2024 年度卒業論文

機械学習を用いた Cygnus-X 領域における大質量星に付随する分子雲の特性の調査

Investigation of the Characteristics of Molecular Clouds Associated with Massive Stars in the Cygnus-X Region Using Machine Learning

大阪府立大学

生命環境科学域　理学類　物理科学課程

電波天文学研究室

1211305117　藤本 湧大

2025 年 2 月 28 日

要旨

大質量星は太陽の 8 倍以上の質量をもつ恒星のことであり、強烈な紫外線の放出や、超新星爆発による 重元素の放出などから銀河進化に多大な影響を及ぼすと考えられている。したがって、大質量星の形成過程を理解することは銀河進化の解明を目指す上で必要不可欠である。この大質量星が形成されるシナリオとして現在有力な説が、分子雲同士の衝突によって発生した高密度領域で形成されるというものと、大質量星からの輻射により発生する圧縮されたバブル状の構造を持つ分子雲内で新たに星が形成される、連鎖的星形成というものがある。

大質量星が発生させる輻射によるバブル構造は、スピッツァー宇宙望遠鏡などで観測された赤外線データに対し、市民科学者を動員した大規模調査や、深層学習を用いた物体検出を行うことによって多数検出されている。

本研究では、赤外線バブルが見られる領域の分子雲が実際に大質量星からの影響を受け、一般的な分子雲とは異なる空間、速度分布の特性を持っているのかどうかを Cygnus-X 領域の分子雲データを使用し、統計的に調査することを目的とした。Cygnus-X は星形成が活発に行われている領域であり、先行研究により赤外線バブル構造が多数検出されている。調査の際、膨大な三次元天文データの統計的解析を人間の手で行うことは非常に困難であるという問題があった。それを解決する方法として、本研究では機械学習の手法を用いた。機械学習の生成モデルの一つである CAE(Convolutional Auto Encoder)の画像再構築の際に生じる、潜在変数と呼ばれる入力データの特徴量が圧縮された変数を用いて、バブルに付随する分子雲と、ランダムな領域における分子雲の潜在変数の分布違いをそれぞれ Cygnus-X 全領域の分子雲の潜在変数と比較した結果、ランダムに選出した領域と比べ、バブル領域で分布が異なると判断された潜在変数は明らかに多いという結果となった。

このことより、赤外線バブルが存在する領域の分子雲は、一般的な分子雲とは異なる空間、速度分布の特性を持っていることが明らかになった。

本研究で作成した機械学習モデルを使用すれば、分子雲データから一般的な領域とは異なる特性を持った領域を特定できる可能性がある。一方、機械学習モデル内のパラメータ調整がブラックボックになっており、何に注目して特徴量を抽出しているのかが不明な点などの課題も残されている。

目次

第 1 章　イントロダクション

1.1　星間物質

星間物質は、主にガスと星間微粒子（ダスト）からなる。ガスの主成分は水素であり、ガスの化学組成は宇宙全体の元素組成に近い。ダストは炭素、ケイ素、酸素、鉄等の重元素から構成され、ガスとダストの質量比は通常 100 対 1 程度である。ガスの温度は放射などによる加熱と冷却のバランスで決まり、星間物質に働く各種の力のバランスによってガスの密度が決まる。そのため、ガスは様々な密度と温度を示す。本節では、星間物質の大まかな性質や放射機構について述べる。

1.1.1　星間物質の密度と温度

図 1.1 は星間ガスの種類を密度と温度の座標軸上にまとめたものである。星間ガスは基本的に中性水素ガス（HI ガス）で構成されている。天文学においては、元素記号 X の中性ガスに対して添え字 I を付けて XI と記し、1 階電離、2 階、3 階電離したガスに対してはそれぞれ、XII、 XIII というようにして表す慣用がある。例えば中性水素ガスは HI ガスとも呼ばれる。星間ガスは密度と温度の違いにより、コロナガス、雲間物質、HI ガス雲、HII 領域、分子雲に分類される。温度が 100 万度程度の高温・低密度ガスをコロナガスという。コロナガスは超新星残骸の中や、スーパーバブルと呼ばれる複数の超新星爆発や星風の効果で形成されたと考えられる巨大な高温低密度領域の内部に観測されるため、超新星爆発により高温化したガスであると考えられる。雲間物質と水素原始ガス雲（HI ガス雲）は圧力がほぼ等しく平衡状態にある。密度と温度の積はこれらのガス同士でほぼ等しく、圧力平衡が近似的に成り立つ。圧力平衡から外れた存在として、電離水素領域（HII 領域）と星間分子雲がある。電離水素領域は、主に若い大質量星の周りに存在する水素ガスが電離した領域である。分子雲は自己重力で束縛された最も高密度の星間ガスであり、星形成の場である。分子雲については 1.1.2 項にて詳しく述べる。

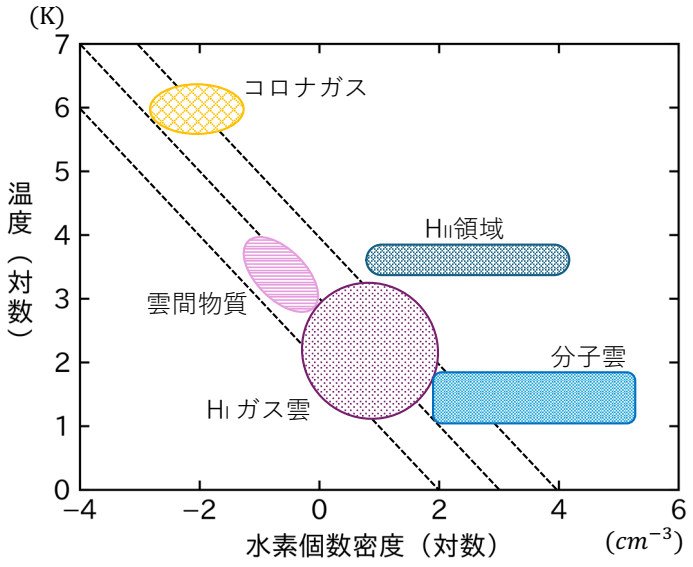

図 1.1: 絶対温度と密度で表された星間物質の様々な存在形態(credit: 福井康雄「星間物質」、シリーズ現代の天文学第 6 巻、福井・犬塚・大西・中井・舞原・水野編『星間物質と星形成』1.1 節図 1.2（日本評論社）を一部改変)

1.1.2 分子雲

星間ガスの中でも特に密度が高く、水素分子を主成分とし空間的なまとまりをもった星間ガス雲を分子雲という。分子雲の温度は数 10 K 程度と非常に冷たく、その主成分である水素分子の直接検出は非常に難しい。なぜなら、水素分子は、2 個の水素原子の距離が変化しない状態では電気双極子モーメントを持たず電気双極子放射を出さないため、電磁波を能率良く放射・吸収する状況が非常に限られているからである。水素分子が高い内部エネルギーを持ち、原子間距離が変わる振動状態と回転運動状態とが同時に変化する場合には電気双極子モーメントが生じ、対応する振動回転遷移が起こる。これに対応する輝線は波長 2 μmなど近赤外線で見られるが、温度・密度が高い領域でしか励起されないため、これが検出されるのは、強い電磁放射で照らされている場合や衝撃波が発生するような領域に限られる。これに対して、回転エネルギー準位間の遷移（回転遷移）による輝線は波長 28.2μm、17.0μm、12.3μmなど中間赤外線から近赤外線にかけて放射されるが、これも温度が 100 K 以上の領域などでないと励起されない。したがって、星間分子の多くを占める 10 K 程度の低温の水素分子の観測は困難である。しかしながら、電気双極子モーメントをもつ分子の回転遷移(ミリ波・サブミリ波帯)は 10 K 程度の低温でも引き起こされる。そこで、分子雲の研究には、水素分子に次ぐ存在量を持つ一酸化炭素分子(CO)などのミリ波・サブミリ波帯の回転遷移による分子輝線の観測がよく利用される。一酸化炭素は、化学的に安定で水素分子との存在比が分子雲の環境によらずほぼ一定（約 1/10000）とみなして良いため、CO の分子輝線を観測することによって、分子雲内でのガスの分布に加えて、温度、密度、質量等の物理量を推定することができる。3.1.1 節でも述べているが、分子雲に含まれる分子ガスの運動（視線速度）は、ドップラー効果により分子輝線の静止周波数からのずれとして観測される。分子輝線を解析すれば、銀河の回転曲線に基づいて分子雲の運動学的距離の計算や、輝線の周波数（速度）方向の広がりから分子雲内部の速度場（速度構造や線幅）の情報を得ることができる。またこれまでの銀河系内の観測から、分子雲では典型的な温度である 10 K から予想される熱的線幅よりも広がった線幅が観測されることから、分子雲は超音速乱流状態にあると考えられている。

1.1.3 CO 分子による回転遷移スペクトル

CO は回転遷移数 J が J=1-0、J=2-1、J=3-2 の様に異なる準位間を遷移する際にそのエネルギー差に相当する周波数の電波を放射する（図 1.2)。エネルギーを得る機構として、

(1) 分子雲の大半を占める水素やヘリウムの熱運動による CO への非弾性衝突
(2) 放出した光子の分子雲内での再捕獲
(3) 分子雲外部からの放射

などがある。エネルギーを失う機構としては(1)による逆励起や自然放射が考えられ、励起状態はこれらの収支バランスにより決まる。異なる遷移間の回転遷移スペクトルを観測することで、温度や質量などの物理的状態を調べることができる。本研究で用いたデータの観測輝線は ^{12}CO(J=1-0)輝線である。

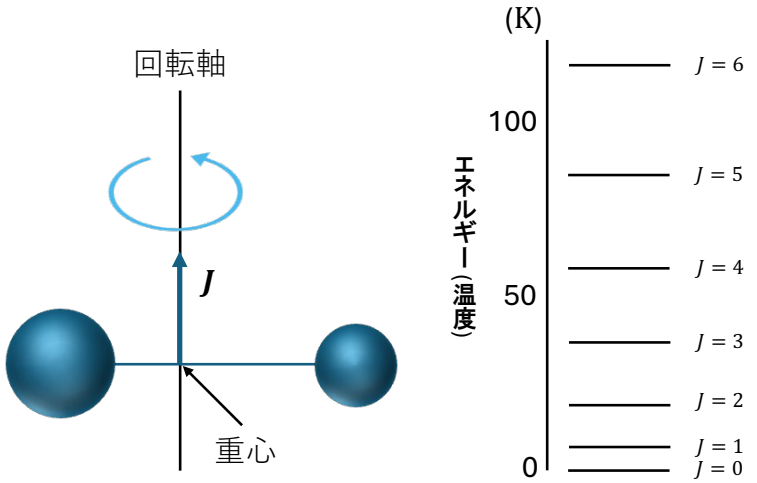

図 1.2: 2 原子分子の回転スペクトル（credit: シリーズ現代の天文学第 16 巻より一部改変）

1.2 星形成

星は宇宙空間に漂う分子雲から生まれる。通常、分子雲は自身の重力と、分子雲中の超音速乱流による圧力が釣り合って存在する。しかし、分子雲の中でも密度が非常に濃い領域ではしばしば重力が圧力を上回り、重力崩壊を起こして星へと進化していく。星は質量によって区別され、太陽の 8 倍以上の質量を持つ星を大質量星、それ以下の質量を持つ星を小質量星と呼ぶ。宇宙には、太陽の 10 倍、100 倍以上の質量を持つ大質量星が存在する。これらの星は太陽のような軽い星と比べて数は少ないが、強い輻射や超新星爆発によって、周りのガスの進化に大きく影響を及ぼす。例えば、大質量星が放出する紫外線は星の材料になる冷たいガス（分子雲）を加熱して星形成を抑制する。一方で、電離領域の膨張や超新星爆発が起これば周りの薄いガスを掃き集めて密度の濃い領域を作り、星形成を誘発する。

小質量星には標準的な形成シナリオが存在するのに対し、大質量星形成シナリオについては謎な点が多い。しかし、大質量星は超新星爆発や強力な輻射により周囲の環境に多大な影響を及ぼし、超新星爆発により重元素を生成して宇宙全体の化学進化の中心的役割を果たす。よって、大質量星形成の理解は活発に議論されている。

1.2.1 小質量星形成シナリオ

小質量星形成は分子雲内の「分子雲コア」と呼ばれる高密度ガスが自己重力で崩壊することにより開始する。材料となる分子雲コアの総質量は約 1 太陽質量（天文学では 1 太陽質量を$M_\odot$と表す）で、これはコア自身の密度 $10^4\,\mathrm{cm}^{-3}$ と温度 10K から見積もられるジーンズ質量（自己重力崩壊に必要な最小質量）とほぼ同程度である。重力崩壊の結果、コアの中心付近に星の赤ちゃん「原始星」が誕生する。誕生時の原始星は約 10^{-3} 太陽質量と非常に軽いが、周囲に残された分子雲コアのガスが降着することで、およそ 10^{-6} 太陽質量／年の割合でその質量を増やしていき、最終的に水素核燃焼による発熱により重力収縮が停止し主系列星となる。この標準シナリオは観測的にも各進化段階に対応する天体が特定されており、初期条件の理解を除いては十分に確立していると言って良い。

1.2.2 大質量星形成シナリオ

1.2.1 項で示したように小質量星の形成には標準的なシナリオが存在する一方で、大質量星の形成シナリオは主に、

(1) 絶対数が少ない
(2) 進化のタイムスケールが短く進化の途中段階に対応するサンプルが少ない
(3) 太陽近傍の巨大分子雲が少ない（大質量星は巨大分子雲の中で集団的に形成されることが観測的研究で明らかになっている）ため遠方の巨大分子雲を観測するしか解析手段がないため空間分解能が悪い

などの問題点から理解が遅れている。

小質量星の形成シナリオを大質量星形成に当てはめて考えることによって大質量星形成も理解できそうに思われるが、大質量星に小質量星形成シナリオを当てはめた場合、2 つの問題に直面する。

寿命問題

1 つ目は寿命問題である。原始星が質量降着により進化する際、初期には降着時間が十分短いため、原始星が主系列星に向けて収縮するよりも早く質量が増加していく。そして、質量が増すに連れてこれらのタイムスケールの値は近くなり、いずれ逆転する。そうすると、中心星は降着を続けながら中心部で核反応が始まる。小質量星形成の際の典型的な降着率 $10^{-5} M_{\odot} \mathrm{yr}^{-1}$ では、中心星は約 $8 M_{\odot}$程度で主系列星に到達することが知られている。しかし、大質量星の寿命は $2\text{~}3 \times 10^{6}$ 年と短く、降着率 $10^{-5} M_{\odot} \mathrm{yr}^{-1}$ 程度では、寿命を終えるまで膠着を続けていたとしても、せいぜい $30\text{~}40 M_{\odot}$程度までしか質量を増やすことができない。図 1.2 は、Wolfire & Cassinelli et al. 1987 によって計算された原始星への質量膠着率と降着により形成可能な上限質量の関係である[1]。図中の線 B は、星の寿命中に獲得可能な質量の制限を表す。$\log \mathrm{M} = 10^{-5}$ の時に、質量が $30\text{~}40 M_{\odot}$であることが確認できる。

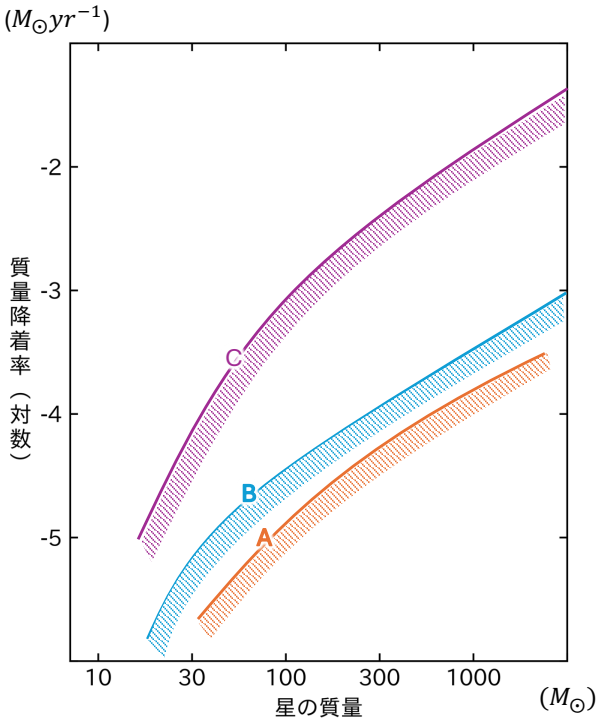

図 1.3: 原始星の質量降着率と、降着により形成可能な上限質量の関係。横軸が星の質量、縦軸は質量降着率（対数）である。(credit: Wolfire & Cassineli et al. 1987 より一部改変)
線 A：H II 領域の膨張により降着が止められることによる制限
線 B：中心星の寿命による制限
線 C：放射圧により降着が止められることによる制限

フィードバック問題

2 つ目はフィードバック（輻射）問題である。小質量星形成は分子雲の高密度部分が自身の重力で潰れることから始まる。このとき、中心部ほど早くつぶされ、最後には圧力で重力を支えた原子星ができる。原始星の質量ははじめ非常に小さく、ガスの大半は原始星を取り巻く外層に残されたままである。ところが、星の光度も星質量とともに大きくなるため、降着外層にダストを介した輻射圧がしだいに効くようになり、星質量が大きくなるに従い降着物質は輻射圧により押し返されてしまう。図 1.2 中の線 A、B、C を比較すると、星間微粒子への放射圧（線 C）が一番厳しい条件を与えることがわかる。通常の小質量星形成の際と同じ程度の降着率（$10^{-5}M_\odot \mathrm{yr}^{-1}$）のもとでは、$10\sim20M_\odot$よりも大きい星を球対称的な降着により形成することは困難である。

これらの問題点より、大質量星形成には小質量星形成のような自己重力のみによる質量降着率を超える環境を生み出す必要があると考えられる。その環境を生み出すシナリオとして現在考えられているのが、Collect & Collapse (C&C)と Cloud Cloud Collision (CCC) である。

Collect & Collapse (C&C)

1 つ目が Collect & Collapse（連鎖的星形成、以後 C & C）である。このシナリオは星の進化段階の違いが空間的に連続変化しているように観測されるから、Elmegreen & Lada 1977 により提案された[2]。C&C の模式図を図 1.4 に示す。大質量星からの輻射で生成された HII 領域(大質量星からの紫外線放射や星風などの輻射により形成される領域)の膨張や超新星爆発などによって生じた衝撃波がトリガーとなり、星形成を誘発させる。衝撃波は周囲の星間媒質を掃き集め、高密度シェルを形成する。この高密度シェルが膨張の過程で、時間が経つとともに重力不安定となり分裂し、最終的に星を形成する。C&C

のような、個々の分子雲内で誘発によって星形成が起こるという説は興味深い可能性であり、理論的、観測的に多くの関心を集めている。しかし、C&C にはそれを証明できる観測事例が存在しない。Dale et al. 2015 では、シミュレーションによって誘発された星形成が存在することを明確に示すことができたが、誘発的な星と自発的な星が空間的に混在しており、この 2 つの集団を区別することは困難であると主張している [3] 。つまり、星々が空間的に連続していたとしても、大質量星からの輻射によりトリガーされた星と同じ位置や速度空間に、自発的な星形成が再分布している可能性があるということである。

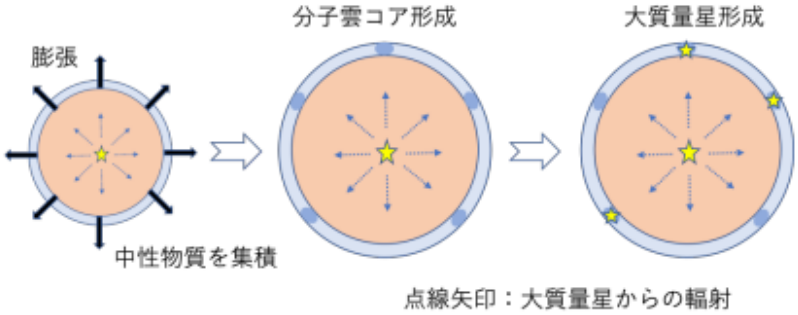

図 1.4: C&C の模式図。H II 領域などの膨張により、周囲の物質が掃き集められ、
星を誘発的に形成する。

Cloud Cloud Collision（CCC）

2 つ目が Cloud Cloud Collision（以後、CCC）である。CCC は、2 つのサイズの異なる分子雲が衝突することで、ガスが急激に圧縮され、乱流速度・磁場の効果が高まった圧縮層で大質量星が形成されるというシナリオである。分子雲衝突による大質量星形成現場は 50 ヶ所以上確認されている。CCC は図 1.5 のように、小分子雲が大分子雲中に空洞を作り空洞の「底」に最も密度の高いガス塊が形成されて重力的に不安定になり、大質量星の形成に至る（Habe–Ohta タイプ）。特徴として、衝突により生じる U 字型空洞がある。実際、大質量星形成領域 RCW 120 (図 1.6)などの観測によって U 字型空洞がしばしば観測されており、大質量星はその底に位置する。RCW 120 の場合、U 字型の圧縮層には数十個の小質量星も付随しており、これらも衝突によって形成されたと推測される。CCC と判断されたほとんどが Habe–Ohta タイプであり、衝突の典型的な描像を与えている。Habe-Ohta タイプを発展させた数値計算により、衝突による「大分子雲の空洞」と「小分子雲」が「相補的分布」を示すこと、および両分子雲間には速度的に「ブリッジ」がつくられることを示されている。これら 2 つの衝突のサインを用いて、多くのサンプルの系統的同定が可能になった[4]。

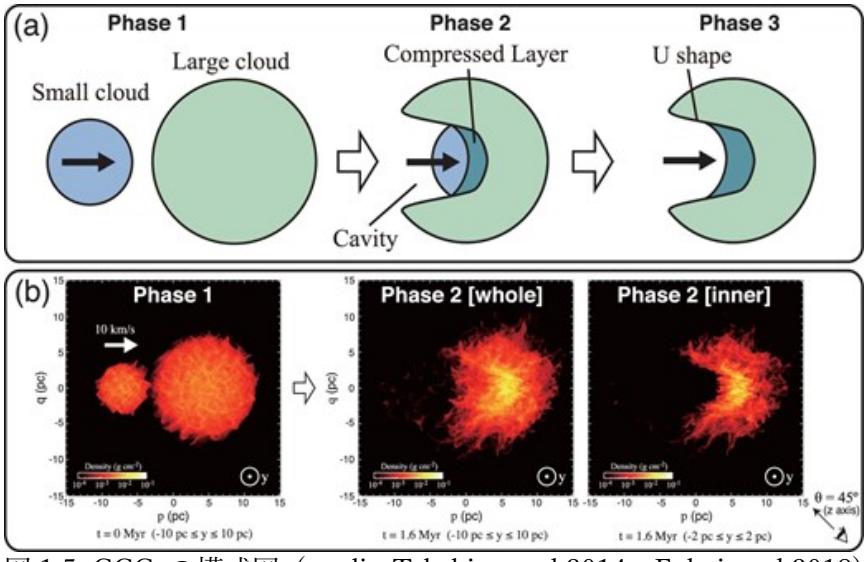

図 1.5: CCC の模式図（credit: Takahira et al.2014、Fukui et al.2018）


図 1.6: RCW120（credit: ESA/Herschel/PACS, SPIRE/Hi-GAL Project）

1.3 Spitzer bubble

Spitzer/GLIMPSE[5]8μmと Spitzer/MIPSGAL [6] 24μmの観測により、銀河面付近において、波長 8 μmの赤外線に 24 μmを内包したバブル状の構造を持つ天体が多数発見された。8 μm の波長帯は、大質量星の紫外線放射による多環芳香族炭化水素（Polycyclic Aromatic Hydrocarbons: PAH）の励起をトレースしており、24 μm はダストによる熱的な放射をトレースしており、これらの天体は赤外線バブル構造、または spitzer bubble と呼ばれる。図 1.7 に、本研究で使用した領域である Cygnus-X 領域で検出された spitzer bubble の例をいくつか示す。

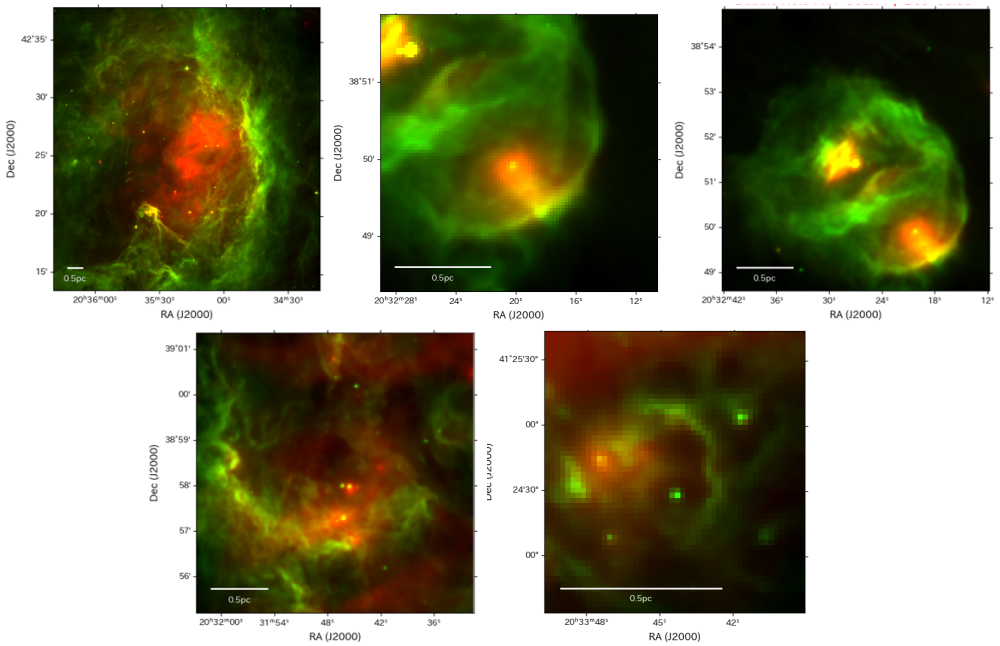

図 1.7: Cygnus-X 領域で検出された spitzer bubble の例
（credit: NASA/IPAC Infrared Science Archive(本研究において可視化、解析を行なった)）

spitzer bubble は以下のような星によって作られていると考えられている[7]。

(1) 強い恒星風を持つ星
(2) すべての光度階級の O 型星や B 型超巨星
(3) 風が弱く周囲の HII 領域の構造に大きな影響を与えない O9 型星 B3 型矮星
(4) HII 領域を作ることはできないが、放射圧によって小さなバブルを作るほど強い放射場を持つ B4 型星や低温の矮星

このような特徴から、spitzer bubble は星形成の良いトレーサーとして用いることができる。そして、星とその周辺環境との相互作用を理解する重要な手掛かりとなりうる。例えば、バブルの大きさ、形態、赤外線光度から、恒星の光度、恒星風の強さ、周囲の星間物質（ISM）密度のような物理的数値を得られる可能性や、実際に大質量星周辺の分子雲の特性を調査するにあたって、spitzer bubble が検出された領域が調査を行う際の参考になる。また、本研究では spitzer bubble が存在する領域の分子雲が、実際に他の領域の分子雲とは異なる空間、速度分布の特性を持っているのかどうかを統計的に判断することを目的とした。

1.4 先行研究

1.4.1 教師なし機械学習を用いた分子雲銀河サーベイデータの解析

本研究は、名古屋大学の山口知留による修士論文「教師なし機械学習を用いた分子雲銀河サーベイデータの解析」(Yamaguchi 2024) で用いられた手法を参考にしている[8]。Yamaguchi 2024 では、野辺山 45m 電波望遠鏡を用いて行われた FUGIN と呼ばれる、銀経10°~50°（中心側）,198°~236°（外縁側）、銀緯−1~1°の分子雲サーベイプロジェクトで得られた大規模な ^{12}CO(J=1-0)輝線データを、CAE(2.3 節にて詳しく述べる)と呼ばれる機械学習モデルを用いて分子雲の特性の評価が行われた。CAE は入力データの再現を行う機械学習モデルであり、画像再構築に際に潜在変数と呼ばれる入力されたデータの特徴量が圧縮された変数を生成する。この潜在変数を用いて天の川銀河の中心側と外縁側の分子雲構造の比較、分子雲が存在する銀河座標と各潜在変数の相関の算出、バブルが付随するか否かによる分子雲構造の比較が行われた。バブルが付随するか否かによる分子雲構造の比較実験に関しては、有意な違いは見られないという結論となった。しかし、Yamaguchi 2024 で行われた実験には、FUGIN データで観測された分子雲は地上からの距離が曖昧である点や、学習データに対するノイズ処理の問題から、CAE がノイズの特徴を重要な特徴であると捉えている可能性が高いという問題点があった。

1.4.2 Milky Way Project

Milky Way Project（以後、MWP）は、Simpson et al. 2012 によって行われた、市民参加型の赤外線バブル検出プロジェクトである[9]。MWP では、Spitzer Space Telescope の GLIMPSE(8 μm) および MIPSGAL(24 μm)データを基に、多数の市民科学者が提供したバブル候補を統合することによって、信頼性の高い赤外線バブルのカタログの作成が行われた。これにより、バブルの位置、サイズ、形状が体系的に記録され、大規模な統計解析が可能となった。バブルの多くは H_{II} 領域に対応し、大質量星形成との関連が示唆されている。本研究では、この MWP によって得られたバブルの座標データと、1.4.3 項で述べる Nishimoto et al. (2025)で検出されたバブル座標を利用して実験を行った。

1.4.3 深層学習を用いた Spitzer bubble の検出

Nhishimoto et al. 2025, "深層学習を用いた Spitzer bubble の検出" [10] では、物体検出の深層学習モデルである Single Shot multibox Detector (SSD)を用いて銀河面広域のバブル検出が行われた。バブルの訓練、評価データには、MWP によって得られた赤外線バブルが使用された。学習によって得られたモデルを用いた銀河面広域のバブル検出の結果、未検出のバブルが計 1413 個検出された。本研究で使用した領域である Cygnus-X 領域でもバブル検出が行われており、計 40 個のバブルが新たに検出された。本研究では、1.4.2 項で述べた MWP により検出されたバブルに加え、Nishimoto et al. 2025 により検出された spitzer bubble の座標を利用した。

1.5　Cygnus-X 領域

Cygnus-X は、太陽系から約 4500 光年(1400pc)の距離に存在する領域であり、星形成が活発に行われている領域として知られている。また、この領域では spitzer bubble が、MWP により 47 個、Nishimoto et al. 2025 により新たに 40 個検出されている。これらのことより、本研究の目的であるバブル領域と一般的な分子雲の特性を比較する対象として最適である。分子雲のデータには、野辺山 45m で観測された ^{2}CO(J=1-0)輝線データを用いた[11]。図 1.8 に本研究で用いた ^{12}CO(J=1-0)データの観測諸元を、図 1.9 と図 1.10 に Cygnus-X 領域のスピッツァー宇宙望遠鏡で撮影された赤外線データと、野辺山 45m 電波望遠鏡で撮影された ^{12}CO(J=1-0)輝線データを可視化したものを示す。

観測輝線	ビームサイズ	ピクセルグリッド	視線速度範囲	速度分解能	ノイズレベル
12CO(J=1-0)	16[arcsec]	7.5[arcsec]	-40~40[km/s]	0.25[km/s]	3.14[K]

図 1.8: 野辺山 45m 電波望遠鏡で観測された ^{12}CO(J=1-0)輝線の観測諸元

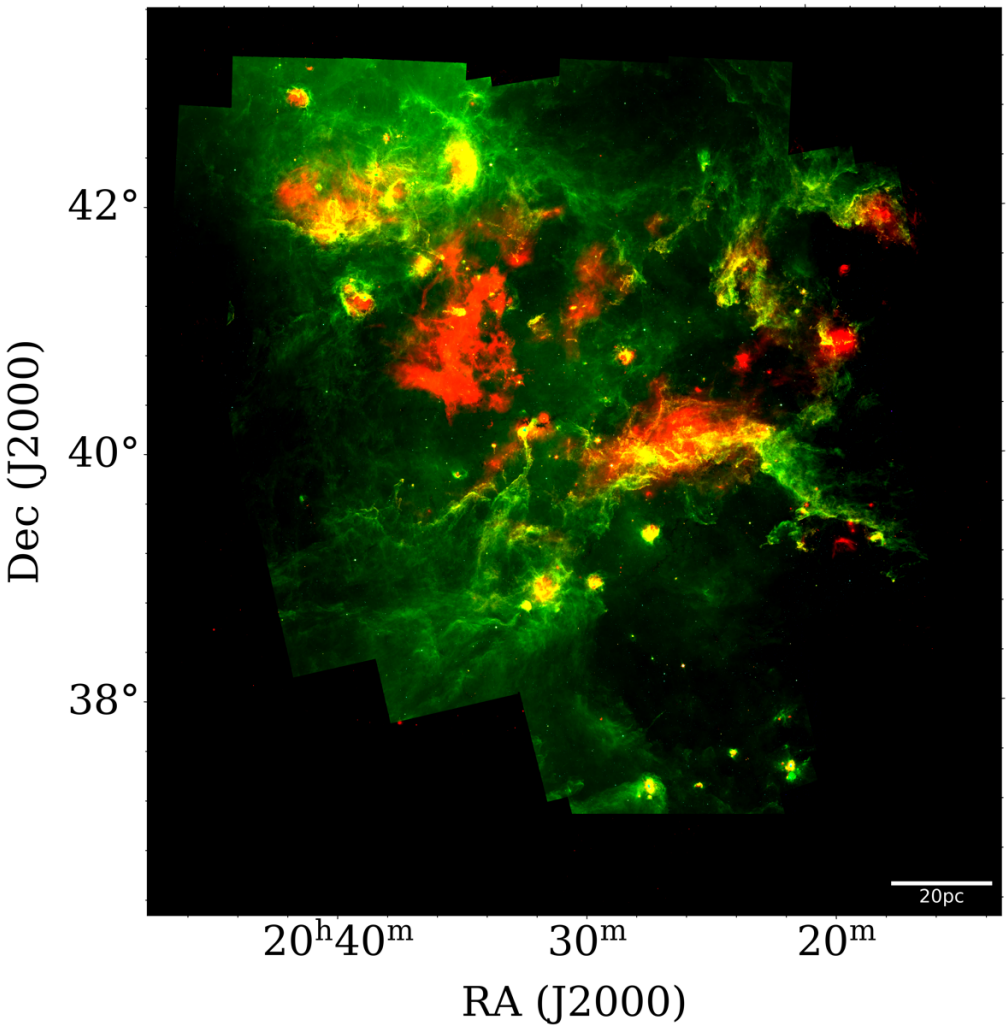

図 1.9: スピッツァー宇宙望遠鏡によって撮影された Cygnus-X 領域の赤外線データを可視化した画像。8μmを緑、24μmを赤に着色している。

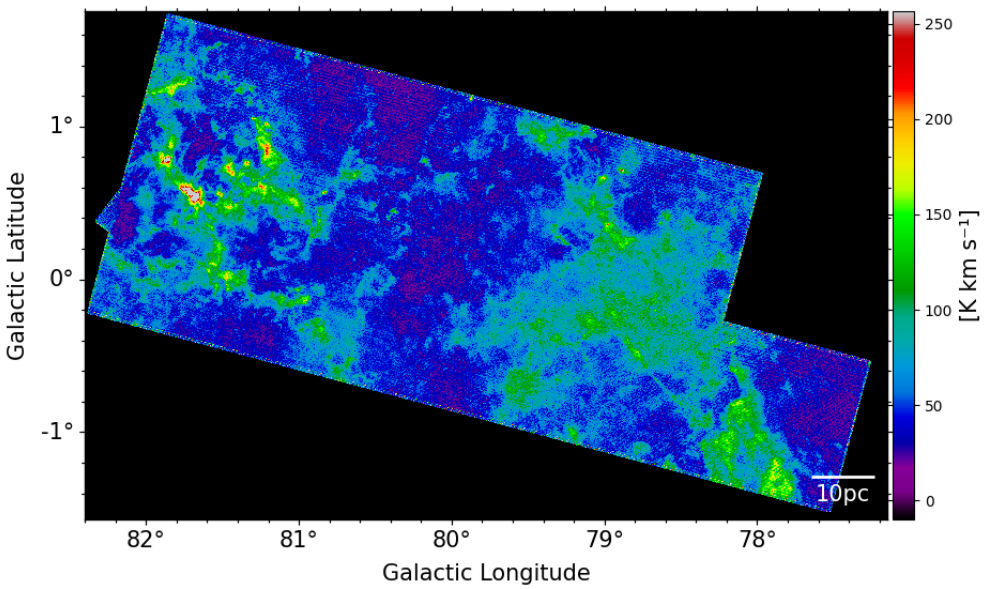

図 1.10: 野辺山 45m 電波望遠鏡で観測された Cygnus-X 領域の ^{12}CO(J=1-0)輝線データの積分強度図

1.6 本研究の目的

大質量星周辺の分子雲の特性を理解することは、大質量星自体の形成過程の解明、ひていは銀河進化の理解に繋がる。本研究の目的は、野辺山 45m 電波望遠鏡によって観測された、天の川銀河内で大質量形成が活発な領域の一つである Cygnus-X 全域の ^{12}CO(J=1-0)輝線データを用いて、先行研究により検出された spitzer bubble 周辺の分子雲が、実際に大質量星からの輻射による影響を受け、空間軸方向、速度軸方向へ、一般的な分子雲と比べて異なる特性を持っているのかどうかを統計的に解析することである。検証には、広大な 3 次元分子雲データの人の手による統計的解析は非常に困難である点から、機械学習の手法を用いて検証を行なった。

第 2 章　検証手法

Spitzer bubble が存在する領域（以降バブル領域）の分子雲はその性質上、中心星からの影響でx, yの空間 2 次元はもちろん、速度軸方向への影響も大きく受けている可能性がある。そのため、バブル領域の分子雲と一般的な分子雲の特徴の違いを検証する際には、x, yの空間 2 次元に加え、速度軸vの次元を加えた計 3 次元データの解析を行うことが有用であると言える。しかし膨大な天文データの 3 次元的解析は非常に困難であるという問題があった。そのため本研究では機械学習の手法を用いて、バブル領域と一般的な領域の分子雲の特徴量の違いを統計的に解析した。機械学習の基本的な仕組みとして知られているのが、人間の脳の神経回路網を人工ニューロンという数式的なモデルで表現したニューラルネットワークである。分子雲のデータから特徴を抽出する手段として、このニュータルネットワークを用いた機械学習の手法である CAE(Convolutional Autoencoder)というモデルを用いた。このモデルは、入力データの特徴量を任意の数の潜在変数と呼ばれる変数に圧縮することができる。バブル領域と一般的な領域の分子雲から抽出した潜在変数の比較を行う際には、K-S 検定(コルゴモロフ-スミルノフ検定)と呼ばれる、二つのヒストグラムの分布の違いを設定した閾値をもとに判定する手法を用いた。本章では、機械学習の概要と手法の解説、今回用いた機械学習モデルである CAE の解説、K-S 検定の詳細について順に述べる。

2.1　機械学習とは

機械学習とは、コンピュータが既存のデータからパターンを学習することで、同じような課題に対し、学習した内容を通してより迅速かつ正確に処理を行うことができるようになる手法のことである。AI と機械学習の違いは、AI は機械が人間の知能を模倣し実行する技術の総称を指すのに対し、機械学習は AI の中の一部分として位置付けられる領域であり、大量のデータを使用して機械に特定のタスクを学習させる技術のことを言う。機械学習モデルを訓練させる方法には主に「強化学習」と「教師あり学習」、「教師なし学習」の三つが存在し、本研究ではこの中の「教師なし学習」の手法を用いた。

2.1.1　強化学習

強化学習は、機械が試行錯誤を繰り返しながら、最良の行動を学んでいく手法である。この学習方法は、よく子どもがゲームをプレイして上手くなっていく過程によく例えられる。強化学習は主に「囲碁 AI」や「将棋 AI」などに活用されている学習方法である。

2.1.2　教師あり学習

教師あり学習は、機械に「これはこれに対応する」といったように、正解を具体的に教えながら学習させる方法である。この中には「分類モデル」と「回帰モデル」という二つの主要手法がある。

「分類モデル」は、入力データを特定のカテゴリに分けるタスクで、例えば写真に写っているのが狼か犬かの判断を行うことや、数字の画像の場合には写っているのがどの数字にあたるのかを分類するなどの活用法がある。一方「回帰モデル」は、数値を予測するタスクに使用される。例えば、家の大きさや立地条件などから、家の価格を予測することや、競輪の順位予想にも使用されるケースなどがある。

この学習方法では、明確な答えを持つ大量のデータで学習を行うため、後に新たなデータが与えられた

際に、高い精度で予測や分類ができるという強みがある。

2.1.3　教師なし学習

教師なし学習は、正解を与えない状態で、その中に潜むパターンやグループを自動的に見つける機械学習の手法である。代表的なものとして、「クラスタリグ」と「次元削減」が挙げられる。

「クラスタリング」では、多数の顧客の購買データがある場合、どの顧客が似たような傾向を持っているのかを自動でグループ化することが可能である。「次元削減」では、多くの情報を持つ複雑なデータをわかりやすくシンプルな形に変換することができる。

本研究では、「教師なし学習」の「次元削減」の手法を用いることによって、分子雲の三次元データの特徴量を潜在変数と呼ばれる任意の数の変数に圧縮することによって分子雲の特徴を評価した。

2.2　ニューラルネットワーク（Neural Network）

ニューラルネットワークは、入力層、隠れ層、出力層から構成され、層と層の間には入力信号の重要性を決定する「重み」や「バイアス」といったパラメータと、重み付けが行われた信号を次の層への出力へと変換する活性化関数が存在する。ニューラルネットワークはこれらのような役割を持った関数を組み合わせることによって、人間の脳内にある神経細胞（ニューロン）とそのつながりである神経回路網を、人工ニューロンという数式的なモデルで表現したものである。図 2.1 にニューロンの一つを数理モデル化した模式図を示す。そして機械学習モデルに「学習をさせる」とは、これらの関数のパラメータを目的に対して最適な値に調整していくことに他ならず、ニューラルネットワークの層をより深くすることにより学習パラメータの数を増やし、より複雑な問題に対して取り組めるようにしたものが深層学習、通称ディープラーニングと呼ばれるものである。本節では、ニューラルネットワークの構造と構成要素について、本研究で使用したものを中心に述べる。

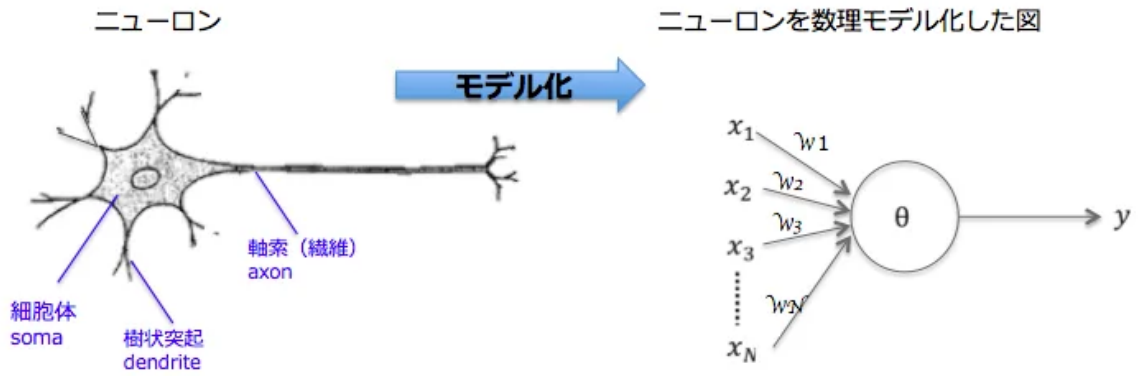

図 2.1: 左図が人間の脳内のニューロンを表しており、右図がそれを数理モデル化したものである。図中には重みのみが記されているが、一般的には重み付けされた入力信号に対してバイアスと呼ばれる数値が加算されたものが次の層へと出力される。
(credit: Udemy メディア. (2024). ニューラルネットワークとは？人工知能の基本を初心者向けに解説！. [オンライン記事より引用])

2.2.1　ニューラルネットワークの構造

ニューラルネットワークとは、人間の脳内にある神経細胞（ニューロン）とその繋がりである神経回路網を、人工ニューロンという数式的なモデルで表現したものである。図 2.2 の模式図に示すように、ニューラルネットワークは複数のノードが連なることによって構成される。

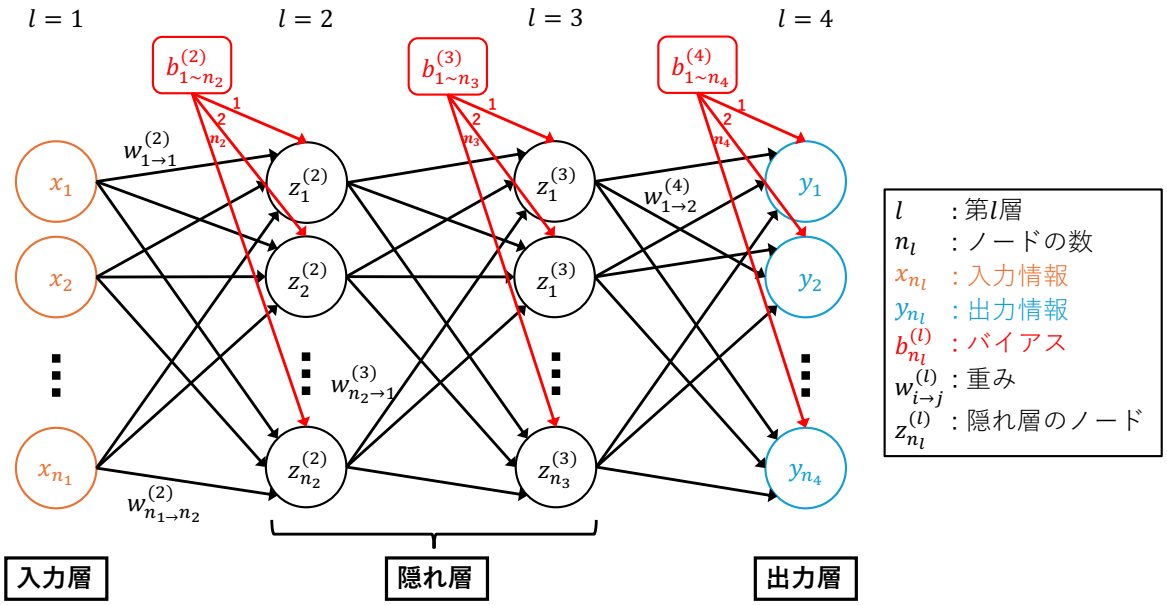

図 2.2: ニューラルネットワークの模式図。右上の添字（例えば$b^{(l)}$の(l)にあたる箇所）は、そのパラメータが何層目のパラメータなのかを示す。$w^{(l)}_{i\to j}$の右下の添字（$i\to j$）は、一つ前のi番目のノードから次のj番目のノードに対する重みであることを表す。紙面の都合上簡略化されているが、実際には図中のすべてのノードからノードへの矢印に対して重みが存在する。

図 2.2 中の◯内に書かれた文字はそのノードの出力を表し、その中でもxは入力情報を、yは最終的な出力情報を表す。一般的にニューラルネットワークを構成する際には、図 2.2 中には示されていないが、ノードからノードへ出力値が伝わるに際に、活性化関数と呼ばれる関数を通してから次のノードへの出力が行われる。詳しくは 2.2.2 項にて述べる。ニューラルネットワークでは縦に並んだノードを一つのまとまりとして「層」と呼び、特に第 1 層はニューラルネットワークに入力を与えるという意味で「入力層」、最後の層は最終的な出力をするので「出力層」と呼ばれる。そして入力層と出力層の間にある層はまとめて「隠れ層」（または「中間層」）と呼ばれる。図 2.2 は入力層と出力層と隠れ層 2 層からなる、計 4 層のニューラルネットワークと言える。

入力層、隠れ層、出力層のそれぞれの役割と内部で行われる計算の詳細を以下に示す。

入力層

入力層は文字通り、ニューラルネットワークに入力を与える層である。入力データが画像の場合には、画像データを数値情報としてニューラルネットワークに入力する。具体的には入力層入力画像の画素数だけノードが並び、それぞれの画素の明るさを示す輝度値 $x_1, x_2\cdots$ が入力として与えられる。例えば、3.1 節で後述する今回使用した学習データのように、(高さ, 幅, 深さ)=(112, 112, 12)の形状を持つ三次元データであれば、図 2.2 の入力層には、n_1=112×112×12=150,528 個のノードが並ぶことになる。

隠れ層

図 2.2 に示す第l層の第j番目のノード、すなわち $z_j^{(l)}$値が決定される過程について考える。このとき、本来であれば活性化関数と呼ばれる関数を通す過程が含まれるが、ここでは割愛する（詳しくは 2.2.2 項にて述べる)。このノードは第$l-1$層からの出力 $z_i^{(l-1)}$（iは 1 からn_{l-1}までの整数）を入力として受け取り、そこに重み $w_i^{(l)}$が掛け算される。そして計算された値$z_i^{(l-1)}$（iは 1 からn_{l-1}までの整数）の合計値にバイアス$b^{(l-1)}$を加えた値が$z_j^{(l)}$となる。つまり、$z_j^{(l)}$の値は(2.1)の計算式で与えられる。

$$z_j^{(l)} = \sum_{i=1}^{n_{l-1}} z_i^{(l-1)} w_{i\to j}^{(l)} + b^{(l)} \tag{2.1}$$

実際に隠れ層での計算を実行するときには、計算時間の短縮のため行列の乗法を利用する。具体的には第$l-1$層のi番目のノードと第l層のj番目のノードの結合の重み$w_{i\to j}^{(l)}$を$(i,\ j)$成分とする$n_l \times n_{l-1}$行列 $\boldsymbol{W^{(l)}}$、第$l-1$層のノードから第l層のj番目のノードへのバイアス$b_j^{(l)}$を表すn_l×1 行列$\boldsymbol{b^{(l)}}$、第$l-1$層のノードの出力値を$z_i^{(l-1)}$表すn_l×1 行列$\boldsymbol{z^{(l-1)}}$を導入する。

$$\boldsymbol{W^{(l)}} = \begin{pmatrix} w_{1\to1}^{(l)} & w_{2\to1}^{(l)} & \cdots & w_{n_{l-1}\to1}^{(l)} \\ w_{1\to2}^{(l)} & w_{2\to2}^{(l)} & \cdots & w_{n_{l-1}\to2}^{(l)} \\ \vdots & \vdots & \ddots & \vdots \\ w_{1\to n_l}^{(l)} & w_{2\to n_l}^{(l)} & \cdots & w_{n_{l-1}\to n_l}^{(l)} \end{pmatrix} \tag{2.2}$$

$$\boldsymbol{b^{(l)}} = \begin{pmatrix} b_1^{(l)} \\ b_2^{(l)} \\ \vdots \\ b_{n_l}^{(l)} \end{pmatrix} \tag{2.3}$$

$$\boldsymbol{z^{(l-1)}} = \begin{pmatrix} z_1^{(l-1)} \\ z_2^{(l-1)} \\ \vdots \\ z_{n_{l-1}}^{(l-1)} \end{pmatrix} \tag{2.4}$$

式 2.2~2.4 を用いて隠れ層の計算を書き換えると、式 2.5 のようになる。

$$\boldsymbol{z^{(l)}} = \boldsymbol{W^{(l)} z^{(l-1)}} + \boldsymbol{b^{(l)}} \tag{2.5}$$

また、ここまでの計算では重みwとバイアスbは区別して表現してきたが、以降では次のようにバイアスを重みに組み込んで考えていく。

$$\boldsymbol{W^{(l)} z^{(l-1)}} + \boldsymbol{b^{(l)}} \tag{2.6}$$

$$= \begin{pmatrix} w_{1\to1}^{(l)} & w_{2\to1}^{(l)} & \cdots & w_{n_{l-1}\to1}^{(l)} \\ w_{1\to2}^{(l)} & w_{2\to2}^{(l)} & \cdots & w_{n_{l-1}\to2}^{(l)} \\ \vdots & \vdots & \ddots & \vdots \\ w_{1\to n_l}^{(l)} & w_{2\to n_l}^{(l)} & \cdots & w_{n_{l-1}\to n_l}^{(l)} \end{pmatrix} \begin{pmatrix} z_1^{(l-1)} \\ z_2^{(l-1)} \\ \vdots \\ z_{n_{l-1}}^{(l-1)} \end{pmatrix} + \begin{pmatrix} b_1^{(l)} \\ b_2^{(l)} \\ \vdots \\ b_{n_l}^{(l)} \end{pmatrix} \tag{2.7}$$

$$= \begin{pmatrix} b_1^{(l)} & w_{1\to1}^{(l)} & w_{2\to1}^{(l)} & \cdots & w_{n_{l-1}\to1}^{(l)} \\ b_2^{(l)} & w_{1\to2}^{(l)} & w_{2\to2}^{(l)} & \cdots & w_{n_{l-1}\to2}^{(l)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ b_{n_l}^{(l)} & w_{1\to n_l}^{(l)} & w_{2\to n_l}^{(l)} & \cdots & w_{n_{l-1}\to n_l}^{(l)} \end{pmatrix} \begin{pmatrix} z_1^{(l-1)} \\ z_2^{(l-1)} \\ \vdots \\ z_{n_{l-1}}^{(l-1)} \end{pmatrix} \tag{2.8}$$

$$\begin{pmatrix} b_1^{(l)} \\ b_2^{(l)} \\ \vdots \\ b_{n_l}^{(l)} \end{pmatrix} = \begin{pmatrix} w_{0\to1}^{(l)} \\ w_{0\to2}^{(l)} \\ \vdots \\ w_{0\to n_l}^{(l)} \end{pmatrix} \tag{2.9}$$

とおいて、

$$= \begin{pmatrix} w_{0\to1}^{(l)} & w_{1\to1}^{(l)} & w_{2\to1}^{(l)} & \cdots & w_{n_{l-1}\to1}^{(l)} \\ w_{0\to2}^{(l)} & w_{1\to2}^{(l)} & w_{2\to2}^{(l)} & \cdots & w_{n_{l-1}\to2}^{(l)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ w_{0\to n_l}^{(l)} & w_{1\to n_l}^{(l)} & w_{2\to n_l}^{(l)} & \cdots & w_{n_{l-1}\to n_l}^{(l)} \end{pmatrix} \begin{pmatrix} z_1^{(l-1)} \\ z_2^{(l-1)} \\ \vdots \\ z_{n_{l-1}}^{(l-1)} \end{pmatrix} \tag{2.10}$$

ここで、

$$\boldsymbol{W}^{(l)} = \begin{pmatrix} w_{0\to1}^{(l)} & w_{1\to1}^{(l)} & w_{2\to1}^{(l)} & \cdots & w_{n_{l-1}\to1}^{(l)} \\ w_{0\to2}^{(l)} & w_{1\to2}^{(l)} & w_{2\to2}^{(l)} & \cdots & w_{n_{l-1}\to2}^{(l)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ w_{0\to n_l}^{(l)} & w_{1\to n_l}^{(l)} & w_{2\to n_l}^{(l)} & \cdots & w_{n_{l-1}\to n_l}^{(l)} \end{pmatrix} \tag{2.11}$$

と新たに置くと、式 2.5 は以下のように書ける。

$$\boldsymbol{z^{(l)} = W^{(l)} z^{(l-1)}} \tag{2.12}$$

ここで説明した計算方法を用いた層は、前の層の全てのノードの情報を入力として計算を行うため、全結合層と呼ばれる。隠れ層の種類は他にも存在し、本研究では全結合層の他に畳み込み層を用いた。畳み込み層については 2.2.4 項にて詳しく述べる。

出力層

ニューラルネットワークの最後の層を出力層と呼ぶ。計算方法は中間層での計算と同じであるため、出力層が第L層であるとると出力$\boldsymbol{y}$は、式 2.13 のように書くことができる。

$$\boldsymbol{y = W^{(L)} z^{(L-1)}} \tag{2.13}$$

2.2.2　活性化関数

ニューラルネットワークの各層の計算は、基本的に式 2.12 のようになっている。このことから、例えば入力$\boldsymbol{x}$を持つ 4 層のニューラルネットワークの出力$\boldsymbol{y}$は式 2.14~2.16 のように表すことができる。

$$\boldsymbol{z}^{(2)} = \boldsymbol{W}^{(2)}\boldsymbol{x} \tag{2.14}$$
$$\boldsymbol{z}^{(3)} = \boldsymbol{W}^{(3)}\boldsymbol{W}^{(2)}\boldsymbol{x} \tag{2.15}$$
$$\boldsymbol{y} = \boldsymbol{W}^{(4)}\boldsymbol{W}^{(3)}\boldsymbol{W}^{(2)}\boldsymbol{x} \tag{2.16}$$

ここで注目したいのが、出力$\boldsymbol{y}$が単なる線形関数になってしまっている点である。これはつまり、どれだけニューラルネットワークの層を増やしたとしても、実質 1 層の線形モデルと変わらないということになる。そしてこの問題を解決するのが活性化関数である。活性化関数をニューラルネットワークに組み込むことで、ネットワークが非線形性を持ち、より複雑な関数を表現できるようになる。

本研究で使用した活性化関数は次の二つである。

ReLU 関数

ReLU 関数は、$x < 0$では定数 0、$x \geqq 0$では$y = x$ の 1 次関数という非常に簡単な関数である。簡単ではあるが、$x = 0$で別の関数に切り替わるため、結果的に「非線形関数」の 1 つとなり、活性化関数としての意味を持つようになる。

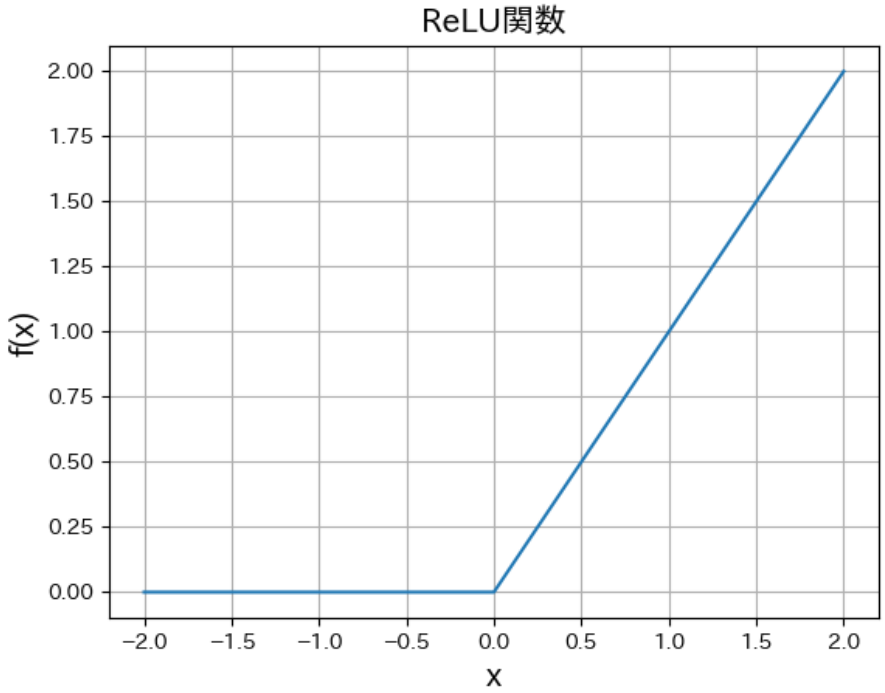

図 2.3: ReLU 関数のプロット図

Sigmoid 関数

シグモイド関数は式 2.17 に表す、値が 0~1 で出力される非線形関数である。また、本研究ではこのシグモイド関数を 2.3 節で述べる CAE の出力部である Decoder の最後に適応することにより、出力データに対し 0~1 に正規化を行なった。

$$f(x) = \frac{1}{1 + \exp(-x)} \tag{2.17}$$

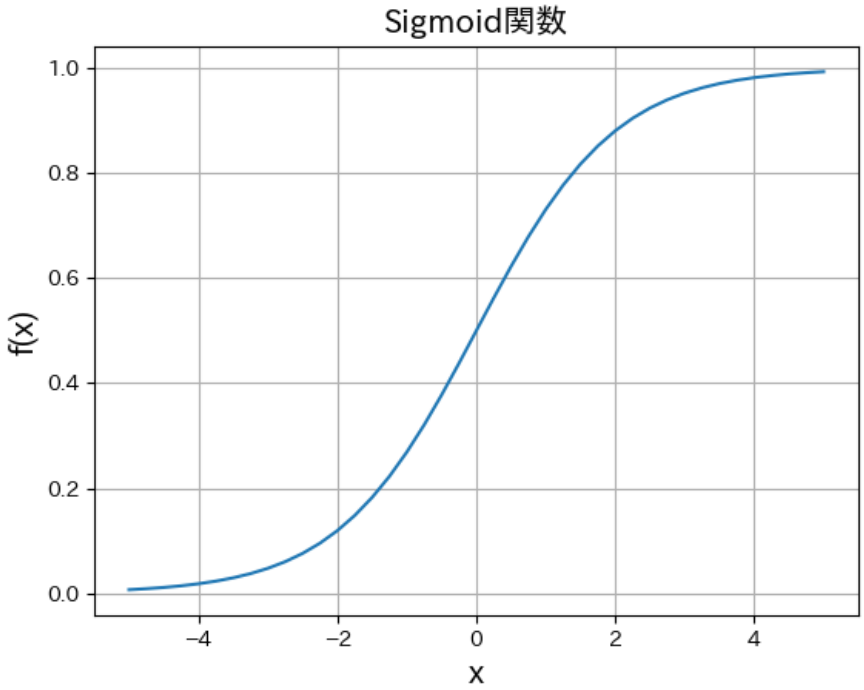

図 2.4: Sigmoid 関数のプロット図

これらのような活性化関数を、各ノードが値を出力する前に入れることにより、ニューラルネットワークが非線形性をもつことが可能になる。第l層の活性化関数を$\sigma^{(l)}(x)$として、ニューラルネットワークに活性化関数を組み込んだ場合の、図 2.2 中で$z_1^{(2)}$の値を出力するノードに注目した計算過程の模式図と、式 2.8 の入力$\boldsymbol{x}$を持つ 4 層のニューラルネットワークの出力$\boldsymbol{y}$の計算式を式 2.18 と図 2.5 に示す。

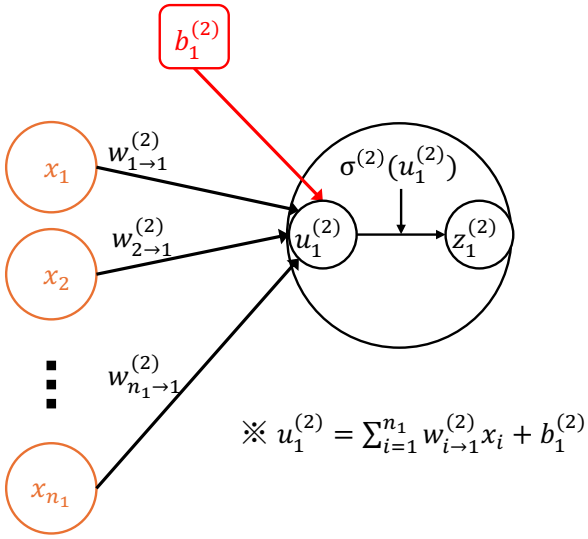

図 2.5: 活性化関数を導入した場合の図 2.2 における $z_1^{(2)}$の値を出力するニューロン内で行われる計算過程の模式図。$u_1^{(2)}$は、活性化関数を通す前の値であり、計算式は図中の通り。

$$\boldsymbol{y} = \boldsymbol{\sigma}^{(4)}\left(\boldsymbol{W}^{(4)}\boldsymbol{\sigma}^{(3)}\left(\boldsymbol{W}^{(3)}\boldsymbol{\sigma}^{(2)}\left(\boldsymbol{W}^{(2)}\boldsymbol{x}\right)\right)\right) \tag{2.18}$$

式 2.18 より、式 2.16 では単なる線形関数の出力結果であった$\boldsymbol{y}$が、非線形関数である活性化関数を通したことによって非線形性をもつ出力結果になっていることがわかる。

2.2.3 畳み込み層(Convolutional Layer)

前の層の全てのノードの情報を入力として計算を行う層は 2.2.1 項で述べたように全結合層と呼ばれる。この全結合層には大きな問題点が一つあり、それはデータの形状が ”無視” されてしまうことである。ニューラルネットワークを全結合層のみで構成することは可能であるが、画像情報を入力にする時などは、ピクセル同士の 2 次元的な（あるいは 3 次元的な）関連性を読み取る必要がある。すべての層を全結合層で構成してしまうと、データの画像情報がただの 1 次元配列として処理が進行してしまうため、入力画像の特徴を抽出するには不十分である。その問題を解決する方法として、機械学習の画像処理の分野でよく使用されるのが「畳み込み層」である。この「畳み込み層」で行われる演算処理の模式図を図 2.6 に示す。

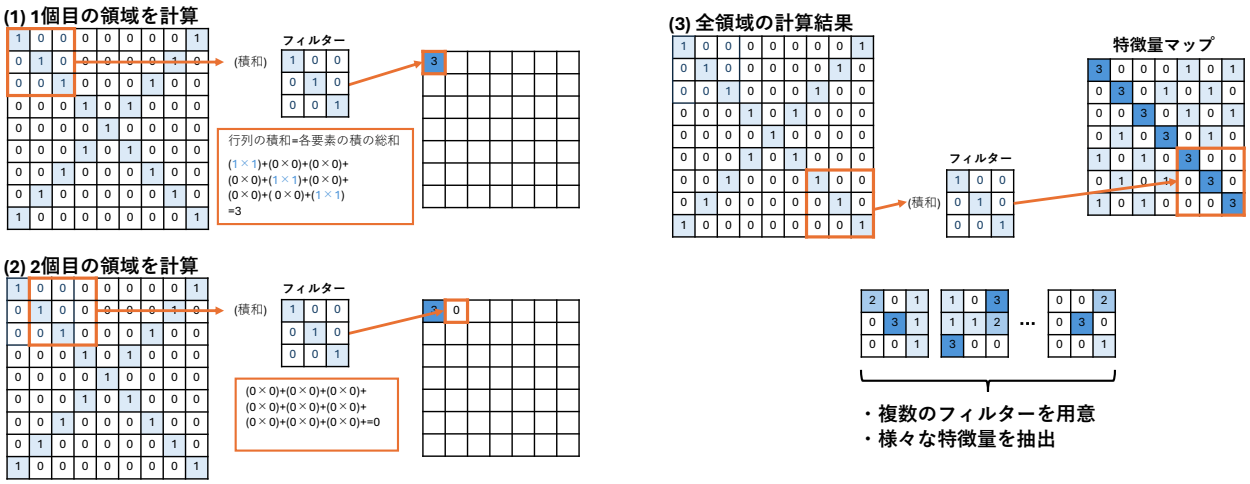

図 2.6: 畳み込みで行われる処理の模式図
(credit: 最短コースでわかる Pytorch & 深層学習プログラミング. 9 章. 畳み込み処理. より一部改変)

図 2.6 の(1)のようにフィルターを用意し、2 次元入力データとフィルターの各ピクセルを掛け算してからその合計値を新たな配列のピクセル値として格納する。その後(2)のようにフィルターを移動し同様の演算を行う。それを全領域に対して繰り返し行い、最終的に(3)のように、用意したフィルターの形状が強調され、かつデータの形状が小さくなった 2 次元配列が出力される。この特徴量が抽出されたデータ配列のことを「特徴量マップ」と呼ぶ。そしてこのフィルターを複数用意し、入力データの様々な特徴量の抽出とデータの圧縮を行っていく。畳み込み層を用いた機械学習モデルでは、入力データの特徴量をより高い精度で抽出できるフィルターのパラメータが学習によって決定される。

3 次元畳み込み演算

3 次元畳み込み演算は、文字通り 3 次元的な特徴を捉えることができる処理である。3 次元データは、本研究で用いたデータの他に動画データなどが挙げられる。基本的な計算手法は先述した 2 次元の畳み込み演算と同様であり、フィルターのサイズで積和計算を行う。ただし 3 次元畳み込み演算の場合は図 2.7 に示すように、入力が 3 次元になることによってフィルターと出力も 3 次元の形状をとる。図 2.7 の模式図のように、3 次元畳み込み層においてはフィルターを 3 方向に一定間隔で動かしていき、3 次元的な特徴を捉えた出力を得る。

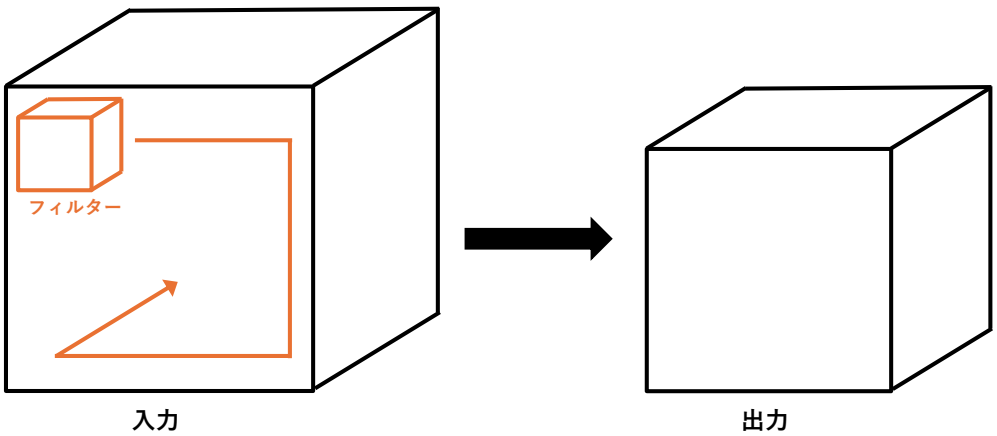

図 2.7: 3 次元畳み込み処理の模式図

2.2.4 CNN (Convolutional Neural Network)

畳み込みニューラルネットワーク(Convolutional Neural Network: CNN)とは畳み込み層を組み込んだニューラルネットワークである。本研究では、この CNN の手法の一つである CAE(Convolutional Auto-Encoder)と呼ばれる機械学習モデルを用いた。2.3 節で、CAE の概要と本研究で用いたモデル構造について述べる。

2.3 CAE(Convolutional Auto-Encoder)

本節では今回の実験で使用した教師なし機械学習モデルである、畳み込みオートエンコーダ(Convolutional Auto-Encoder: CAE)について記述する。CAE は 2.1.3 項で述べた、次元削除を行うことのできる機械学習モデルであり、入力データの本質的な特徴を抽出し、より少ないパラーメタで入力データを表現することができる。CAE の概要と本研究で用いたモデル構造の詳細について順に述べる。

2.3.1 CAE の概要

CAE とは、2.2.4 項で述べた CNN の手法の一種であり、入力データの次元削除を行うことのできる機械学習モデルである。CAE の内部で行われる処理の模式図を図 2.8 に示す。

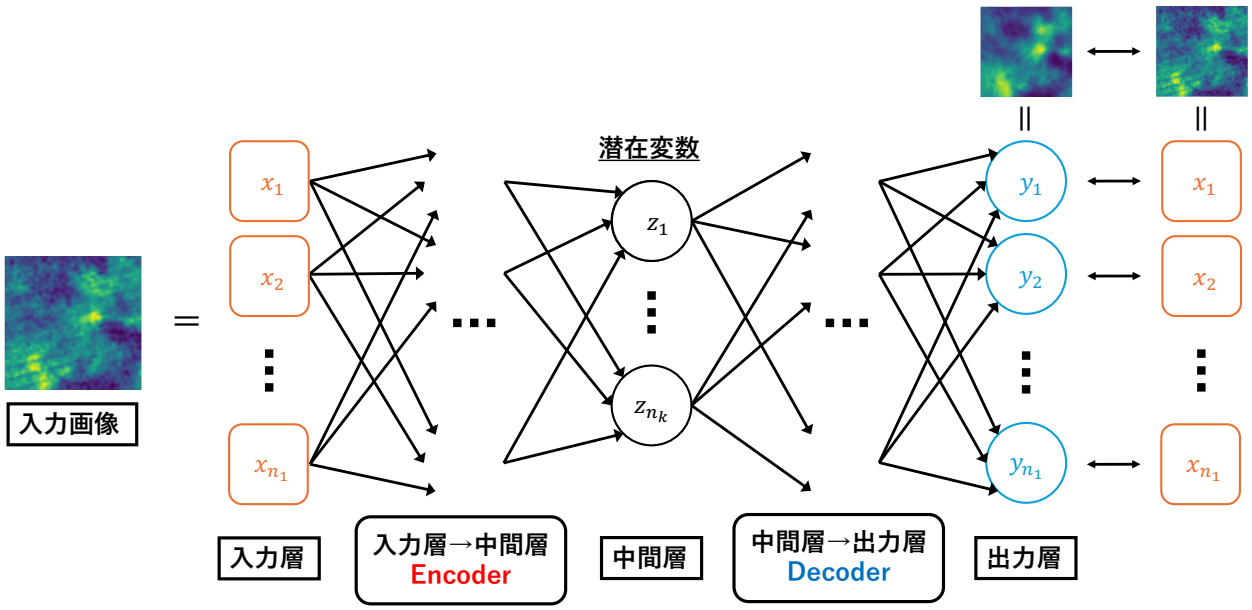

図 2.8: CAE で行われる内部処理の模式図

入力画像の配列を$\boldsymbol{x}$とする。入力層には入力画像のピクセル数だけノードが並ぶことになり、本研究で使用したデータであれば、入力層には 112×112×12=150528(3.1.8 にて後述)個のノードが並ぶことになる。入力されたデータに畳み込み処理を繰り返し行い、中間層まで特徴量の抽出とデータの圧縮を行なっていく。このデータの圧縮を行う入力層から中間層までの部分は「Encoder(符号化器)」と呼ばれる。抽出された特徴量は中間層の手前で全結合層を経由し、1 次元のデータ配列となる。この入力データの特徴量が圧縮された 1 次元のデータ配列$\boldsymbol{z}$は「潜在変数」と呼ばれる。この潜在変数の数は任意に設定する

ことができるが、CAE を使用する目的は入力画像をより少ないパラメータで表現することであるため、入力画像のピクセル数をn_1、潜在変数の数をn_kとして、通常であれば$n_1 > n_k$となるように潜在変数の数を決定する。この潜在変数については 2.3.2 項にて詳しく述べる。中間層から出力層は、特徴量が圧縮された潜在変数から入力画像をできる限り復元する役割を持ち、Encoder に対して「Decoder(複合化器)」と呼ばれ、出力層のピクセル数は入力画像と同様n_1となる。

CAE の学習目標は、入力画像を潜在変数に圧縮し、圧縮された潜在変数からできる限り入力画像を再現するために最適な Encoder と Decoder 内のパラメータを決定することである。

2.3.2 潜在変数

本研究では、分子雲のデータを CAE により潜在変数に圧縮し、2.3.1 項で述べたように、CAE の学習目標は入力画像を潜在変数に圧縮し、バブル領域から抽出された潜在変数と一般的な領域の潜在変数を比較することによりバブル領域の分子雲の特性を評価した。このことから、潜在変数は本研究で非常に重要な役割を持っている。

潜在変数の数は任意の数に設定することができるが、その数を設定する際には注意が必要である。潜在変数の数は多ければ多いほど入力データのより詳細な特徴を抽出することができるが、重要ではない余分な特徴まで抽出してしまう。また、CAE の入力データを少数のパラメータで表現できるという強みを活かせなくなってしまう。逆に少なすぎれば、入力データをより少数のパラメータで議論することができるようになるが、その潜在変数は入力データの特徴を十分に捉えきれていない可能性が高い。3 章で詳しく述べるが、本研究ではこの潜在変数の数を変化させて入力データの再現度を比較する実験も行なった。

2.3.3 本研究で使用した CAE の構造

CAE の一般的な構造は図 2.8 で示した通りであるが、畳み込み層の数やフィルターの数、活性化関数の有無などは任意に設定することができる。本研究で使用した CAE のモデル構造と用語の詳細を図 2.9 に示す。

Encoder

```
Conv3d(1, 64, kernel_size=(4, 4, 4), stride=(2, 2, 2), padding=(1, 1, 1)),  #output=(6, 56, 56)
ReLU(True),
Conv3d(64, 32, kernel_size=(4, 4, 4), stride=(2, 2, 2), padding=(1, 1, 1)),  #output=(3, 28, 28)
ReLU(True),
Flatten(),
Linear(32*3*28*28, latent)
```

Decoder

```
Linear(latent, 32*3*28*28),
Unflatten(1, (32, 3, 28, 28)),  # Unflatten back to (32, 5, 25, 25)
ConvTranspose3d(32, 64, kernel_size=(4, 4, 4), stride=(2, 2, 2), padding=(0, 1, 1)), #output=(8, 56, 56)
ReLU(True),
ConvTranspose3d(64, 32, kernel_size=(3, 3, 3), stride=(1, 2, 2), padding=(0, 1, 1)), #output=(10, 111, 111)
ReLU(True),
ConvTranspose3d(32, 1, kernel_size=(3, 2, 2), stride=(1, 1, 1), padding=(0, 0, 0)), #output=(12, 112, 112)
Sigmoid()  # to scale output between 0 and 1
```

用語	説明
Conv3d	3次元畳み込み層。入力データにフィルタを適用して特徴を抽出。
ConvTranspose3d	3次元転置畳み込み（逆畳み込み）。通常の畳み込みの逆操作を行い、空間的なスケールを拡大。
kernel_size	フィルタの大きさ。`(高さ, 幅, 深さ)` の3次元で指定。
stride	フィルタの適用間隔（ストライド）。値を大きくすると出力サイズが縮小（Conv3d）または拡大（ConvTranspose3d）。
padding	入力データの周囲に追加するゼロパディングの数。出力データのサイズ調整に使用。
Flatten	多次元のデータを1次元の配列に変換。
Unflatten	`Flatten` された配列を元の形状に復元。
Linear	全結合層。データの次元を圧縮・変換する。
ReLU	活性化関数。負の値をゼロにし、非線形性を導入。
Sigmoid	出力を `[0,1]` へ正規化を行う
latent	潜在変数の数を表す。

図 2.9: 本研究で使用した CAE モデルの構造と用語の説明

本研究で行なった実験（詳細は 3 章にて後述）で使用した CAE は全てこのモデルであり、潜在変数の値のみを変化させて実験を行なった。

2.4 学習の進行

2.1~2.3 節では機械学習の概要と CAE の構造について述べた。この節では、機械学習においてどのように学習が進んでいくのかを CAE の場合に着目して述べる。

2.3.1 項で述べたように、CAE の学習目標は Encoder と Decoder 内のフィルターなどのパラメータを、圧縮された潜在変数から再現される出力画像が入力画像に近づくように調整することである。学習対象であるパラメータは、学習初期ではランダムに生成される（任意に設定することも可能）。そのパラメータを使用して入力画像の特徴を潜在変数へ圧縮、潜在変数から出力画像を生成すると、たいてい入力画像とはかけ離れた画像が出力される。そして入力画像と出力画像の各ピクセルの差分を計算し、入力と出力がどれだけ離れているのかを表す「損失(Loss)」という値を計算する。各パラメータの値を、この損失が少なくなる方向へ少しずつ動かしていき、できる限り損失が小さくなるようなパラメータを探索していくのが「学習」である。そしてこのパラメータをどのようにどれだけ動かすのかを決定するのが最適化手法と呼ばれるものである。

2.4.1 損失関数

機械学習においてモデルに学習をさせるとは、損失の値をより少なくなるようなパラメータを探索させることである。この損失の計算を行う関数のことを「損失関数」と呼び、損失関数が行う計算方法は幾つも存在し、これは機械学習が取り組む問題によって決定される。本研究で使用した損失関数は「平均二乗誤差（Mean Squared Error: MSE）」である。計算式は式 2.19 の通り。

$$Loss = \frac{1}{N}\sum_{k=1}^{n}(y_k - x_k)^2 \tag{2.19}$$

Nはモデルに一度に入力するデータの数（バッチサイズと呼ばれる）、nは入出力データのピクセル数、kはピクセル番号、x_kはk番目の入力データのピクセル強度、y_kはk番目の出力データのピクセル強度を表す。入力値と出力値の差を二乗することによって、ピクセル毎の強度差が正の値になり、それと同時により大きな値に変換される。損失の値が大きければ、モデルのパラメータは入力画像を再現するにあたって不適なものとなっており、値を大きく変更する必要があるといえる。逆に損失の値が小さければ、モデルのパラメータは入力画像を再現するにあたって的確な値に近づいていることになり、より細かなパラメータ調整が必要になる。損失の値の大小によってパラメータをどれだけどのように調整するのかを決定するのが 2.3.2 項で述べる最適化手法である。

2.4.2 最適化手法

学習時に機械学習モデルへ入力を行う際、一つの入力に対して一つの出力を損失関数の計算結果として得る。このことより、機械学習モデルは無数のパラメータを持つ 1 つの大きな関数であると言うことができ、損失が最小となるパラメータを探索する操作は、損失関数の最小値を求めることに等しい。であれば、単純な微分計算により最小値を求められるように思えるが、一般に損失関数はパラメータが多く非常に複雑な形をしており、解析的に最小値を求めることはできない。そこで「勾配降下法」という手法で損失関数が最小となるパラメータを探索する。ある一つのパラメータwを調整する際に勾配降下法により行われる計算を式 2.20 に示す。

$$w_{n+1} = w_n - \eta \left. \frac{\partial L(w)}{\partial w} \right|_{w=w_n} \tag{2.20}$$

w_n ：n回目に調整されたパラメータ
μ ：学習率
$L(w)$ ：損失関数
$\frac{\partial L(w)}{\partial w}$ ：パラメータwに対する損失関数の勾配

μは学習率と呼ばれ、一度に更新されるパラメータの幅を調整する値である。この値は通常任意に設定され、大きすぎても小さすぎても学習がうまく進まない。本研究で行なった実験では一律で1.0×10^{-6}とした。この操作を何度も繰り返すことによって、損失関数の最小化を行う。ただし、勾配降下方によって求められる最小値は、実際には最小値ではなく最小値に近い極小値で収束してしまうことがほとんどである。しかし、Choromanska et al. 2014 により損失関数の値が最小値ではない極小値であったとしても実用上十分な性能を発揮することが知られている[11]。そして「最適化手法」とは、この勾配降下法をベースにして、より効率的に損失を最小化するための手法の総称である。現在様々な最適化手法が考案されており、本研究では Kingma & Ba 2014 により考案された Adam と呼ばれる最適化手法を用いた[13]。

2.5　K-S 検定(コルモゴロフ-スミルノフ検定)

本研究ではバブル領域と一般的な領域の分子雲の特性の違いを、それぞれのデータを CAE に入力した際に得られた潜在変数を比較することにより検証した。具体的な検証方法については 3.2 節で述べる。潜在変数を比較する方法として今回用いたのが、コルゴモロフ-スミルノフ検定、通称 K-S 検定と呼ばれる手法である。この検定は、二つのヒストグラムが統計的に異なるのかどうかを評価するものであり、有意水準と呼ばれる閾値を設け、二つのヒストグラムが一致する場合と異なる場合を判断する。図 2.10 に、K-S 検定により一致されると判断される場合と異なると判断される場合の例を示す。

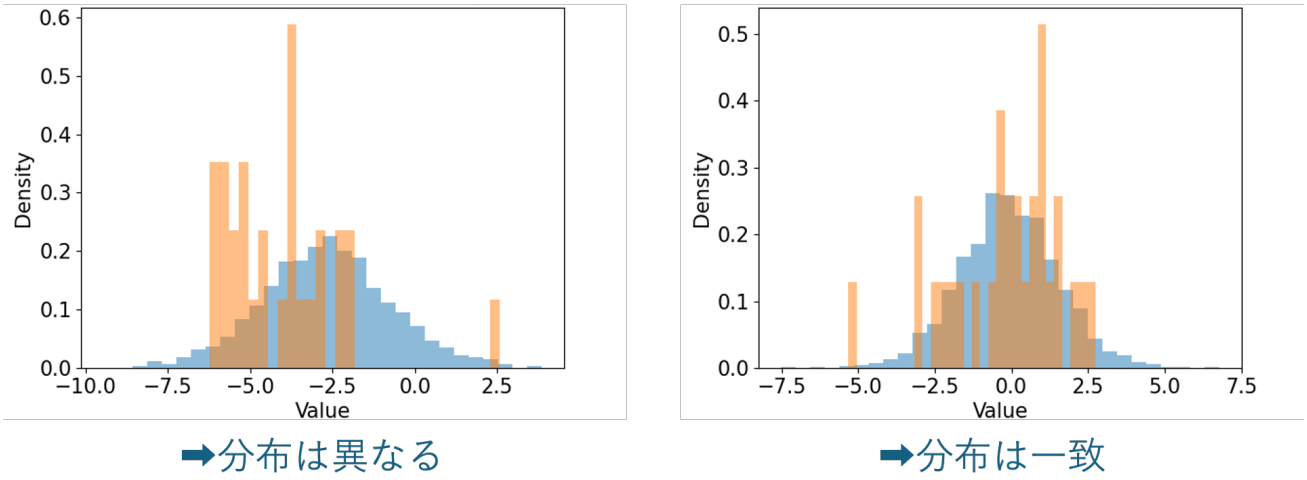

図 2.10: K-S 検定によって評価されたヒストグラムの例

第 3 章　実験と結果

本節では、実験の際に行なった学習データの作成、実験の実行、結果について順に述べる。なお、学習データの作成と CAE の実装、実験結果の検証はすべてプログラム言語 Python を用いて行なった。

3.1　学習データの作成

学習データの作成は野辺山 45m 電波望遠鏡で観測された Cygnus-X 領域 ^{12}CO(J=1-0)輝線のデータを使用した。この領域は星形成が活発に行われている領域として有名であり、先行研究によって spitzer bubble が検出されている。そのため、spitzer bubble 周辺の分子雲と一般的な分子雲の空間分布と速度分布の特徴量を比較する上で最適な領域であると言える。

学習データは以下の手順で作成した。
[1] 使用するチャンネルの選択（3.1.1）
[2] エミッションに対するマスク処理（3.1.2）
[3] データの切り抜き（3.1.3）
[4] 欠損値の含まれるデータを削除（3.1.4）
[5] 12 層になるように速度軸方向へ積分（3.1.5）
[6] ガウシアンフィルター処理（3.1.6）
[7] データの正規化（3.1.7）
[8] 完成したデータを目視で確認後エラーデータの削除（3.1.8）
[9] Data Augmentation (3.1.9)
[10] データの分割
本節では、データ作成の際に行った処理の詳細について一つずつ述べていく。

3.1.1　使用するチャンネルの指定

天文データの観測は、観測天体の静止周波数を基準に、ドップラーシフトにより変化した周波数域も含めて観測を行い、観測されたデータの周波数軸を速度軸方向へ変換し出力する。その際設定される周波数範囲は観測天体が十分入るように調整されるため、観測されたデータには分子雲がほとんど受かっていない周波数域も含まれる。図 3.1 は Cygnus-X データ全域に対し、チャンネルごとに強度の平均値を計算してプロットしたものである。天文データでは一般的に速度分解能毎に区切られた速度軸に番号を振ったものをチャンネルと呼ぶ。図 3.1 からわかる通り、中央付近のチャンネルにはよく強度が出ているが、両端のチャンネルにはほとんどノイズしか受かっていないことが分かる。本実験では観測データの 121~240 チャンネルを使用した。

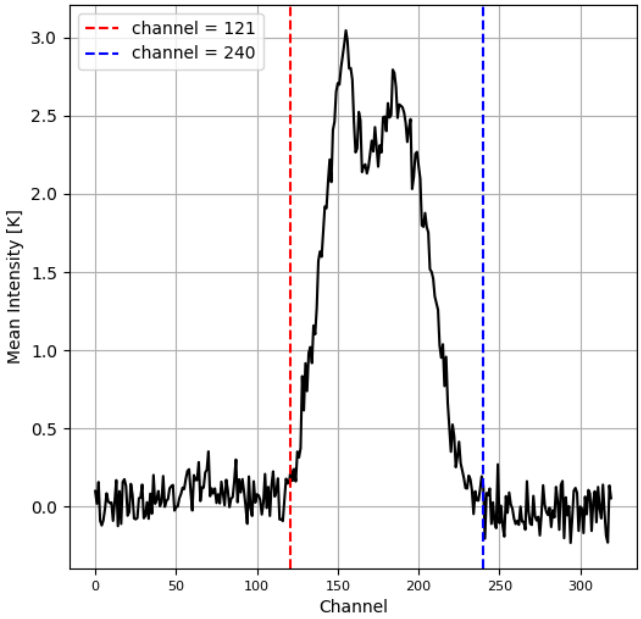

図 3.1: 野辺山 45m 電波望遠鏡で観測された Cygnus-X 領域 ^{12}CO(J=1-0)データをチャンネル毎に平均値を計算しプロットしたグラフ

3.1.2 エミッションに対するマスク処理

図 3.1 の 0 から 120 チャンネル、241 から最後のチャンネルを確認すると分かるが、天文データには受信機により発生するシステム雑音や、大気による吸収と放射による雑音など、さまざまな要因でノイズが発生する。それは天体が受かっているチャンネルでも例外ではないため、データに対してこのノイズを処理する操作を行った。本実験で行った処理方法は、RMS(ニ乗平均平方根)をベースに閾値を設定しマスキングを行う手法である。処理の具体的な方法は以下の通り。
[1] 0~120 チャンネルの全ピクセルの強度を 2 乗する
[2] 1pixel 毎に速度軸方向の平均値を計算
[3] 計算した 1pixel 毎の平均値の平方根を計算
[4] [3]で計算した 1pixel 毎のニ乗平均平方根未満の値を持つピクセルの値を全て 0 にする

この操作によって、ピクセル毎のノイズパターンに沿ったノイズ処理を行うことができる。図 3.2 はエミッションに対するマスク処理を行う前と行った後の比較画像の例であり、分子雲の構造がより鮮明になっていることが分かる。

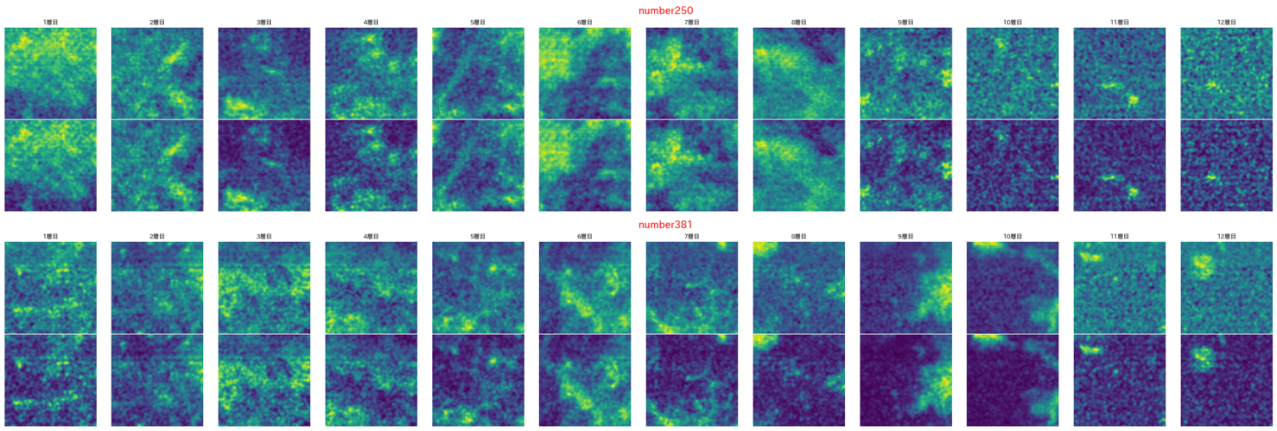

図 3.2: 3.1.1~3.1.8 の処理を全て行う際、マスク処理を入れた場合と入れなかった場合のデータを一部抜粋。二組の比較画像はともに上がマスク処理を行わなかった場合であり、下がマスク処理を行った場合である。

3.1.3　データの切り取り

CAE にデータを学習させる際に、一度に全領域から詳細な特徴量を抽出できる方法があればそれが最適ではあるが、今回のような大規模なデータをモデルに学習させることは、計算リソースが膨大になり過ぎてしまい、計算機の性能面で現実的ではない点や、全領域をそのまま学習データにしてしまうと分子雲の詳細構造が広域データの中で埋もれてしまい、局所的な構造を学習できないなどの問題がある。機械学習のデータ作成でそれらの問題点を解決する方法として、全体を一定の間隔で切り取り学習データを作成するというものがある。本実験では、116×116pixel の大きさで、全領域から学習データを切り取っていった。その際、116pixel で分割した領域をただ切り抜いていくだけでは、区間の間にある構造を分割してしまい、重要な特徴を破壊してしまう可能性がある。それを避けるため、今回は図 3.3 のように切り取りピクセル数である 116pix の 1/4 の大きさである 29pixel だけスライドしてデータの切り取りを行った。切り取り後のデータの形状は、速度軸方向も含めて(116, 116, 120)であり、データ数は 4233 個となった。

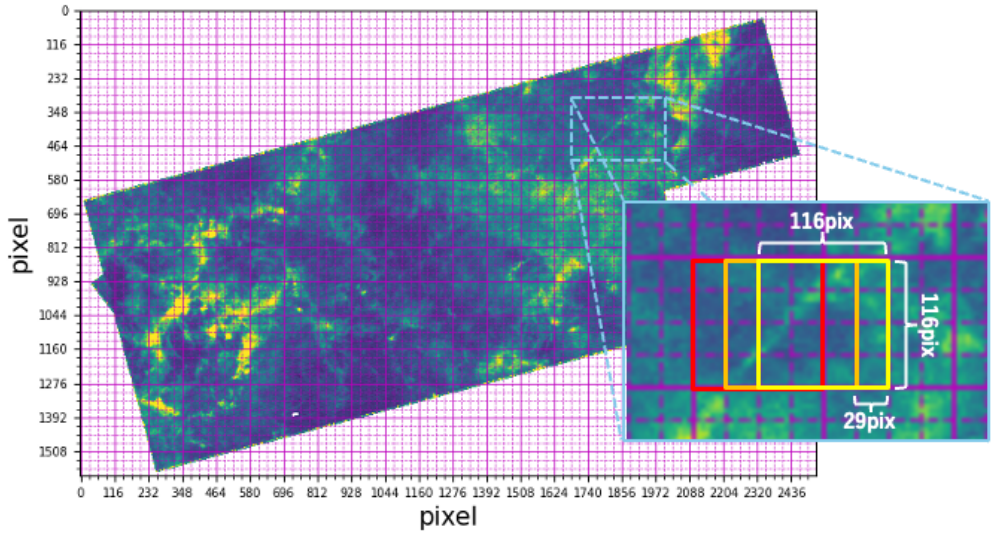

図 3.3: Cygnus-X 全域から 116x116pix を 116 の 1/4 である 29pix ずつ切り取っていく際の模式図

3.1.4　欠損値の含まれるデータを削除

3.1.3 で切り取りを行ったデータの中には、図 3.2 からわかる通りデータの存在しない領域が多く含まれている。それらの削除を行った結果、データ数は 1976 個となった。

3.1.5　12 層になるように速度軸方向へ積分

ここまでの処理の結果、一つあたりのデータの形状は(高さ, 幅, 深さ)=(116, 116, 120)となっているが、この形状のまま CAE へ学習を行わせてしまうと、使用している計算機の性能の都合で学習を進めることができなかった。そのため本実験では速度軸方向に対して 10 層ずつ積分処理を行い、一つあたりの学習データの形状を(116, 116, 12)とした。図 3.4 は 10 層ずつ積分する際の模式図である。

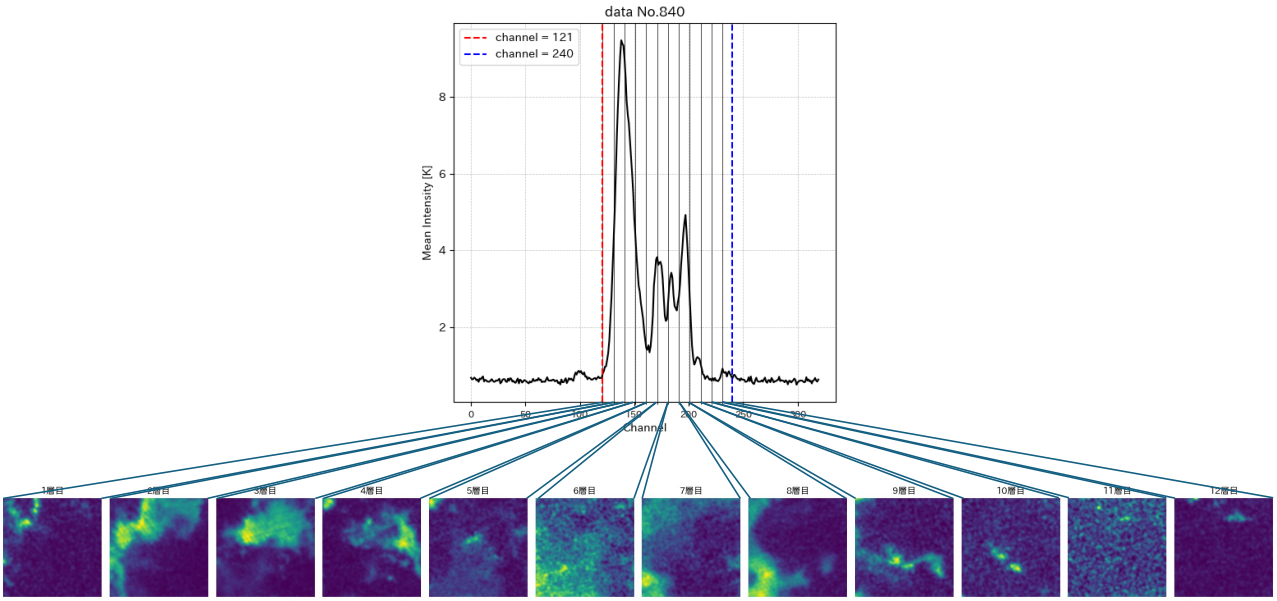

図 3.4: 速度軸方向へ 12 層に 10 チャンネルずつ積分処理を行った模式図

3.1.6　ガウシアンフィルター処理

ガウシアンフィルター処理は、FUGIN データの前処理として行ったものである。ガウシアンフィルターは平滑化フィルターと呼ばれるフィルターの 1 つであり、ピクセル強度を周辺のピクセル強度を加味した値に変更し、画像を滑らかにする効果がある。一般的にフィルター処理では、一定サイズのフルターを用意し、入力画像上の一部とフィルターの積和計算の結果を新たな値とする。そしてそのフィルターを一定間隔で動かすことによって入力画像全体にフィルター処理を施す。ガウシアンフィルターの場合は、この重み付けをガウス分布により決定する。ガウス分布は式 3.1 のように表され、注目する画素に近いピクセルほど重みが大きくなる。また本実験では図 3.4 に示す 5×5 のガウシアンフィルターを使用した。

$$f(x,y) = \frac{1}{2\pi\sigma^2}\exp\left(-\frac{x^2+y^2}{2\sigma^2}\right) \tag{3.1}$$

1/256	4/256	6/256	4/256	1/256
4/256	16/256	24/256	16/256	4/256
6/256	24/256	36/256	24/256	6/256
4/256	16/256	24/256	16/256	4/256
1/256	4/256	6/256	4/256	1/256

図 3.4: 本実験で使用したガウシアンフィルター

ガウシアンフィルター処理を行う前と後の比較画像として、FUGIN プロジェクトと呼ばれる野辺山 45m 電波望遠鏡を用いた天の川銀河の広域分子雲サーベイプロジェクトにより観測されたデータの一部を例として示す。

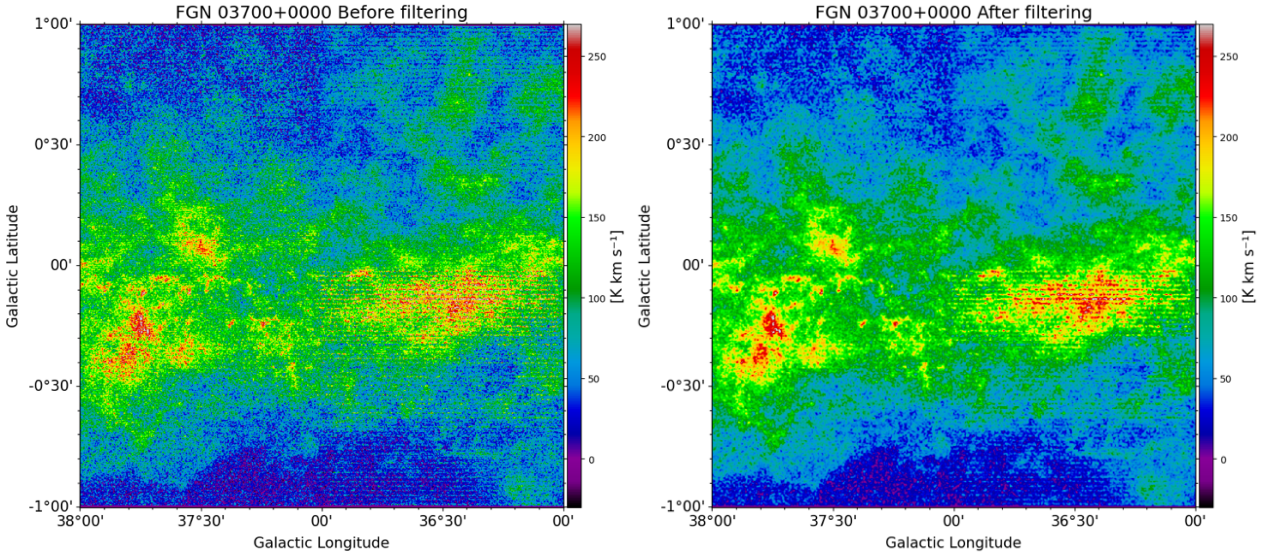

図 3.6: FUGIN プロジェクトにより観測された銀経36°~38°、銀緯-1°~1°の ^{12}CO(J=1-0)輝線の積分強度図に対し、ガウシアンフィルター処理を行わなかったもの(左)と行ったもの(右)の比較画像。ノイズが軽減され画像が滑らかになっていることが分かる

ガウシアンフィルター処理で積和計算を行う際、その性質上周囲 2 ピクセルにはガウシアンフィルターがかからない。学習データ内に条件が異なる箇所が含まれるのは、モデルがその箇所を重要な特徴であると認識してしまうなどの問題が起こる可能性があるため、本実験ではガウシアンフィルターがかからない周囲 2 ピクセルは削除した。その結果、一つあたりのデータの形状は
(高さ, 幅, 深さ)=(112, 112, 12)となった。

3.1.7　データの正規化

機械学習モデルに画像の学習を行わせる際に、モデル内での計算の高速化や安定性の向上のために、各画像の中で最大値が 1、最小値が 0 となるように正規化を行う処理がしばしば行われる。本実験内でも式 3.2 のように、それぞれのデータのi番目のピクセルx_iに対し、x_iからデータ内のピクセルの最小値x_{min}を引いた値を、最大値x_{max}からx_{min}を引いた値で割り、新たなピクセルの値x_i^{new}として格納する処理を行った。

$$x_i^{new} = \frac{x_i - x_{min}}{x_{max} - x_{min}} \tag{3.2}$$

3.1.8　完成したデータを目視で確認後エラーデータの削除

3.1.1~3.1.7 の処理により、(高さ，幅，深さ)=(112, 112, 12)の形状を持つ 1976 個のデータが完成した。しかし完成したデータの中には、欠損データが含まれなくとも正常ではないと思われるものが存在する。これは全領域の端から切り取られたデータに顕著に見られる。学習データにそのようなデータを使用することは学習を妨げる原因となるため、完成したデータを一つずつ目視で確認することによってエラーデータの確認、削除を行った。その結果、エラーデータの数は 30 個であり、データ数は計 1946 個となった。図 3.7 にエラーデータの一つを速度軸方向の 12 層を 1 層ずつ横に並べた画像を示す。また、完成した正常なデータの例を 5 つ図 3.8 に示す。

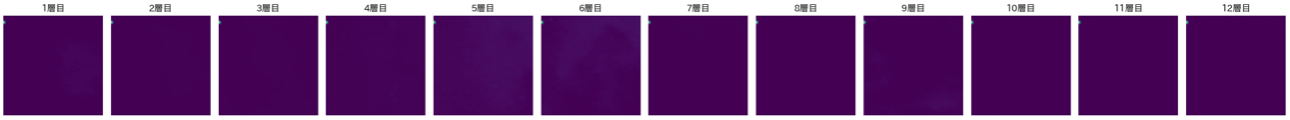

図 3.7: 完成したデータに含まれていたエラーデータの一例を速度方向の 12 層を 1 層ずつ横に並べた画像

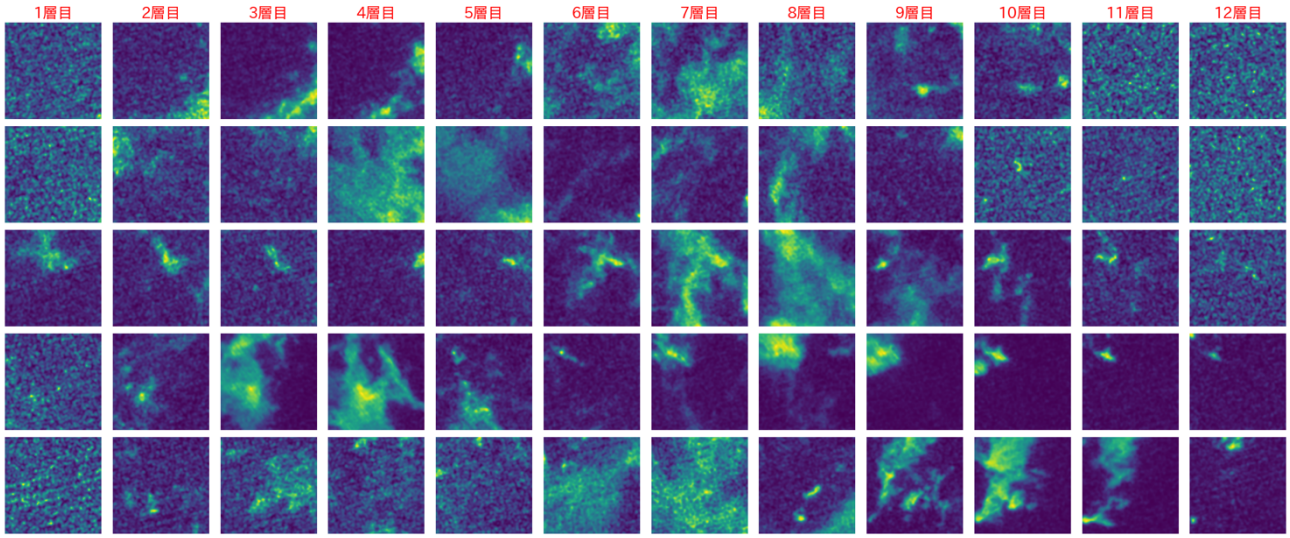

図 3.8: 完成したデータのうち例として 5 つを 1 層ずつ横に並べた画像

3.1.9 Data Augmentation

完成したデータの数は計 1946 個となったが、このデータ数で学習を行った際のモデルの精度は良いとは言えない結果となった。そこで Data Augmentation と呼ばれる手法を用いた。Data Augmentation とは、学習前の入力データに加工を施すことで学習データのバリエーションを増やす手法のことである。モデルからすると、学習の度に異なるパターンの学習データがやってくるため、局所的な特徴ではなく、本質的な特徴を捉えることが出来るようになる。本実験では学習データを上下反転させたものと 90°、180°、 270°回転させたデータを加え、データ数は計 11676 となった。

3.1.10 データの分割

膨大なパラメータを持つ機械学習モデルでは、特定のデータのみを使用し学習を行うと、いくらでも精度を上げていくことが可能である。しかしここで注意すべき重要なことが、学習データに対して精度良かったからといって、それ以外のデータでも精度が良いとは必ずしも言えないということである。機械学習（特にパラメータが膨大である深層学習の分野で）学習用データに対してのみ精度が良く、それ以外のデータに対する精度が良くない状態を「過学習」と呼ぶ。この過学習対策として簡単に実行できて最もわかりやすい操作としてあげられるのが、「訓練データと検証データの分割」である。本研究では過学習対策の一つとしてこの「訓練データと検証データの分割」を採用し、訓練データによって学習されたパラメータを使用し、検証データの精度を測ることによってモデルの性能を測り、検証データに対するモデルの精度の向上を目指して学習を行う。ここで言うモデルの精度とは、検証データに対する損失(2.3.1)の値の大小を指す。（損失の値は必ずしも精度を表すわけではないが、本研究では入力データと出力データの強度差が精度に直結するため、本研究では損失の値を指標に精度を測ることとした）。また、本研究では全学習データを訓練データと検証データに分割することに加え、テストデータにも分割した。このテストデータは、モデルの精度を実際に人間の目で確認する際に使用した（詳しくは 3.3 節にて後述）。テストデータもモデルの学習には使用していない。分割によって訓練データ、検証データ、テストデータの 3 つに分けられたそれぞれのデータ数を図 3.9 に記載する。

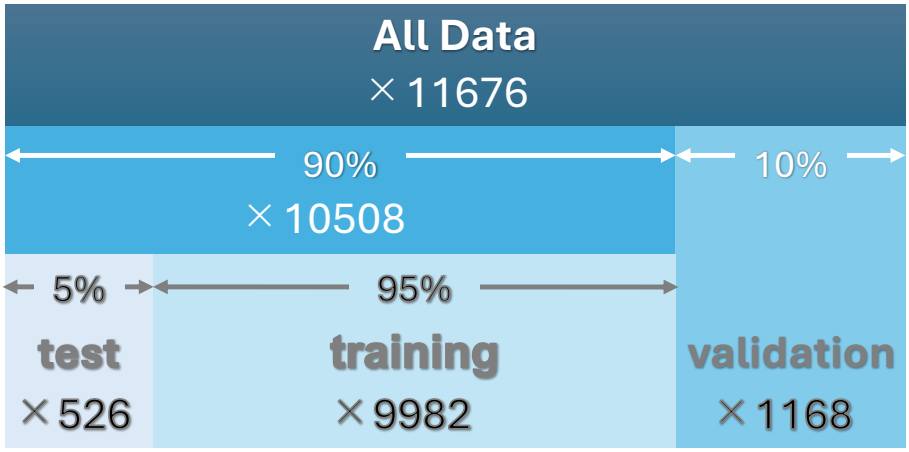

図 3.9: 分割によって分けられたそれぞれのデータ数
テストデータ ：526 個
訓練データ ：9982 個
検証データ ：1168 個 となった。

3.2　学習の実行

第 2 章で述べた CAE の学習を、3.1 で述べた学習データを用いて行なった。学習を行う際には、過学習対策の一種である Early Stopping（早期終了）という手法を用いた。また、入力データの特徴量の圧縮結果である潜在変数の数は、50、100、150、200、250、300、400、500、1000、2000 に設定し、それぞれ学習を行なった。

3.2.1　Early Stopping（早期終了）

3.1.10 で述べたように、機械学習モデルの学習は、学習データを訓練データと検証データに分けて行う。その際、一般的に訓練データに対する精度は上昇しても、検証用データへの精度はどこかで頭打ち（または低下）していく傾向がある。Early Stopping とは、それを防ぐための手法の一つである。具体的には、学習の際に損失の値が減少しない場合が連続で続いた際に、それ以上の精度の向上は難しいと判断し学習を終了する。その際、最終的なモデルのパラメータは、早期終了されるまでで一番損失関数の値が少なかったものが採用される。本研究では、早期終了の回数を 20 に設定し学習を行なった。

3.2.2　モデルのまとめ

本研究で使用した CAE の構造と関数及びパラメータを、改めてまとめたものが図 3.10 である。

Encoder

```
Conv3d(1, 64, kernel_size=(4, 4, 4), stride=(2, 2, 2), padding=(1, 1, 1)),  #output=(6, 56, 56)
ReLU(True),
Conv3d(64, 32, kernel_size=(4, 4, 4), stride=(2, 2, 2), padding=(1, 1, 1)),  #output=(3, 28, 28)
ReLU(True),
Flatten(),
Linear(32*3*28*28, latent)
```

Decoder

```
Linear(latent, 32*3*28*28),
Unflatten(1, (32, 3, 28, 28)),  # Unflatten back to (32, 5, 25, 25)
ConvTranspose3d(32, 64, kernel_size=(4, 4, 4), stride=(2, 2, 2), padding=(0, 1, 1)), #output=(8, 56, 56)
ReLU(True),
ConvTranspose3d(64, 32, kernel_size=(3, 3, 3), stride=(1, 2, 2), padding=(0, 1, 1)), #output=(10, 111, 111)
ReLU(True),
ConvTranspose3d(32, 1, kernel_size=(3, 2, 2), stride=(1, 1, 1), padding=(0, 0, 0)), #output=(12, 112, 112)
Sigmoid()  # to scale output between 0 and 1
```

損失関数	最適化手法	バッチサイズ	学習率	早期終了回数	データ形状
MSE	Adam	8	1.0×10^{-6}	20	$112 \times 112 \times 12$

図 3.10: 本研究で使用した CAE のモデル構造と諸パラメータを改めて記載。ここでバッチサイズとは、一度にモデルに入力するデータ数のことである。8 個のデータがモデルに入力され損失関数の値に応じてパラメータが修正、また新たに 8 個のデータが入力されるとい流れである。

この CAE モデルの構造と諸パラメータを使用して学習を行なった。

3.2.3　潜在変数毎の損失関数の推移

モデルの学習は潜在変数の値を 50、100、150、200、250、300、400、500、1000、2000 と変化させ、その値毎に行なった。これは本実験で入力する分子雲のデータ形状である(高さ，幅，深さ)=(112, 112, 12)から特徴を抽出するのに十分な潜在変数の数が不明であり、適切な潜在変数の数を探索するためである。バブル領域と一般的な領域の分子雲の特性を評価する際に、潜在変数の数はできるだけ少なく、かつ重要な情報を抽出できる数が望ましい。図 3.11 は学習の際に損失関数の値が推移していく様子である。学習はすべて、モデル精度が向上しない回数が 20 回を迎え Early Stopping により終了されるまで行なった。

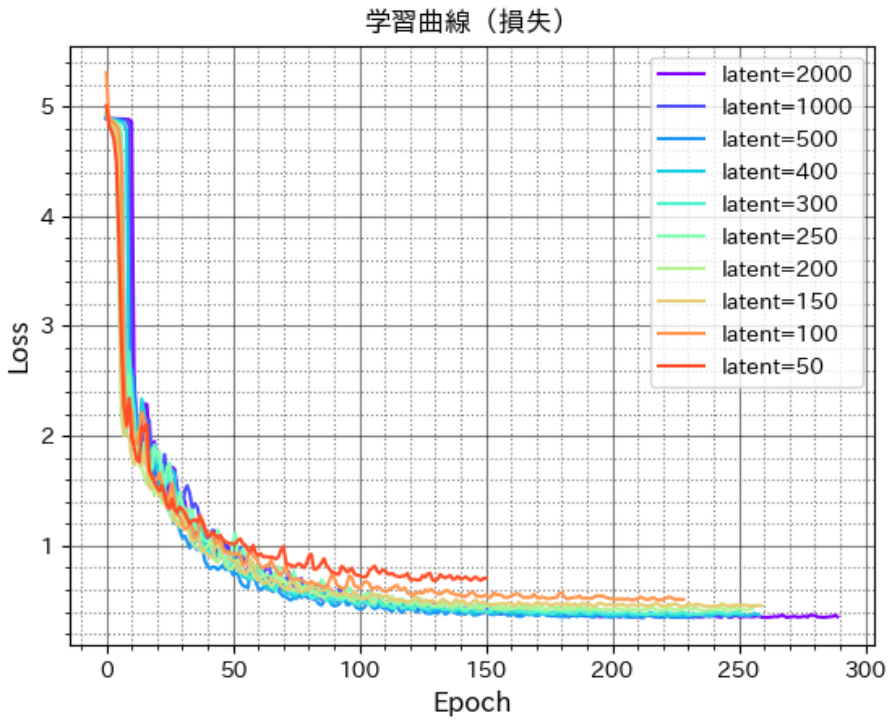

図 3.11: 潜在変数毎の学習が進んでいく様子。縦軸(Loss)は損失の値であり、横軸(Epoch)はモデルが訓練データセット全体を 1 回学習し終えたことを 1Epoch としている。機械学習では一般的にこの Epoch が学習の単位として使用される。

3.3　学習結果の確認

モデルの精度を視覚的に判断するために、学習によって得られたパラメータを使用した CAE に対してテストデータを入力し、出力結果を潜在変数毎に比較した。それに加え、潜在変数毎の最小損失値を比較することによって、分子雲の重要な特徴を抽出するうえで、潜在変数の値をどの程度まで減らすことができるのかどうかの判断を行なった。これは分子雲の重要な特徴のみを抽出するためと、第 4 章で行う抽出された特徴量が何を表しているのかを判断する際、潜在変数の数が多過ぎると議論が難しくなることが理由である。

3.3.1 潜在変数毎の再現画像の確認

潜在変数の数を 50、100、150、200、250、300、400、500、1000、2000 として学習を行い、得られたパラメータを使用した CAE に対してテストデータを入力し再現画像を作成した。結果の一例を図 3.12 に示す。

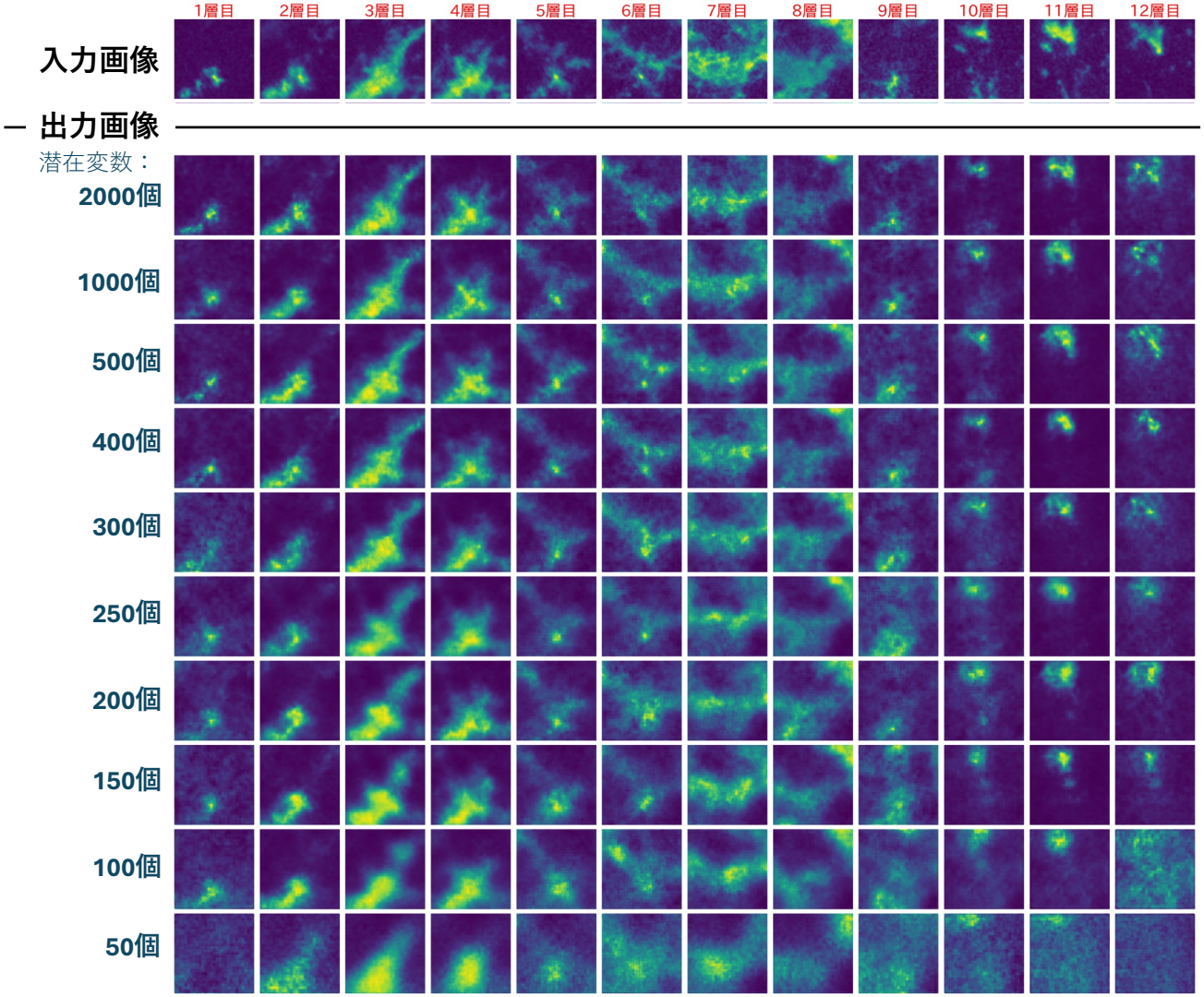

図 3.12: 入力画像と潜在変数毎の再現画像の比較例

出力画像を入力画像と比較していくと、潜在変数の数 2000~400 個までは入力画像をよく再現しており、精度にもそこまで差がないように思える。潜在変数の数が 300 以下になると、少しずつ再現画像の様子が崩れ始め、抽象的な画像になっている様子が分かる。潜在変数 100 個までは入力画像の再現がかろうじてできているように思えるが、50 個になると分子雲の特徴を再現しているとは言えない結果となっている。

3.3.2 潜在変数毎の最小損失値の確認

潜在変数毎の精度の違いを確認するために、潜在変数の数ごとの最小損失値をプロットしたものが図3.13である。

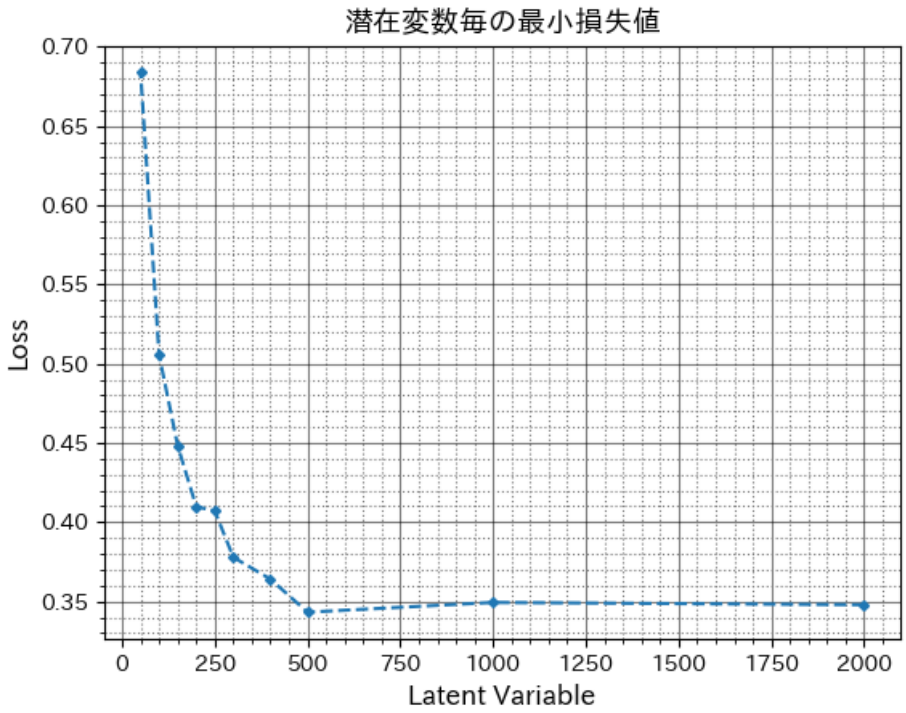

図 3.13: 潜在変数毎の最小損失値のプロット図
縦軸が損失値、横軸が潜在変数の数である。

図3.13のグラフを見ると、損失の値は潜在変数が500個のときに最も小さくなっている。500個よりも潜在変数の数を増やした場合は、逆に損失の値が微量ではあるが増加しているのがわかる。これはノイズなどの重要ではない細かな特徴量まで抽出してしまっている可能性が考えられる。

図3.13のグラフより、今回行った実験で一番精度が良いと考えられる潜在変数の数は500であり、図3.10の再現画像の比較から、潜在変数の数が100個まではかろうじて分子雲の特徴を再現できていると言える。このことから、第4章で行う抽出された潜在変数の値が何を意味しているのか考察には、先ほど述べた二つの潜在変数の値にその中間の300を加えた、100、300、500個で行う。

3.4　バブル領域と一般的な領域の潜在変数を比較

この章では、3.1~3.3 で学習を行なった CAE を使用して分子雲のデータから潜在変数を実際に取り出し、バブル領域と一般的な領域の分子雲の潜在変数を比較していく。その際学習データとはまた別に、spitzer bubble を中心において分子雲のデータを切り取ったデータを作成した。比較方法は、まず Cygnus-X 全領域から切り取ったデータを CAE に入力し潜在変数を抽出。その後同じようにバブル領域を切り取ったデータを CAE に入力し潜在変数を抽出。全領域から抽出された潜在変数とバブル領域から抽出された潜在変数の分布の違いを K-S 検定を用いて評価し、分布が異なると判断される潜在変数の数を数えた。また対照実験として、全領域からランダムにデータを、バブル領域のデータ数と同数選出し、バブル領域と同様の比較を全領域と行った。

3.4.1 バブル領域の分子雲データの作成

Cygnus-X 領域は大質量星の形成が活発に行われている領域として知られており、それに伴い spitzer bubble の検出も盛んに行われている。本研究では、Tharindu et al. 2019 によって行われた、市民参加型のリング検出プロジェクトである Milky Way Project により同定された spitzer bubble と、Nishimoto et al. 2025 によって同定された spitzer bubble の座標をもとにバブル領域の分子雲データの作成を行なった。図 3.14 は spitzer 宇宙望遠鏡により撮影された Cygnus-X 領域の赤外線画像に今回使用した野辺山 45m 望遠鏡で観測された Cygnus-X の分子雲データが存在する領域のコントアと、MWP と Nishimoto et al.2025 によって同定された spitzer bubble の位置を描画したものである。

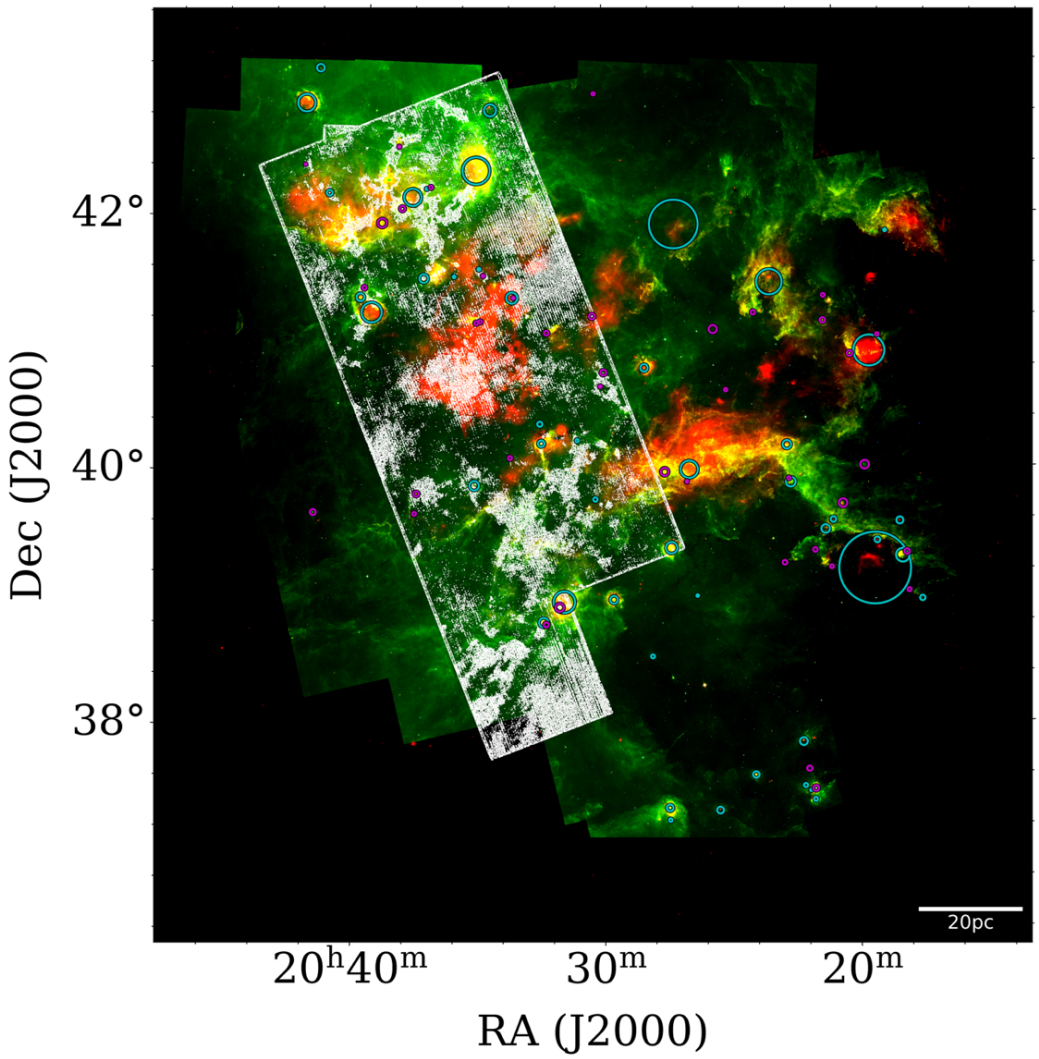

図 3.14: spitzer 宇宙望遠鏡によって撮影された赤外線画像に野辺山 45m で観測された Cygnus-X の ^{12}CO(J=0-1)データが存在する領域の積分強度図のコントアと、MWP と Nishimoto et al.によって同定された spitzer bubble の位置を描画した画像。コントアは 1, 5, 100, 125, 150 の強度で引いた。
シアンの円が MWP により同定されたバブルであり、マゼンダが Nishimoto et al. 2025 により同定されたバブルである。

バブル領域の分子雲データ作成には、spitzer bubble の中心座標を起点に 116×116pixel の領域を切り取った。その際 spitzer bubble の位置によっては、切り取りピクセルがデータ内におさまらないものも存在する。それらの削除を行なった結果、バブル領域の切り取りデータは全部で 29 個となった。切り取り以外のデータ処理は全て 3.1 と同様の操作を行なった。図 3.15 にバブル領域のデータの例を示す。

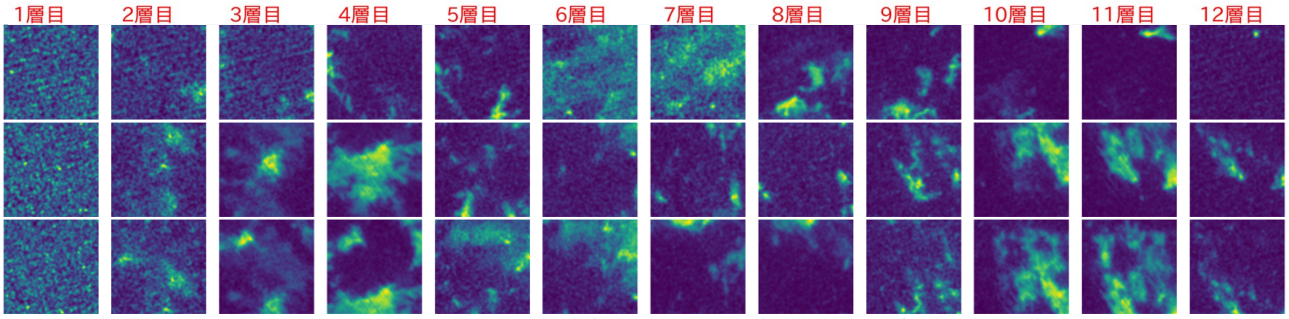

図 3.15: バブル領域のデータを 1 層ずつ並べたデータ例

3.4.2 潜在変数の比較方法と結果

バブル領域と一般的な領域の分子雲の特性の比較方法の詳細を述べる。全領域から抽出された潜在変数とバブル領域から抽出された潜在変数の分布の違いを K-S 検定を用いて評価し、分布が異なると判断される潜在変数の数を数え、対照実験として、全領域からランダムにデータをバブル領域のデータ数と同数選出し、バブル領域と同様の比較を全領域と行った。この二つで得られた全領域とは異なると判断された潜在変数の数を比べることで、仮に一般的な領域とバブル領域の分子雲に特性の違いがないのであれば、全領域と異なると判断される潜在変数の数は同程度になるはずであり、二つの特性に違いがあるのなら、全領域と異なると判断される潜在変数の数はバブル領域の方が多くなるはずである。

潜在変数の分布の違いは、2.5 節で述べた K-S 検定を用いて次のように行なった。まずバブル領域のデータ 29 個を CAE に入力、潜在変数の抽出を行う。潜在変数を 100 個に設定している場合では、潜在変数毎に 29 個の値が出力され、100×29 個の潜在変数の値が出力されることになる。次に潜在変数 1 つ毎にヒストグラムを作成すると、29 個分の値を持つヒストグラムが 100 個できる。これと同様の操作を Data Augmentation する前の全領域のデータにも行うと、全領域のデータ数は 1946 個なので 100×1946 個の潜在変数が出力され、1946 個分の値を持つヒストグラムが 100 個できることになる。これらのヒストグラムを潜在変数毎に K-S 検定により比較していき、分布が異なる潜在変数の数をカウントした。図 3.16 はヒストグラムの作成と K-S 検定の模式図であり、図 3.17 はバブル領域とランダムな領域の潜在変数を比較する際の模式図である。

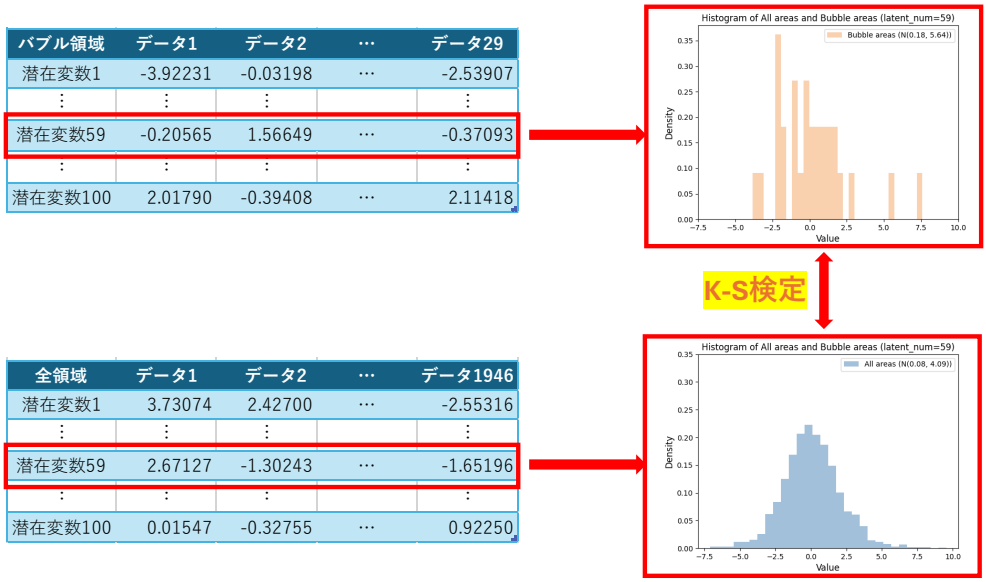

図 3.16: ヒストグラムの作成と K-S 検定の模式図
59 個目のバブル領域と全領域から抽出した潜在変数のヒストグラムを示す。
また、図中のヒストグラムは共に面積が 1 になるように正規化されている。

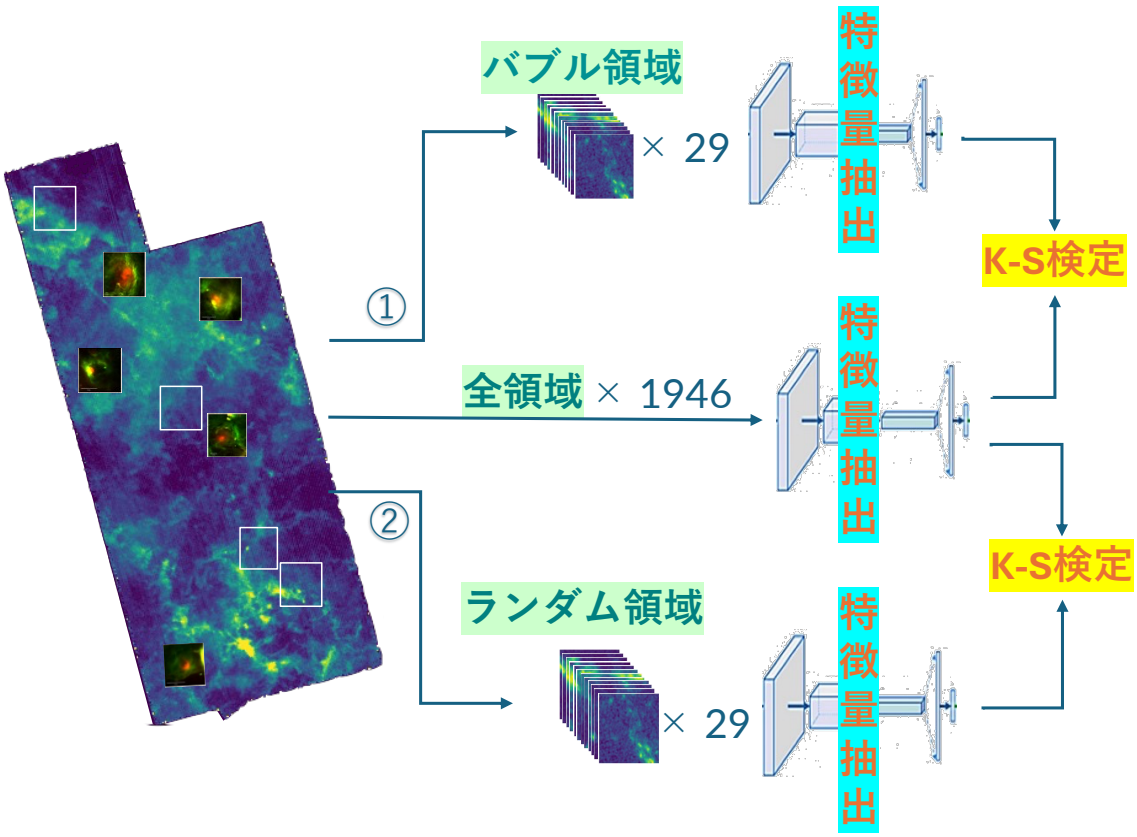

図 3.17: バブル領域とランダムな領域の潜在変数を比較する際の模式図。
バブル領域 29 個のデータから潜在変数を抽出し、全領域の潜在変数とヒストグラムを図 3.13 のように K-S 検定によって比較。同様の操作をランダムな領域から選出した 29 で行った。

結果

比較結果は図 3.18 のようになった。

潜在変数の数	2000	1000	500	400	300	250	200	150	100	50
異なる潜在変数の数(バブル領域)	122	48	22	28	25	23	22	12	12	8
異なる潜在変数の数(ランダム領域)	17.6	7.1	5.8	3.5	2.8	2.7	1.5	1.2	0.5	0.1

図 3.18: 潜在変数毎にバブル領域とランダムな領域を K-S 検定によって比較した結果、異なると判断された潜在変数の数を表にしたもの。なお、ランダムな領域は 10 回の平均値を記載している。ランダム領域と全領域を比較した全 10 回の結果は付録 3.1 にて記載する。

図 3.18 の結果から分かる通り、どの潜在変数の値でもランダムに選出した領域と比較してバブル領域の方が、全領域と異なると判断された潜在変数の数は明らかに多いという結果になった。これより、イントロダクションで示した spitzer bubble が存在する領域の分子雲は、大質量星からの影響を受け、他の領域とは異なる空間、速度分布の特性をもっているという説は正しいという結果となった。

第 4 章　議論と考察

第 3 章の実験結果より、spitzer bubble が存在する領域は他の領域とは異なる空間、速度分布の特性を持っているということが明らかになった。この章では、異なると判断された潜在変数が何を表しているのかを考察する。検証には、損失の値から一番精度が良いと判断した、潜在ヘンスが 500 の場合と、潜在変数は数が少なければ少ないほど重要な特徴のみを抽出できている可能性が高いという観点から 100 個の場合、そしてその二つの間である 300 個の場合で行なった。潜在変数が 500、300、100 の時の CAE の学習パラメータを使用し、それぞれにバブル領域とバブルが存在しない領域から抽出した潜在変数を、異なると判断された潜在変数以外の値を 0 にして CAE の Decoder に入力を行い、再現画像の比較を行なった。この操作により、バブル領域とその他の領域で異なる特徴量のみが反映された出力画像を比較することができる。図 4.1 がバブル領域から選択したデータ例であり、図 4.2 が spitzer bubble の無い領域から選択したデータ例である。

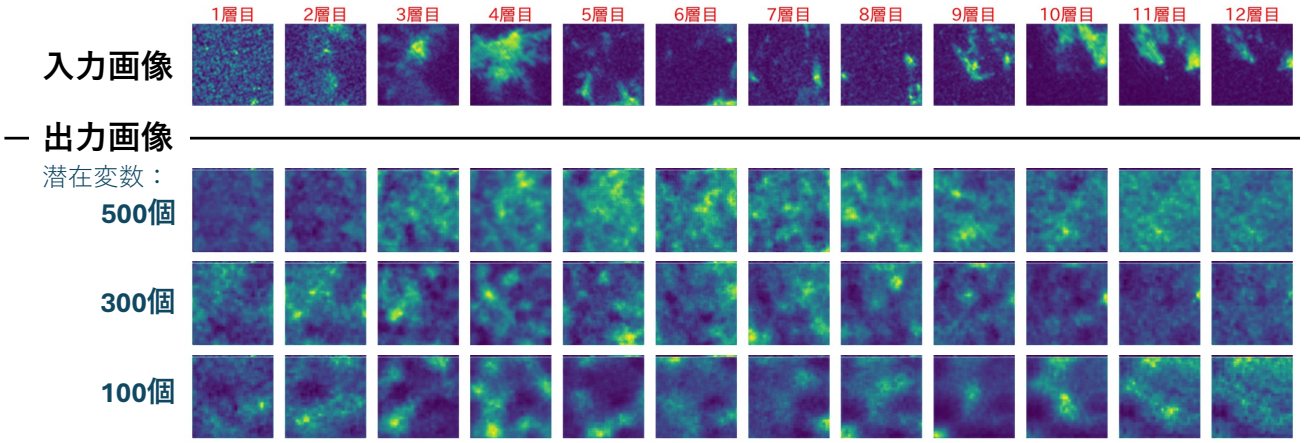

図 4.1: バブル領域から選択したデータ例。1 番上が入力画像であり、下 3 つが潜在変数毎の出力画像である。

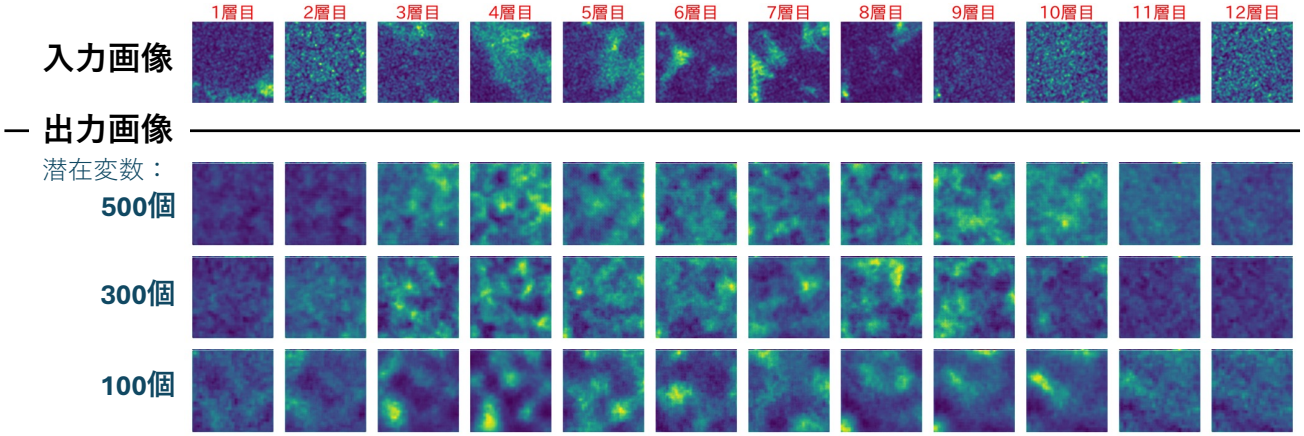

図 4.2: spitzer bubble の無い領域から選択したデータ例。1 番上が入力画像であり、下 3 つが潜在変数毎の出力画像である。

二つの出力結果を視覚的に比較して、違いがあるのかどうか判断することを試みたが、有意な違いがあるとは言えなかった。ここでは一例のみを挙げたが、他のデータを使用し同様な比較を行なった際も結果は似たようなものであった。ここまでの実験では、CAE によって抽出された特徴量に対して、それが何を表しているのかを人間が判断することは難しいという結論となった。これは人間が、目に入ったものに対してそれが一体何なのかを判断することはできても、具体的な理由を述べることが難しいことと同義であると考える。カバンを例にあげると、カバンには手提げカバンやリュック、ショルダーバッグなど様々なものが存在し、その中でもさらに大きさの大小や開け口の種類、持ち手の位置など、異なる要素が多数存在する。にもかかわらず人間は高確率でそれらが全て「カバン」であることを認識できる。それは経験則に基づく「なんとなく」の感覚であり、そこに意味を持たせるのは難しい。その経験則をパラメータとして高速に学習できるものが機械学習であると考えるなら、今回の CAE によって抽出された特徴量に対して名前付けができないことにも納得できるのではないかと考える。ただし、バブル領域特有の特徴が判明すれば、分子雲のデータ解析分野で大いに役立つと考えられるため今後の研究の課題である。

第 5 章　まとめと今後

5.1 まとめ

本研究では、spitzer bubble が存在する領域の分子雲が一般的な領域の分子雲と異なる空間、速度分布の特性をもっているのかどうかを、CAE の画像再構築の際に生じる、入力データの特徴量が圧縮された潜在変数を用いて検証を行なった。主な結果を以下にまとめる。

・CAE に対して作成した分子雲のデータを学習させることにより、3 次元データから特徴量を任意の数の潜在変数に圧縮できるモデルが作成できた。

・バブル領域とランダムな領域の分子雲の潜在変数の分布をそれぞれ全領域から抽出した潜在変数と K-S 検定を用いて比較した結果、全領域とは異なる分布を持つ潜在変数の数はバブル領域の方が明らかに多いという結果となった。これより、バブル領域の分子雲は一般的な領域の分子雲とは異なる空間、速度の特性を持っていることが明らかになった。

・バブル領域の全領域と異なる分布をもつと判断された潜在変数以外の値をゼロにし、CAE の Decoder から画像を再現し、同様に Decoder から出力されたバブルを含まない画像と比較することによって異なると判断された潜在変数が何を表しているのを考察する実験を行なったが、有意な違いを判断するには至らなかった。

5.2 今後

本研究で、バブル領域は一般的な領域の分子雲と比較して異なる特性を持っていることがわかった。これは裏を返せば、モデルに入力した分子雲の潜在変数が一般的な潜在変数の値と異なるものが多かった場合に、その場所は大質量や、あるいは他の何かしらから影響を受けている領域である可能性が高いと言える。今後の研究では、今回作成した機械学習モデルが分子雲のデータから一般的な領域とは異なる領域を検出できる可能性を追求していく。また、モデルが抽出した特徴量が何を表しているのかを明らかにすることも今後の課題である。

謝辞

本研究を進めるにあたり、様々な方にお世話になりました。

指導教員である大西利和教授には、度々指針となるアドバイスを頂きました。大学院試験の願書に記載する研究テーマに関する相談に乗っていただいた際には自分が進むべき道を示してくださりました。また、卒業発表練習の際には私が自身の研究に対してしていた勘違いを指摘していただき、さらにそこから発展した今後の研究の方向性を示して下さりました。大西先生のご助言があったからこそ、自分が進むべき方向を定めることができました。

小川英夫客員教授には、電波天文学の観測に関する基礎知識を、ゼミを通して教えていただきました。そこで学んだ知識は、野辺山 45m 電波望遠鏡で観測を行なった際に、望遠鏡内部で行われていることを理解するうえで非常に役に立ちました。また、たまに小川先生から頼まれる、荷物の梱包や機材の組み立てなどといったゲリライベントは、良いガス抜きになっていました。

村岡和幸准教授には発表形式のゼミを実施していただき、天文学の知識を主体的に学んでいく姿勢を身に付けさせていただきました。ゼミを通して資料の作成や発表を行なっていたおかげで、その後の定例会や卒業発表ではスムーズに準備を進めることができ、本番を迎えることができました。

九州大学大学院理学府徳田一起特任助教には私が現在行なっている研究テーマに繋いでくださり、その後の指導もしていただきました。研究がなかなか思うように進まず、進捗報告が滞る場面も度々ありましたが、それでも変わらずご助言と今後の方針を示してくださいました。

同研究室の研究員の方や先輩方にも大変お世話になりました。

博士課程後期 2 年の西本さんには、この研究を進めるにあたって最もお世話になりました。天文学もプログラミングもまったくの初心者で、なかなか要領を得ない自分にその両輪の知識を懇切丁寧に教えてくださいました。天文の知識や機械学習の手法、卒業発表などの資料の英語のチェックまで、西本さんに頼りすぎていた感が否めませんが、これからも何卒よろしくお願い致します。

博士課程後期 2 年の小西亜郁さん、博士課程前期 2 年の國年悠里さん、東野康助さん、博士課程前期 1 年の安達大揮さんには、分野は少し異なりますが同じサイエンスグループで共通する部分も多く、データ解析などで行き詰まった際には大変お世話になりました。また、雑談を通して研究や健康面に対する向き合い方への学びと気付きを与えてくださいました。サイエンスグループ以外の先輩方にも、研究室配属後から気さくに話しかけて下さり、また様々なご助言をいただきました。

同期の岡本結人くん、角越仰くん、宮崎正成くん、山下晃矢くん、山本美咲さんとは、分野は皆バラバラでしたが、だからこそ自分達の研究内容について話し合うときは本当におもしろかったです。一緒にご飯を食べたり、個人的に遊びに行ったり、この一年間楽しく過ごすことができたのは同期のおかげです。来年度からもよろしくおねがいします。

最後に、大学生活で一人暮らしが始まり会える機会が少なくなった後も、これまでと変わらず自分のことを考え、支え続けてくれた両親に最大の感謝の意を示します。

参考文献

[1] Wolfire, M. G. and Cassinelli, J. P., “Conditions for the Formation of Massive Stars,” The Astrophysical Journal 319, 850 (Aug. 1987).

[2] Elmegreen, B. G and Lada, C. J., “Sequential formation of subgroups in OB associations,” Astrophysical Journal, 214, 725 (Jun 1977)

[3] Dale, J. E., Haworth, T. J., and Bressert, E., “The dangers of being trigger-happy,” 450, 1199–1211 (June 2015).

[4] Fukui, Y., Habe, A., Inoue, T., Enokiya, R., and Tachihara, K., “Cloud-cloud collisions and triggered star formation,” Publications of the Astronomical Society of Japan 73, S1– S34 (Jan. 2021).

[5] Benjamin, R. A., Churchwell, E., Babler, B. L., Bania, T. M., Clemens, D. P., Cohen, M., Dickey, J. M., Indebetouw, R., Jackson, J. M., Kobulnicky, H. A., Lazarian, A., Marston, A. P., Mathis, J. S., Meade, M. R., Seager, S., Stolovy, S. R., Watson, C., Whitney, B. A., Wolff, M. J., and Wolfire, M. G., “GLIMPSE. I. An SIRTF Legacy Project to Map the Inner Galaxy,” Publications of the Astronomical Society of the Pacific 115, 953–964 (Aug. 2003).

[6] Carey, S. J., Noriega-Crespo, A., Mizuno, D. R., Shenoy, S., Paladini, R., Kraemer, K. E., Price, S. D., Flagey, N., Ryan, E., Ingalls, J. G., Kuchar, T. A., Pinheiro Gon¸calves, D., Indebetouw, R., Billot, N., Marleau, F. R., Padgett, D. L., Rebull, L. M., Bressert, E., Ali, B., Molinari, S., Martin, P. G., Berriman, G. B., Boulanger, F., Latter, W. B., MivilleDeschenes, M. A., Shipman, R., and Testi, L., “MIPSGAL: A Survey of the Inner Galactic Plane at 24 and 70 μm,” Publications of the Astronomical Society of the Pacific 121, 76 (Jan. 2009).

[7] Churchwell, E., Watson, D. F., Povich, M. S., Taylor, M. G., Babler, B. L., Meade, M. R., Benjamin, R. A., Indebetouw, R., and Whitney, B. A., “The Bubbling Galactic Disk. II. The Inner 20° ,” The Astrophysical Journal 670, 428–441 (Nov. 2007).

[8] Yamaguchi, T. 2024, "教師なし機械学習を用いた分子雲銀河サーベイデータの解析," MSc thesis, Nagoya Univ.

[9] Kendrew, S., Simpson, R., Bressert, E., Povich, M. S., Sherman, R., Lintott, C. J., Robitaille, T. P., Schawinski, K., and Wolf-Chase, G., “The Milky Way Project: A Statistical Study of Massive Star Formation Associated with Infrared Bubbles,” The Astrophysical Journal 755, 71 (Aug. 2012).

[10] Nishimoto, S., Toshikazu O., Atsushi, N., Shinji F., Yasutomo, K., Shuyo, N., Kazuki T., Yoshito, S., Hiroyuki, K., Yusuke, M., Tsuyoshi, I., and Atsushi, M. I., “Infrared Bubble Recognition in the Milky Way and Beyond Using Deep Learning”, Publications of the Astronomical Society of Japan doi: 10.1093/pasj/psaf008

[11] Takekoshi, T., Fujita, S., Nishimura A., Taniguchi, K., Yamahishi, M., Matsuo, M., Ohashi, S., Tokuda

K., and Minamidani, T., “Nobeyama 45 m Cygnus-X CO Survey. II. Physical Properties of C18O Clumps”, The Astrophysical Journal 235, 9 (Mar. 2018)

[12] Choromanska, A., Henaff, M., Mathieu, M., Ben A, G., and LeCun, Y., “The Loss Surfaces of Multilayer Networks” arXive-prints, arXiv:1412.0233

[13] Kingma. D. P., and Ba, J., “Adam: A Method for Stochastic Optimization” arXive-prints, arXiv:1412.6980

福井康雄、犬塚修一郎、大利和、中井直正、舞原俊憲、水野亮 編，シリーズ現代の天文学（第 6 巻）星間物質と星形成，日本評論社

中井直正、坪井昌人、福井康雄 編，シリーズ現代の天文学（第 16 巻）宇宙の観測 II，日本評論社

斉藤康毅，ゼロから作る Deep Learning —Python で学ぶディープラーニングの理論と実装，オライリー・ジャパン

赤石雅典，最短コースでわかる PyTorch &深層学習プログラミング，日経 BP

付録

3.1 ランダムに選出した領域と全領域の潜在変数の分布を比較した際に行なった 10 回の実験結果。

回数	2000	1000	500	400	300	250	200	150	100	50
1	25	9	3	2	5	1	1	0	0	0
2	35	7	2	2	2	0	1	1	0	0
3	12	12	4	0	6	2	1	0	0	0
4	22	5	7	7	1	2	3	0	2	0
5	17	10	3	7	5	1	3	0	0	0
6	23	4	14	4	1	5	1	1	1	1
7	11	7	1	5	0	4	1	6	1	0
8	11	5	13	4	2	1	1	1	0	0
9	10	4	6	0	4	1	1	1	1	0
10	10	8	5	4	2	10	2	2	0	0
平均	**17.6**	**7.1**	**5.8**	**3.5**	**2.8**	**2.7**	**1.5**	**1.2**	**0.5**	**0.1**