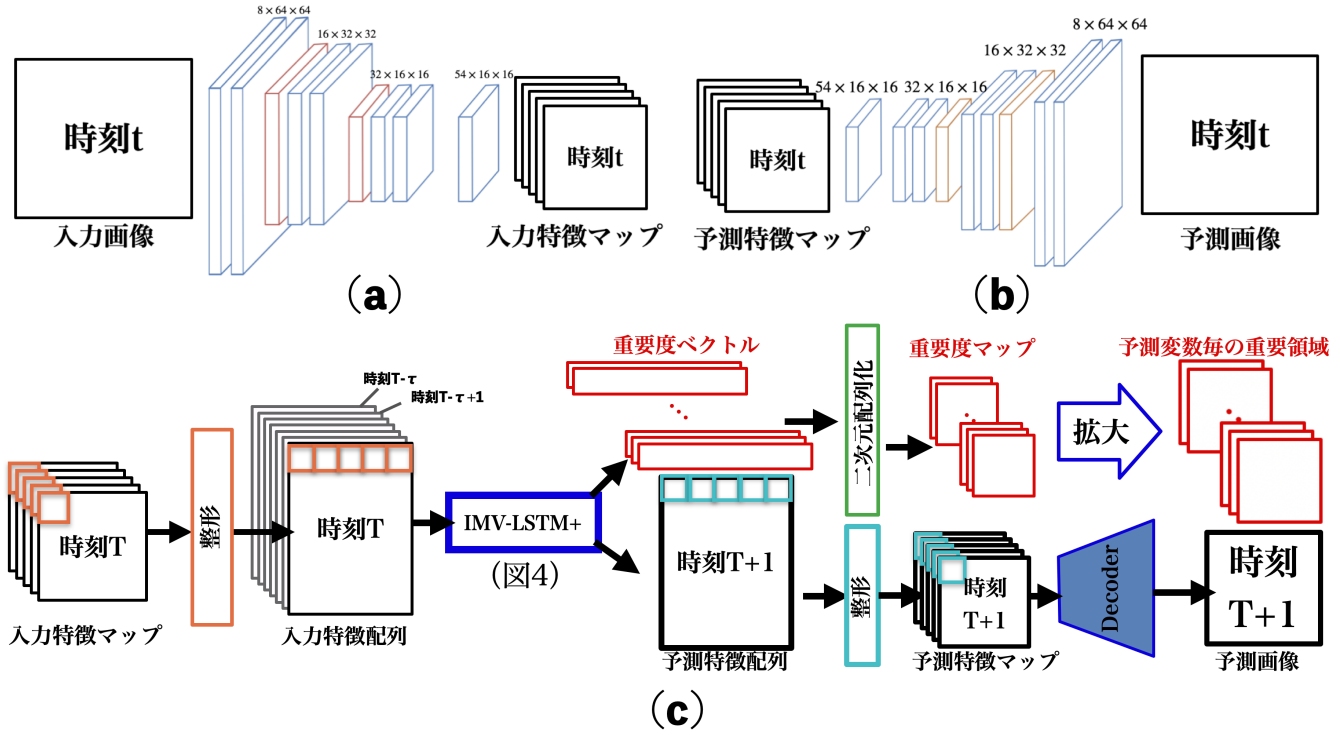

図 5 使用する動画予測モデル構造．(a) エンコーダ．(b) デコーダ．(c) 入力特徴マップを IMV-LSTM+に入力するため，再配列する．入力特徴配列から IMV-LSTM+は，予測した特徴配列を出力し，重要度を格納した重要度ベクトルを予測変数分得る．それぞれ整形し，予測特徴マップはデコーダによって画像へ復元，重要度マップは画像と同じ大きさに拡大することで画像と対応した重要領域を可視化する．

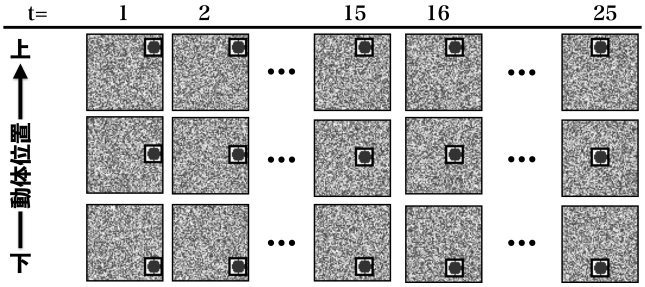

図 6 作成したデータセットの例．学習用の 200 動画，テスト用の 50 動画の計 250 動画からなる．各動画は 25 フレームの画像 (64 画素 ×64 画素) から構成され，ランダム背景の右端から 16 画素 ×16 画素の内部に黒円を含む矩形が画像が切り替わる毎に一画素分移動する．また矩形の初期高さはランダムで変化する．

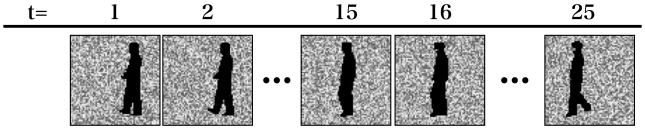

図 7 作成したデータセットの例．学習用の 160 動画，テスト用の 40 動画の計 200 動画からなる．各動画は 25 フレームの画像 (64 画素 ×64 画素) から構成され，ランダム背景の右端から黒色の人が歩くフレームで構成される．

レームの画像 (64 画素 ×64 画素) から構成され，画像右端から 16 画素 ×16 画素の内部に黒円を含む矩形が画像が切り替わる毎に一画素分移動する．また矩形の初期高さはランダムで変化する．人のような複雑な造形ではなく，また 60〜255 のランダムな画素値から構成される背景にすることで，よりモデルが動体に着目することを期待した．また，人のような複雑な造形での有効性を確かめるため，図 7 のようなデータセットも作成した．学習用の 160 動画，テスト用の 40 動画の計 200 動画からなり，各動画は 25 フレームの画像 (64 画素 ×64 画素) から構成され，ランダム背景の右端から黒色の人が歩くフレームで構成される．動体予測モデルのためのモデル構造を図 5 に示す．

図 5 (a) より入力画像 ($\mathbb{R}^{1\times64\times64}$) はエンコーダにより処理され，入力特徴マップ ($\mathbb{R}^{54\times16\times16}$) が得られる．図 5 (c) より入力特徴マップは，IMV-LSTM+に入力するため，領域毎の情報が並ぶよう $\mathbb{R}^{54\times256}$ に整形され，この整形された入力特徴マップを入力特徴配列と定義する．この際，IMV-LSTM +に二次元配列を入力するため，写像行列を $U \in \mathbb{R}^{N\times b\times d}$ に変更した．また b は入力特徴マップのチャネル数を表し，本実験では b=54 である．入力特徴配列を受け取った IMV-LSTM+は予測特徴配列 ($\mathbb{R}^{54\times256}$) と，各変数を求めるための重要度が格納された重要度ベクトル ($\mathbb{R}^{256}$) を出力する．この重要度ベクトルは，予測特徴マップサイズに対応する 256 個分存在し，入力特徴マップサイズに対応する 256 次元ベクトルからなる．ある重要

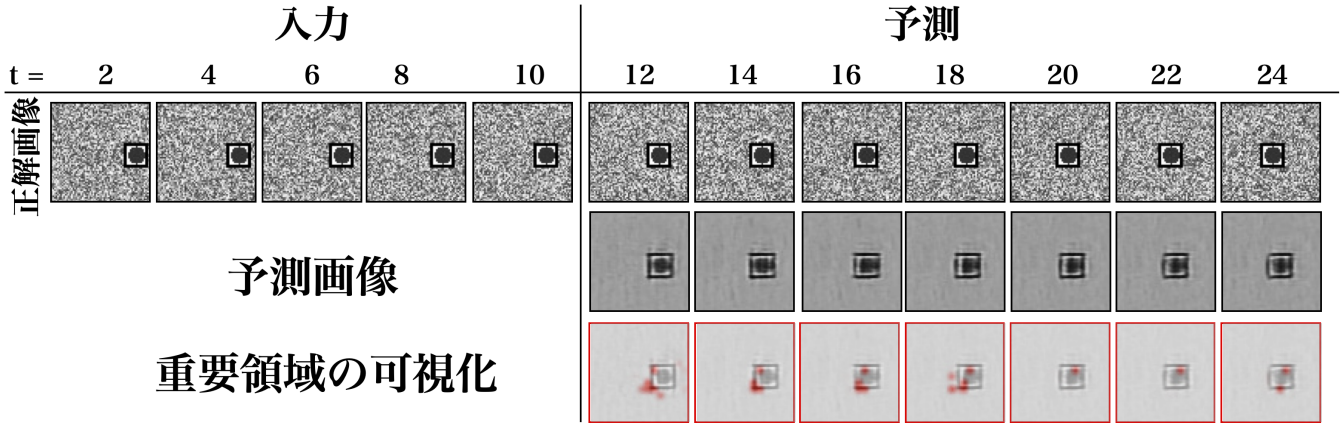

図 8 矩形が移動するデータセットを用いた実験結果．重要領域の可視化とは，動体部分を生成するための重要領域を表し，赤い領域ほどモデルが重要視していることを表す．前フレーム画像を重畳表示している．

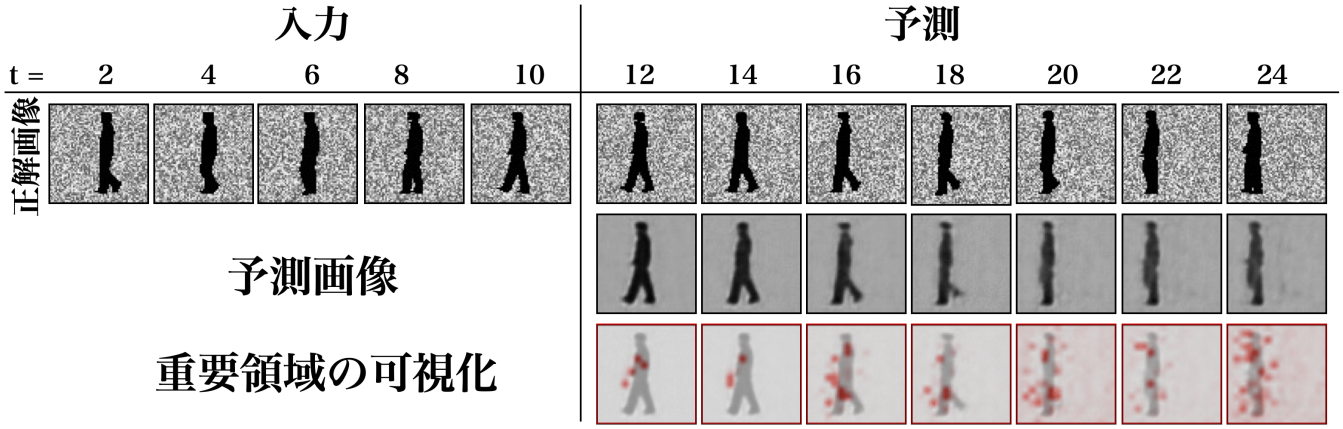

図 9 人が移動するデータセットを用いた実験結果．重要領域の可視化とは，動体部分を生成するための重要領域を表し，赤い領域ほどモデルが重要視していることを表す．前フレーム画像を重畳表示している．

度ベクトルは，ある変数を予測するための 256 個ある入力変数の各重要度が格納されたものである．そして，重要度ベクトルと予測特徴配列を整形し，重要度マップと予測特徴マップを得る．予測特徴マップはデコーダ（図 5（b））を用いて画像に復元され，重要度マップは画像の大きさへ拡大することで画像に対応した重要領域を可視化する．

学習時では，学習データから得た連続する 10 フレームの画像を入力とし，次時刻の画像の予測を行う処理を 11 フレーム目から 25 フレーム目を予測するまで繰り返す．予測画像と正解画像のピクセル間の平均二乗誤差が小さくなるように学習した．学習終了後は，テストデータを用いて，定性的評価を行った．この際，テストデータの最初の 10 フレームのみを入力し，11 フレームの予測以降は，予測画像を新しい入力として 25 フレーム目まで予測した．

動体予測のための重要領域の可視化画像の生成方法について述べる．IMV-LSTM+では，変数毎の予測の重要領域を可視化する．その変数が動体を表している場合，その変数を予測するための重要領域は動体を予測するための重要領域といえる．しかし厳密には，IMV-LSTM+から得られる重要領域は予測特徴マップの各領域を予測するための入力特徴マップの重要領域を表している．まず特徴マップと画像の対応を考える．特徴マップはエンコーダにより画像の $\frac{1}{4}$ スケールに次元削減されている．従って，画像中の（x，y）の位置にある画素は特徴マップ上では（$\lfloor \frac{x}{4} \rfloor$，$\lfloor \frac{y}{4} \rfloor$）に写像されている．画像中の（a，b）の画素が動体を構成していた場合，特徴マップの（$\lfloor \frac{a}{4} \rfloor$，$\lfloor \frac{b}{4} \rfloor$）を生成するための重要度マップから，その動体を構成する画素を生成するための重要領域を得る．全ての動体を構成する画素に対して，対応する重要度マップを求め，それらの重要度マップを足し合わせることで，動体部分を生成するための重要度マップを得る．この際，式 (11) に示す温度付き softmax 関数を用いて可視化した（T=0.3）．この温度付き softmax 関数を使用することで予測に重要な領域がより強調されることを期待した．得た動体予測のための重要度マップを画像と同じ大きさに拡大し，重要領域の可視化画像を生成した．加えて，予測画像は正解画像とほぼ合致することを期

待し，正解画像の動体位置を予測するための重要領域を動体予測のための重要領域として可視化する．

$$\sigma_T(x_i) = \frac{e^{\frac{x_i}{T}}}{\sum_{k=1}^{N} e^{\frac{x_k}{T}}} \tag{11}$$

図 8，図 9 に IMV-LSTM+を作成したデータセットを用いて学習し，テストを行った例を示す．背景予測では，背景はランダムな画素値から構成されるため，予測画像はそれらをぼかしたような灰色の平均的な背景を生成した．図 8 では，動体予測は動体の初期高さに合わせて動体が左に移動する画像が生成され，定性的に見て多少ズレはあるが予測できていると考えられる．ズレが発生する原因として，学習時では正解画像を用いて予測しているが，推論時では予測画像を用いて予測するためと考えられる．これより，提案手法である IMV-LSTM+でも十分動画予測が可能であり，図 9 から人のような複雑な造形をもつ動体でも足等細部を含め予測可能であることが示された．次に重要領域の可視化画像について述べる．動体予測のの重要領域では，赤い領域ほど重要視していることを表しており，前フレームの画像を重畳表示している．二つの実験結果共に，重要領域は動体付近にあり，動体の移動と共に移動している．従って，動画予測モデルは動体の予測のために，我々の直感通り過去の動体付近を重要視して動体予測していると考えられる．しかし，人のような複雑な造形を予測した際の重要領域は図 9 のように動体周辺を重要視していることは解釈できるが，細部にわたって具体的にどこをみて予測しているか解釈することは難しく，今後の課題である．

実験より，本研究で提案した IMV-LSTM+は動画予測が可能であることが確認できた．また動画予測モデルは我々の直感通り，動体近傍を重要視して未来の動体を予測していることが示された．

6. まとめと今後の課題

近年では，モデルの判断根拠を明確にする研究に多くの関心が持たれているが，画像生成タスクや予測タスクの判断根拠の可視化手法は，CNN モデルの可視化手法と比較して未開拓である．本研究ではその未開拓な分野の先駆け，既存の LSTM の可視化手法である IMV-LSTM を応用することで，動画予測モデルの判断根拠の可視化を研究した．実験により，本研究で提案した IMV-LSTM+は動画予測が可能であり，またモデルは我々の直感通り，動体近傍を重要視していることが示された．今後はさらに人から見て解釈しやすい可視化画像の生成や実画像での動画予測の判断根拠の可視化をテーマに研究を行う予定である．

しかしながら，本手法はエンコーダから得られる特徴マップの大きさに依存し，小さな動体に対応できない．動体が小さいもしくは細長い場合には，エンコーダから得られる特徴マップは動体を含む領域に対応する変数であっても，その情報は動体情報より背景情報を多く含むことが考えられる．判断根拠を可視化した場合，その特徴変数は動体情報よりも背景情報を多く含んでおり，対応している領域の背景を予測するための重要領域が可視化されることが想定される．従って，エンコーダから得られる特徴マップの大きさが大きいほど適切な重要領域が可視化されることが期待される．しかし一般的な動画予測モデルでは，エンコーダを用いて画像をより低次元化して扱うため，本手法は多くの動画予測モデルの手法から逸脱する．ここで近年の CNN の可視化研究ではアテンションを用いることで，CNN の性能を向上させる手法が提案されていることを受け，本研究でもアテンションを用いて動画予測モデルの性能向上，正確な可視化手法を提案する．

参考文献

[1] Hiroshi Fukui, Tsubasa Hirakawa, Takayoshi Yamashita, and Hironobu Fujiyoshi. Attention branch network: Learning of attention mechanism for visual explanation. In *CVPR*, pages 10705–10714, 2019.

[2] Bolei Zhou, Aditya Khosla, Àgata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *CVPR*, pages 2921–2929, 2016.

[3] Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *ICCV*, pages 618–626, 2017.

[4] Aditya Chattopadhyay, Anirban Sarkar, Prantik Howlader, and Vineeth N. Balasubramanian. Grad-cam++: Generalized gradient-based visual explanations for deep convolutional networks. *Arxiv*, abs/1710.11063, 2017.

[5] Tian Guo, Tao Lin, and Nino Antulov-Fantulin. Exploring interpretable LSTM neural networks over multi-variable data. In *ICML*, volume 97 of *Proceedings of Machine Learning Research*, pages 2494–2504, 2019.

[6] W. James Murdoch, Peter J. Liu, and Bin Yu. Beyond word importance: Contextual decomposition to extract interactions from lstms. In *ICLR*, 2018.

[7] W. James Murdoch and Arthur Szlam. Automatic rule extraction from long short term memory networks. In *ICLR*, 2017.

[8] Leila Arras, Grégoire Montavon, Klaus-Robert Müller, and Wojciech Samek. Explaining recurrent neural network predictions in sentiment analysis. In *WASSA@EMNLP*, pages 159–168, 2017.

[9] Yao Qin, Dongjin Song, Haifeng Chen, Wei Cheng, Guofei Jiang, and Garrison W. Cottrell. A dual-stage attention-based recurrent neural network for time series prediction. In *IJCAI*, pages 2627–2633, 2017.

[10] Edward Choi, Mohammad Taha Bahadori, Jimeng Sun, Joshua Kulas, Andy Schuetz, and Walter F. Stewart. RETAIN: an interpretable predictive model for healthcare using reverse time attention mechanism. In *NeurIPS*, pages 3504–3512, 2016.

[11] Yanbo Xu, Siddharth Biswal, Shriprasad R. Deshpande, Kevin O. Maher, and Jimeng Sun. RAIM: recurrent attentive and intensive model of multimodal patient monitoring data. In *KDD*, pages 2565–2573, 2018.