第3章

発言データの要約手法

3.1 序言

Shi ら [?] の提案した Convolutional LSTM(以降，ConvLSTM) 層を用いた時空間データの予測モデルは，単一のモダリティを入力として予測に用いている．一方，降水量の予測においては，雲の厚さや風向きなど，複数の要素 (モダリティ) が関わり，複数のモダリティを予測に用いることが望まれる．そこで本章では，Shi ら [?] のモデルを拡張した，マルチモーダル学習による時空間データの予測手法について述べる．提案手法は**複**数モダリティを多チャンネル画像として入力する手法と**正**準相関分析を取り入れた手法の 2 つである．まず，複数モダリティを多チャンネル画像として入力する提案手法について**??**節で述べる．次に，正準相関分析を取り入れた手法について**??**節で述べる．最後に，**??**節で本章のまとめを示す．

3.2　多チャンネル画像としてのマルチモーダル学習

3.2.1　複数モダリティの結合による多チャンネル画像の生成

時空間データは図??に示すように，時系列性と空間性を持つデータである．各

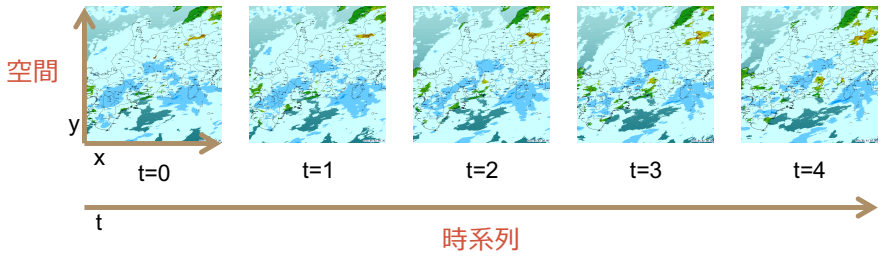

図 3.1: 時空間データの例: 各時系列データは空間性を持つ

タイムステップにおいて，空間性を持つ 1 つのデータは画像として扱うことができる．

複数種類の画像をニューラルネットで用いる手法として，??節で述べた，Zhang ら [?] の提案する手法を用いることができる．本論文で提案するモデルでは，Zhang らの手法を採用し，複数モダリティの時空間マルチモーダル学習を行う．すなわち，図?? に示すように，空間性を持つ複数モダリティを複数チャンネルから成る 1 つの画像として扱う．複数モダリティを複数チャンネルから成る 1 つの画像として扱うことにより，画像中の同じ位置の各チャンネルに各モダリティの値が入ることとなる．

複数のチャンネルを持った各画像は??節で述べた通り，3 次元のテンソルに置き換えることができる．従って，複数モダリティを多チャンネルから成る画像として扱う方法では，図??に示す 3 次元テンソルの各格子に，モダリティの数だけ観測値が入っているものと考えることができる．

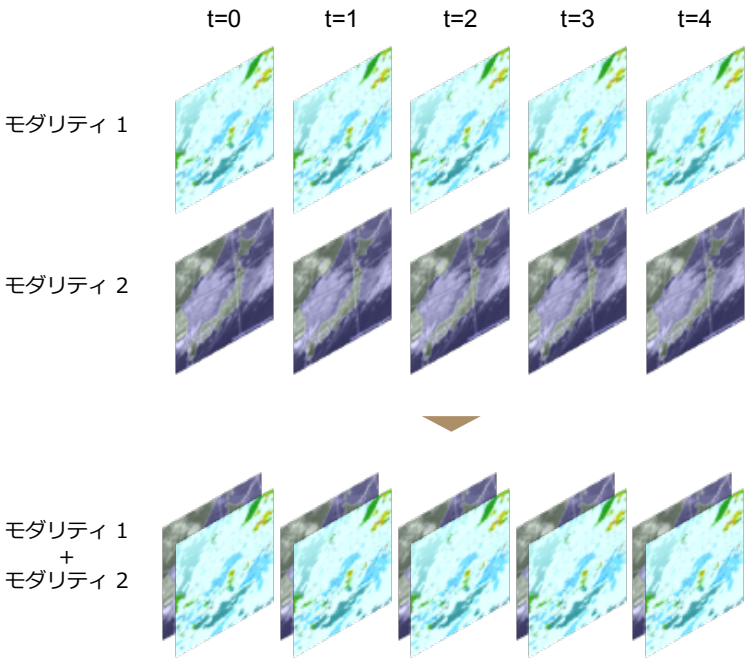

図 3.2: 複数モダリティの結合

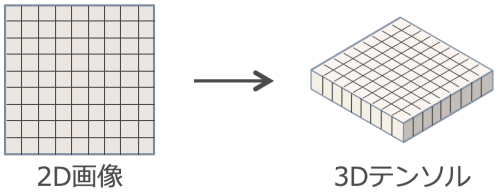

図 3.3: 複数チャンネルを持った画像としての 3 次元テンソル

パッチの作成

提案手法は Shi ら [?] のモデルで用いられていた ConvLSTM 層を用いる．従って，畳み込み演算の操作が伴う．畳み込み演算に関して，??節で述べた通り，計算コストを下げるために，入力画像を複数の小さな画像 (パッチ) へ分割する操作を行うことが多い．提案手法では，パッチ作成の方法について??節で述べた Shi ら [?] の用いた手法を採用する．従って，複数モダリティからなる多チャンネル画像について，その画像からのパッチの作成は図?? に示すように行う．すなわち，入

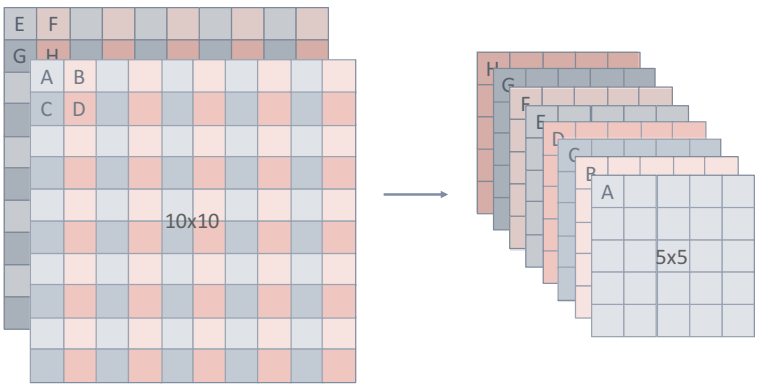

図 3.4: パッチの生成

力画像の空間的に隣り合うピクセルがパッチの同じピクセルの複数の特徴マップに取り入れられるように行う．以上の手法により各チャンネルのパッチを生成し，得られるパッチを特徴マップについて結合することで，最終的な画像パッチを得る．

3.2.2 Convolutional LSTM でのマルチモーダル学習

ConvLSTM を用いてマルチモーダル学習を行うために，??節で述べた複数モダリティを多チャンネル画像として扱う方法を用いる．複数モダリティから成る多

チャンネル画像を ConvLSTM への入力とし，マルチモーダル学習を行う．本節では，マルチモーダル学習を行う上で ConvLSTM 層に行った拡張について述べる．

局所正規化層

提案するモデルでは，ConvLSTM への入力として，複数モダリティを結合した多チャンネルの画像を用いる．多チャンネル画像での入力では，図??で示したように，入力画像の同じ位置における特徴マップ (チャンネル) に，各モダリティにおける観測値が入っている．??節で述べた通り，マルチモーダル学習ではモダリティ間の相関を見つけることが重要であり，ネットワークへの各モダリティの入力は正規化された値を用いることが望まれる．空間性を持つ画像として入力する場合，モダリティの正規化は局所正規化層の行う操作により実現できる．従って，提案するモデルでは，??節で説明した局所正規化の操作を ConvLSTM 層に組み込む．なお，局所正規化の操作では式 (??) から式 (??) を用いる．

Convolutional LSTM 層の処理

??節で説明した ConvLSTM では，LSTM の入力ユニットからの入力と隠れユニットからの入力の両方に畳み込み演算が用いられている．局所正規化の操作は畳み込みニューラルネットにおいて畳み込み層の直後に入れることが一般的 [?] である．従って ConvLSTM 層においても，畳み込み演算の結果を正規化する操作を行う (図??の①,②)．また，活性化関数を通した後に正規化することを踏まえ，ConvLSTM 層の出力について局所正規化の操作を行うモデルも検討する (図??の③)．検討する 3 か所のそれぞれに局所正規化の操作を取り入れた各モデルについ

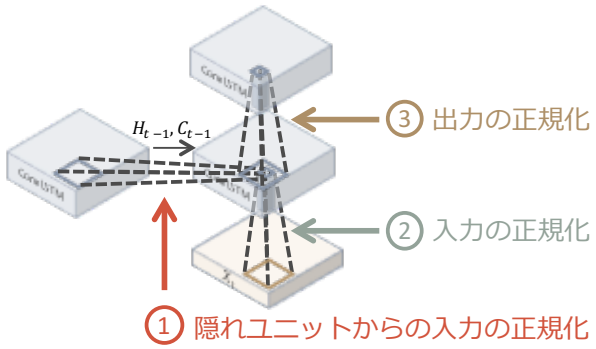

図 3.5: ConvLSTM 層への局所正規化操作の導入

て説明を行う．ただし，ConvLSTM の式中で用いる関数 norm は，??節で説明した，式 (??) から式 (??) を用いて局所正規化を行う操作を示す．

① 隠れユニットからの入力の正規化

隠れユニットからの入力 (図??の①) について局所正規化の操作を行うモデルでは，ConvLSTM の順方向計算に式 (??) を用いる．

$$
\begin{aligned}
i_t &= \sigma(W_{xi} * \mathcal{X}_t + \mathrm{norm}(W_{hi} * \mathcal{H}_{t-1}) + W_{ci} \circ \mathcal{C}_{t-1} + \boldsymbol{b}_i) \\
f_t &= \sigma(W_{xf} * \mathcal{X}_t + \mathrm{norm}(W_{hf} * \mathcal{H}_{t-1}) + W_{cf} \circ \mathcal{C}_{t-1} + \boldsymbol{b}_f) \\
\mathcal{C}_t &= f_t \circ \mathcal{C}_{t-1} + i_t \circ \tanh(W_{xc} * \mathcal{X}_t + \mathrm{norm}(W_{hc} * \mathcal{H}_{t-1}) + \boldsymbol{b}_c) \\
o_t &= \sigma(W_{xo} * \mathcal{X}_t + \mathrm{norm}(W_{ho} * \mathcal{H}_{t-1}) + W_{co} \circ \mathcal{C}_{t-1} + \boldsymbol{b}_o) \\
\mathcal{H}_t &= o_t \circ \tanh(\mathcal{C}_t)
\end{aligned}
\tag{3.1}
$$

② 入力の正規化

入力層からの入力 (図??の②) について局所正規化の操作を行うモデルでは，ConvLSTM の順方向計算に式 (??) を用いる．

$$
\begin{aligned}
i_t &= \sigma(\mathrm{norm}(W_{xi} * \mathcal{X}_t) + W_{hi} * \mathcal{H}_{t-1} + W_{ci} \circ \mathcal{C}_{t-1} + \boldsymbol{b}_i) \\
f_t &= \sigma(\mathrm{norm}(W_{xf} * \mathcal{X}_t) + W_{hf} * \mathcal{H}_{t-1} + W_{cf} \circ \mathcal{C}_{t-1} + \boldsymbol{b}_f) \\
\mathcal{C}_t &= f_t \circ \mathcal{C}_{t-1} + i_t \circ \tanh(\mathrm{norm}(W_{xc} * \mathcal{X}_t) + W_{hc} * \mathcal{H}_{t-1} + \boldsymbol{b}_c) \\
o_t &= \sigma(\mathrm{norm}(W_{xo} * \mathcal{X}_t) + W_{ho} * \mathcal{H}_{t-1} + W_{co} \circ \mathcal{C}_{t-1} + \boldsymbol{b}_o) \\
\mathcal{H}_t &= o_t \circ \tanh(\mathcal{C}_t)
\end{aligned}
\tag{3.2}
$$

③ 出力の正規化

ConvLSTM の出力 (図??の③) について局所正規化の操作を行うモデルでは，順方向計算に式 (??) を用いる．

$$
\begin{aligned}
i_t &= \sigma(W_{xi} * \mathcal{X}_t + W_{hi} * \mathcal{H}_{t-1} + W_{ci} \circ \mathcal{C}_{t-1} + \boldsymbol{b}_i) \\
f_t &= \sigma(W_{xf} * \mathcal{X}_t + W_{hf} * \mathcal{H}_{t-1} + W_{cf} \circ \mathcal{C}_{t-1} + \boldsymbol{b}_f) \\
\mathcal{C}_t &= f_t \circ \mathcal{C}_{t-1} + i_t \circ \tanh(W_{xc} * \mathcal{X}_t + W_{hc} * \mathcal{H}_{t-1} + \boldsymbol{b}_c) \\
o_t &= \sigma(W_{xo} * \mathcal{X}_t + W_{ho} * \mathcal{H}_{t-1} + W_{co} \circ \mathcal{C}_{t-1} + \boldsymbol{b}_o) \\
\mathcal{H}_t &= \mathrm{norm}(o_t \circ \tanh(\mathcal{C}_t))
\end{aligned}
\tag{3.3}
$$

なお，式 (??)，(??)，および (??) 中の $*$ は畳み込み演算を指し，$\circ$ はアダマール積を指す．

3.2.3 Encoding-Forecasting ネットワーク

提案するモデルは，Shi ら [?] の Encoding-Forecasting ネットワークに，??節で説明した，拡張した ConvLSTM を用いてマルチモーダル学習を行うものである．すなわち，図??に示すように，Encoding ネットワークと Forecasting ネットワーク

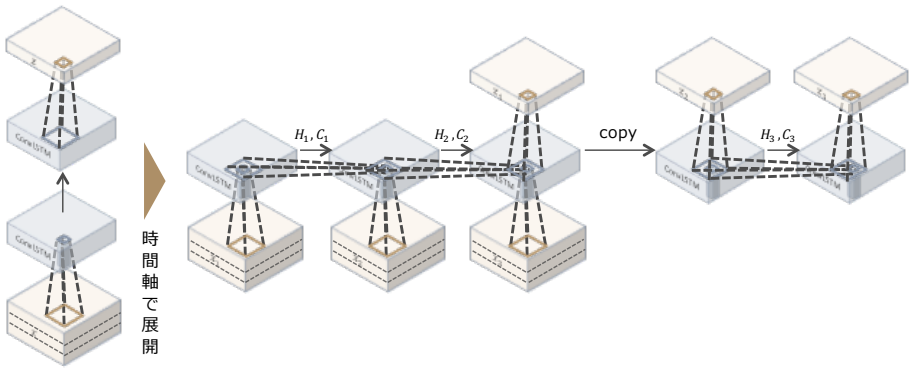

図 3.6: 拡張した ConvLSTM 層による Encoding-Forecasting ネットワーク

を時間軸で結合させ，Forecasting ネットワークからの出力をネットワーク全体の出力とする．各タイムステップにおいて M 個のモダリティの入力 $\mathcal{X}^1$,..., $\mathcal{X}^M$ は結合され，3 次元テンソルの各特徴マップとして Encoding ネットワークに入力される．このとき，タイムステップ t におけるネットワークへの入力テンソルを $\mathcal{X}_t = [\mathcal{X}_t^1$, ..., $\mathcal{X}_t^M]$ と表記すると，J 個の現在を含む過去の連続的な時空間データから，引き続くであろう K 個の将来の連続的な時空間データを予測するモデルは式 (??) に示

すように表される.

$$\begin{aligned}\hat{\mathcal{X}}_{T+1},...,\hat{\mathcal{X}}_{T+K} &= p(\mathcal{X}_{T+1},..., \mathcal{X}_{T+K} \mid \mathcal{X}^1_{T-J+1}, \mathcal{X}^1_{T-J+2},..., \mathcal{X}^1_T, \mathcal{X}^2_{T-J+1},..., \mathcal{X}^M_T) \\ &\approx p(\mathcal{X}_{T+1},..., \mathcal{X}_{T+K} \mid f_{encoding}(\mathcal{X}_{T-J+1}, \mathcal{X}_{T-J+2},..., \mathcal{X}_T) \\ &\approx g_{forecasting}(f_{encoding}(\mathcal{X}_{T-J+1}, \mathcal{X}_{T-J+2},..., \mathcal{X}_T))\end{aligned} \tag{3.4}$$

ただし，$f_{encoding}(\mathcal{X})$，$g_{forecasting}(\mathcal{X})$ はそれぞれ Encoding ネットワーク，Forecasting ネットワークへの入力 $\mathcal{X}$ に対する出力を求める関数を示す.

また，Forecasting ネットワークにおいて，ConvLSTM の出力の特徴マップ数とネットワークの出力 (目標) の特徴マップ数が異なるが，??節で示した，大きさ 1×1 のフィルタによる畳み込み演算の操作を ConvLSTM の出力に対して行うことで，ネットワークの出力 (目標) の特徴マップ数と一致させる.

なお，提案する時空間データのマルチモーダル学習モデルは，式 (??) に示した通り，Encoding ネットワークへの入力は複数モダリティであるが，Forecasting ネットワークからの出力は単一モダリティであることを想定している.

3.3 正準相関分析を取り入れたマルチモーダル学習

本節では，提案手法である正準相関分析 (CCA) を取り入れたマルチモーダル学習について述べる．まず，Sequence to Sequence モデルへの CCA の取り入れ方について??節で述べる．次に，CCA を取り入れた，提案する 2 種類の Sequence to Sequence モデルについて，それぞれ??節と??節で述べる．

3.3.1 Sequence to Sequence モデルへの CCA の適用

Andrew ら [?] によって提案された Deep Canonical Correlation Analysis (DCCA) モデルを参考に，Sequence to Sequence モデルに CCA を取り入れる方法を考える．DCCA モデルでは，エンコーダネットワークによって得られた各モダリティに対する特徴ベクトルについて，特徴ベクトル間の相関を最大化するよう学習を行う．従って，Sequence to Sequence モデルにおいても，同様にエンコーダネットワークから得られた特徴ベクトル間の相関を最大化する学習を行う．

LSTM を用いる Sequence to Sequence モデルでは，入力系列 $\boldsymbol{x}^1, ..., \boldsymbol{x}^t$ がエンコーダに入力されたのち，時刻 t における LSTM の状態 (隠れユニットの状態 $\boldsymbol{h}^t$ およびセルの状態 $\boldsymbol{c}^t$) をデコーダの初期状態として用い，デコーダからネットワークの出力を求める．このとき，入力の最終時刻 t における LSTM の隠れユニットの状態 $\boldsymbol{h}^t$ は，DCCA モデルにおけるエンコーダネットワークから得られた特徴ベクトルと同等に捉えることができる．提案手法では，入力の最終時刻 t における LSTM の隠れユニットの状態 $\boldsymbol{h}^t$ をエンコーダネットワークから得られる特徴ベクトルとして考え，CCA を適用する．すなわち，2 つのモダリティの入力系列 $\boldsymbol{X} = [\boldsymbol{x}^1, ..., \boldsymbol{x}^t]$ および $\boldsymbol{Y} = [\boldsymbol{y}^1, ..., \boldsymbol{y}^t]$ があったとき，異なるエンコーダ f および

g を通して得られる各モダリティに対する特徴ベクトル $f(\boldsymbol{X}) = \boldsymbol{h}_f^t$ と $g(\boldsymbol{Y}) = \boldsymbol{h}_g^t$ の相関を最大化するよう学習を行う．

一方，ConvLSTM を用いた Sequence to Sequence モデルでは，特徴量として 1 次元の特徴ベクトル $\boldsymbol{h}^t$ ではなく 3 次元のテンソルから成る特徴マップ $\mathcal{H}^t$ を用いる．従って，2 つのモダリティの入力系列 $\boldsymbol{\mathcal{X}} = [\mathcal{X}^1, ..., \mathcal{X}^t]$ および $\boldsymbol{\mathcal{Y}} = [\mathcal{Y}^1, ..., \mathcal{Y}^t]$ があったとき，異なるエンコーダ f および g を通して得られる各モダリティに対する特徴マップ $f(\boldsymbol{\mathcal{X}}) = \mathcal{H}_f^t$ と $g(\boldsymbol{\mathcal{Y}}) = \mathcal{H}_g^t$ の相関を最大化するよう学習を行う．

計算量の削減

入力系列の各フレームのサイズが $1 \times H \times W$(それぞれ，チャンネル数，高さ，および幅) のとき，K 個のフィルタから成る ConvLSTM により，サイズ $K \times H \times W$ の特徴マップが得られる．3 次元テンソルである特徴マップ $f(\boldsymbol{\mathcal{X}})$ および $g(\boldsymbol{\mathcal{Y}})$ に対し DCCA を適用する場合，そのままでは式 (??) に示すような相関項を持つ目的関数の最適化問題を解くこととなる．

$$\begin{aligned} \max_{W_f, W_g, U, V} \quad & \frac{1}{N} \mathrm{tr}\left(U^{\mathrm{T}} f(\boldsymbol{\mathcal{X}}) g(\boldsymbol{\mathcal{Y}})^{\mathrm{T}} V\right) \\ \text{where} \quad & f(\boldsymbol{\mathcal{X}}) \in \mathbb{R}^{K \times H \times W} \\ & g(\boldsymbol{\mathcal{Y}}) \in \mathbb{R}^{K \times H \times W} \end{aligned} \tag{3.5}$$

ただし，式 (??) における W_f および W_g はモデルのパラメータであり U および V は DCCA における射影行列を示す．このとき，そのままでは計算できないため，安直に特徴マップ $f(\boldsymbol{\mathcal{X}})$ および $g(\boldsymbol{\mathcal{Y}})$ を 3 次元テンソルからベクトルへと平らにし，特徴ベクトル $f(\boldsymbol{\mathcal{X}}) \in \mathbb{R}^{KHW}$ および $g(\boldsymbol{\mathcal{Y}}) \in \mathbb{R}^{KHW}$ とすることで，計算が可能となる．しかし，この安直な方法では射影行列が $U \in \mathbb{R}^{KHW}$ および $V \in \mathbb{R}^{KHW}$ と

なり，相関項 $\mathrm{tr}\left(U^{\mathrm{T}} f(\boldsymbol{\mathcal{X}}) g(\boldsymbol{\mathcal{Y}})^{\mathrm{T}} V\right)$ の計算に大きな空間計算量を要する．

そこで，特徴マップ $f(\boldsymbol{\mathcal{X}})$ および $g(\boldsymbol{\mathcal{Y}})$ を平らにするのではなく，それぞれ k 番目の特徴マップ $f(\boldsymbol{\mathcal{X}})_{(k)} \in \mathbb{R}^{H \times W}$ および $g(\boldsymbol{\mathcal{Y}})_{(k)} \in \mathbb{R}^{H \times W} (k = 1, ..., K)$ を平らにしたものを 1 つの特徴ベクトルとして捉え，相関項を式 (??) のように近似する．

$$\mathrm{tr}(\boldsymbol{U}^{\mathrm{T}} \boldsymbol{f}(\boldsymbol{\mathcal{X}}) \boldsymbol{g}(\boldsymbol{\mathcal{Y}})^{\mathrm{T}} \boldsymbol{V}) \approx \sum_{k} \mathrm{tr}(\boldsymbol{U}^{\mathrm{T}} \boldsymbol{f}(\boldsymbol{\mathcal{X}})_{(k)} \boldsymbol{g}(\boldsymbol{\mathcal{Y}})_{(k)}^{\mathrm{T}} \boldsymbol{V}) \tag{3.6}$$

すなわち，K 個の特徴マップのそれぞれを，独立したサイズ $1 \times W \times H$ の特徴マップとして捉え，平らにして出来上がる特徴ベクトルすべてについて同じ射影行列を用いる近似を行う．式 (??) による近似により，射影行列が $\boldsymbol{U} \in \mathbb{R}^{WH}$ および $\boldsymbol{V} \in \mathbb{R}^{WH}$ となり，相関項の計算における空間計算量が削減できる．

3.3.2 提案モデル 1: DCCA ConvLSTM Model

DCCA を導入した Sequence to Sequence モデルとして DCCA ConvLSTM モデルを提案する．本モデルの概要を図??に示す．

DCCA ConvLSTM モデルでは，ConvLSTM を用いた Sequence to Sequence モデルにおいて，目的関数にモダリティ間の相関を表す相関項を単純に追加することで，DCCA を取り入れている．従って，モデルの学習は式 (??) の目的関数を用いて行われる．

$$\begin{aligned} \min_{\boldsymbol{W_f}, \boldsymbol{W_g}, \boldsymbol{W_h}, \boldsymbol{U}, \boldsymbol{V}} & -\frac{\lambda}{N} \mathrm{tr}\left(\boldsymbol{U}^{\mathrm{T}} \boldsymbol{f}(\boldsymbol{\mathcal{X}}) \boldsymbol{g}(\boldsymbol{\mathcal{Y}})^{\mathrm{T}} \boldsymbol{V}\right) \\ & + \mathrm{Loss}\big(\boldsymbol{h}(\boldsymbol{f}(\boldsymbol{\mathcal{X}})), \boldsymbol{\mathcal{Z}}\big), \end{aligned} \tag{3.7}$$

ただし，式 (??) において $\mathrm{Loss}\big(\boldsymbol{h}(\boldsymbol{f}(\boldsymbol{\mathcal{X}})), \boldsymbol{\mathcal{Z}}\big)$ は，ネットワークの出力系列 $\boldsymbol{h}(\boldsymbol{f}(\boldsymbol{\mathcal{X}}))$ と目標系列 $\boldsymbol{\mathcal{Z}}$ との誤差を表す．

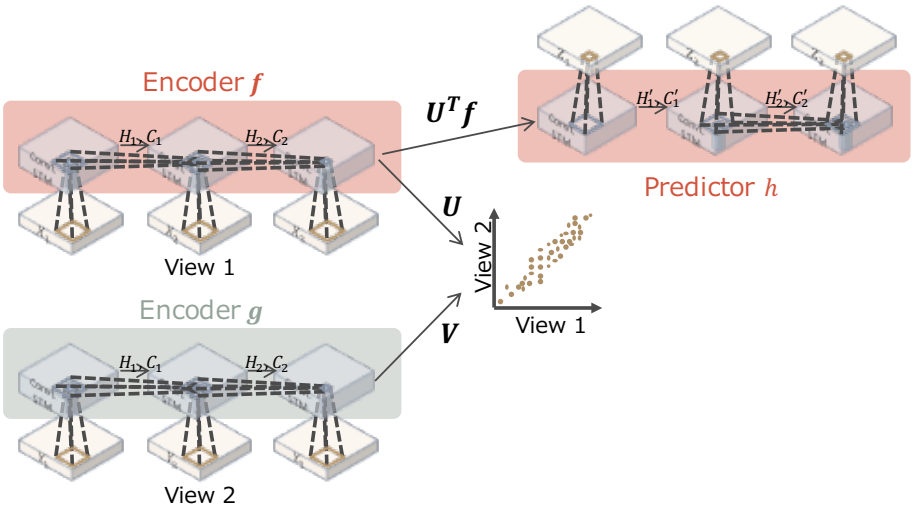

図 3.7: DCCA ConvLSTM-seq2seq model

3.3.3 提案モデル 2: DCCA-Pretrain ConvLSTM Model

DCCA の導入によりマルチモーダル学習を行うもう一つのモデルとして，DCCA-Pretrain ConvLSTM モデルを提案する．本モデルは，DCCA を用いた事前学習を取り入れることで，マルチモーダル学習を行う．

DCCA-Pretrain ConvLSTM ではまず，図??に示すような自己符号化器型のネットワークにて，モダリティ間の相関を最大化しつつ入力系列から入力系列を復元する事前学習 (Pretraining) を行う．具体的には，事前学習を式 (??) の目的関数を用いて行う．

$$\begin{aligned}\min_{\boldsymbol{W_f},\boldsymbol{W_g},\boldsymbol{W_p},\boldsymbol{W_q},\boldsymbol{U},\boldsymbol{V}} & -\frac{\lambda}{N}\mathrm{tr}\left(\boldsymbol{U}^{\mathrm{T}}\boldsymbol{f}(\boldsymbol{\mathcal{X}})\boldsymbol{g}(\boldsymbol{\mathcal{Y}})^{\mathrm{T}}\boldsymbol{V}\right) \\ & + \mathrm{Loss}\big(\boldsymbol{p}(\boldsymbol{f}(\boldsymbol{\mathcal{X}})),\boldsymbol{\mathcal{X}}\big) + \mathrm{Loss}\big(\boldsymbol{q}(\boldsymbol{g}(\boldsymbol{\mathcal{Y}})),\boldsymbol{\mathcal{Y}}\big),\end{aligned} \tag{3.8}$$

ただし，式 (??) において $\mathrm{Loss}\big(\boldsymbol{p}(\boldsymbol{f}(\boldsymbol{\mathcal{X}})),\boldsymbol{\mathcal{X}}\big)$ および $\mathrm{Loss}\big(\boldsymbol{q}(\boldsymbol{g}(\boldsymbol{\mathcal{Y}})),\boldsymbol{\mathcal{Y}}\big)$ はそれぞれ

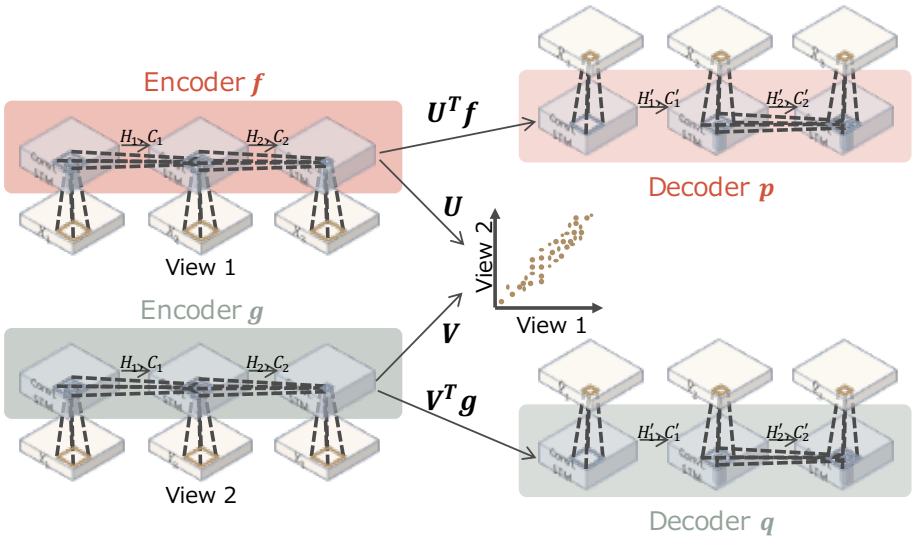

図 3.8: DCCA-Pretrain ConvLSTM-seq2seq model

復元された入力系列 $\boldsymbol{p}(\boldsymbol{f}(\boldsymbol{\mathcal{X}}))$ および $\boldsymbol{q}(\boldsymbol{g}(\boldsymbol{\mathcal{Y}}))$ と入力系列 $\boldsymbol{\mathcal{X}}$ および $\boldsymbol{\mathcal{Y}}$ の誤差を示す.

事前学習の後，図??に示すように 2 つのデコーダを取り外し，新たなデコーダを片方のモダリティについて取り付け，目標系列を出力するよう学習 (finetune) を式 (??) の目標関数を用いて行う.

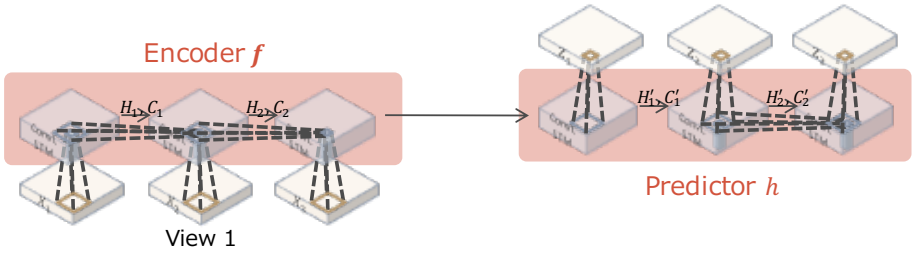

図 3.9: DCCA-Pretrain ConvLSTM-seq2seq model

$$\min_{\boldsymbol{W_f},\boldsymbol{W_g},\boldsymbol{W_h}} \mathrm{Loss}\big(\boldsymbol{h}(\boldsymbol{f}(\boldsymbol{\mathcal{X}})), \boldsymbol{\mathcal{Z}}\big) \tag{3.9}$$

ただし，式 (??) において $\mathrm{Loss}\big(\boldsymbol{h}(\boldsymbol{f}(\boldsymbol{\mathcal{X}})), \boldsymbol{\mathcal{Z}}\big)$ は，ネットワークの出力系列 $\boldsymbol{h}(\boldsymbol{f}(\boldsymbol{\mathcal{X}}))$ と目標系列 $\boldsymbol{\mathcal{Z}}$ との誤差を表す．

3.4 結言

本章では，提案する時空間データのマルチモーダル学習手法について述べた．提案手法は，Shi ら [?] の提案した ConvLSTM による Sequence to Sequence モデルを基に，多チャンネル画像として複数モダリティを入力する手法，および正準相関分析を取り入れた手法であり，それぞれの手法について説明を行った．