物体配置タスクにおける構造的知識を用いた衝突予測および視覚的説明生成

○松尾榛夏 †，畑中駿平 †，平川翼 ‡，山下隆義 ‡，藤吉弘亘 ‡，杉浦孔明 †
† 慶應義塾大学　　‡ 中部大学

1. はじめに

要支援者が増加する現代社会において，介助従事者の不足が社会問題になっている．この人手不足の解決策の一つとして，家庭用の生活支援ロボットが有望視されている．生活支援ロボットにとって物体配置タスクは基本的動作の一つであり，高い安全性が求められる．

本研究では，生活支援ロボットが物体配置を行う際の，物体-ロボット-環境間の衝突に関する衝突予測タスクを扱う．ここでは軽微な接触の連鎖から物体の転倒や落下などが生じる危険な衝突も含む．例えば，食器が置かれている机に物体を置く場合における，食器と物体との衝突や，ロボットのアームと食器との衝突の推論を扱う．環境内に複数の物体が存在する場合，物理的相互作用の連鎖を考慮し，衝突を予測することは難しい．現状では，2 次元環境であっても正確に物理的相互作用の連鎖の推論を行うことは困難である [1].

既存手法の Transformer PonNet [2] では配置場所の画像に関する構造的知識を用いていなかった．また，配置方策をしないことを前提としていた．ここで，構造的知識とは，配置場所に存在する障害物の幾何的特徴量・画像特徴量同士の関係を表す．

そこで本研究では，構造的知識も考慮して物体配置における衝突リスクを推定し，配置方策も行う手法を提案する．提案手法では，配置領域および対象物体の画像に加えて構造的知識も考慮することで，物体配置における衝突リスクを推定することができる．また，提案手法を用いることで，家庭用ロボットの物体配置における衝突リスクを事前にユーザに伝え，実際に物体を配置させるかどうかの判断を仰ぐことができる．さらに，提案手法は，配置方策を導入することで既存手法より安全な位置に配置することができる．

本研究の新規性は以下である．

- Transformer PonNet を拡張し，新たに構造的知識を扱うために，Structural Causal Encoder モジュールを導入する．
- [3] を元にしたモデルが予測した安全領域に基づく配置方策を導入する．

2. 関連研究

物体配置分野は生活支援ロボットにとって重要なタスクであり，これまでにも多くの研究がなされている．関連するトピックとして，物体姿勢推定 [4]，平面検出 [5]，動作計画 [6]，選択推定 [7] がある．また，[8]，[5]，[9] では，障害物のない領域の推定に視覚やマルチモーダル入力を用いている．Harada らは [6] において，凸性，密着性，安定性を考慮したモデルベースの物体配置計画を提案している．動作計画では，サンプリングベースの手法 [10, 11] や深層強化学習 [12] による物体配置動作計画の研究が多く行われている．Jiang らは [7] において，物体間の幾何学的関係や，配置における人間の意向など，複数の性質を扱うグラフィカルなモデルを提案している．また，物体配置タスクに対する衝突予測 [13] や指示理解 [9] の手法を提案している．[13] では，安全な物体配置を行うために，物体の衝突可能性を推定する PonNet を提案している．また，[2] では大きさが未知の対象物体についても扱えるように PonNet を改良し，かつ PonNet の Perception Branch モジュールに transformer を導入した Transformer PonNet を提案している．さらに，[3] では衝突確率の低い安全な領域の候補を可視化する手法を提案している．

図 1　対象タスクの例．左：実験環境．右：カメラ画像

3. 問題設定

本論文は，対象物体を指定した場所に配置する際の物体間の衝突予測タスクを対象とする．本タスクでは，対象物体と配置領域の RGBD 画像から作成した attention map に基づいて衝突確率が高い場合には衝突，低い場合には接触と予測することが望ましい．

図 1 に本対象タスクの代表例を示す．図 1 の状況において，配置領域に対象物体を置く際の衝突確率を予測する．

本研究では，以下の入出力を想定する．

- **入力**： 対象物体の RGBD 画像と，それを配置する領域の RGBD 画像
- **出力**： 衝突が起こる確率の予測値

本論文で使用する用語を以下に定義する．

- **配置領域**： 棚や机などのロボットが対象物体を置く場所
- **対象物体**： ペットボトルや缶などのロボットが把持する日常的な物体
- **障害物**： 配置領域にすでに置かれている物体
- **衝突**： 物体同士の相対速度が閾値を超えた危険な接触

物体配置を行なった際の衝突確率の予測が目的であるため，ロボットは対象物体を持った状態で配置領域の前にいることを前提とする．さらに，水の入ったコップのような物体は衝突に付随した別の危険性があるため本タスクの前提としない．

4. 手法

4.1 ネットワーク構造

提案手法のネットワーク構造を図 2 に示す．Feature Extractor および Target Embedder は [2] と同様の構

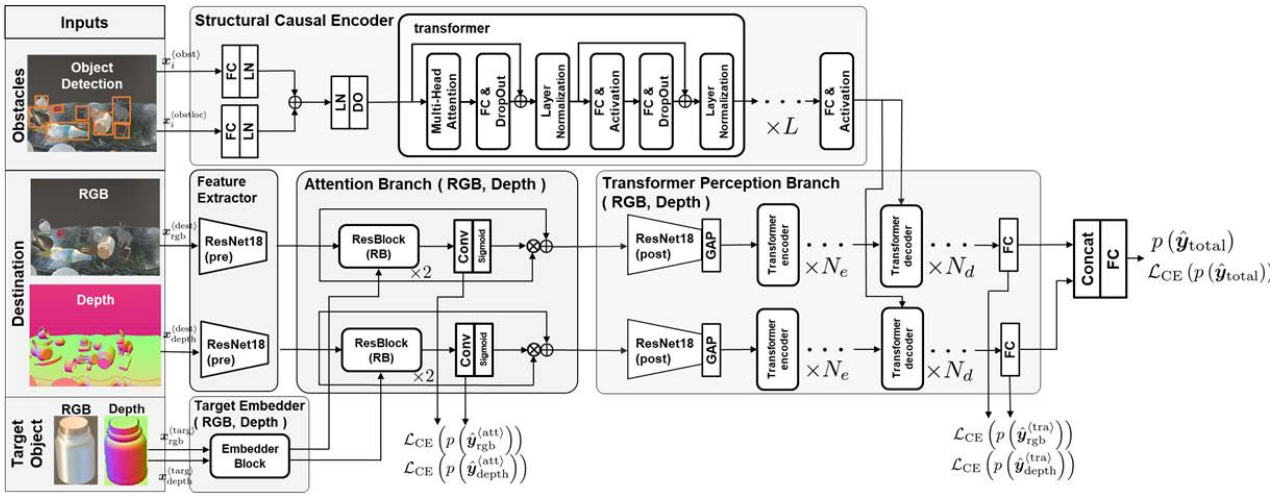

図 2 提案手法のネットワーク構造．図において FC，AP，Conv，LN，DO はそれぞれ全結合層，平均プーリング層，畳み込み層，Layer Normalization，ドロップアウト層を表す．

造である．

ネットワークの入力 $\boldsymbol{x}$ は配置領域の RGBD 画像 $\boldsymbol{X}^{\langle \mathrm{dest} \rangle}$，対象物体の RGBD 画像 $\boldsymbol{X}^{\langle \mathrm{targ} \rangle}$，障害物の物体領域群 $\boldsymbol{X}^{\langle \mathrm{obst} \rangle}$ である．ここに，

$$\boldsymbol{X}^{\langle \mathrm{dest} \rangle} = \left\{ \boldsymbol{x}_{\mathrm{rgb}}^{\langle \mathrm{dest} \rangle}, \boldsymbol{x}_{\mathrm{depth}}^{\langle \mathrm{dest} \rangle} \right\},$$

$$\boldsymbol{X}^{\langle \mathrm{targ} \rangle} = \left\{ \boldsymbol{x}_{\mathrm{rgb}}^{\langle \mathrm{targ} \rangle}, \boldsymbol{x}_{\mathrm{depth}}^{\langle \mathrm{targ} \rangle} \right\},$$

$$\boldsymbol{X}^{\langle \mathrm{obst} \rangle} = \left\{ \left(\boldsymbol{x}_i^{\langle \mathrm{obst} \rangle}, \boldsymbol{x}_i^{\langle \mathrm{obstloc} \rangle} \right) \,\middle|\, i = 1, \ldots, N_{\mathrm{obst}} \right\}$$

である．さらに，$\boldsymbol{x}_{\mathrm{rgb}}^{\langle \mathrm{dest} \rangle}$，$\boldsymbol{x}_{\mathrm{depth}}^{\langle \mathrm{dest} \rangle}$，$\boldsymbol{x}_{\mathrm{rgb}}^{\langle \mathrm{targ} \rangle}$，$\boldsymbol{x}_{\mathrm{depth}}^{\langle \mathrm{targ} \rangle} \in \mathbb{R}^{224 \times 224 \times 3}$，$\boldsymbol{x}_i^{\langle \mathrm{obst} \rangle} \in \mathbb{R}^{1024}$，$\boldsymbol{x}_i^{\langle \mathrm{obstloc} \rangle} \in \mathbb{R}^7$ はそれぞれ，配置領域の RGB 画像，配置領域の Depth 画像，対象物体の RGB 画像，対象物体の Depth 画像，i 番目の障害物の物体領域，物体領域の幾何学特徴量を表す．また，N_{obst} は Faster R-CNN [14] によって検出した障害物の個数を表す．$\boldsymbol{x}_{\mathrm{rgb}}^{\langle \mathrm{dest} \rangle}$，$\boldsymbol{x}_{\mathrm{depth}}^{\langle \mathrm{dest} \rangle}$，$\boldsymbol{x}_{\mathrm{rgb}}^{\langle \mathrm{targ} \rangle}$，$\boldsymbol{x}_{\mathrm{depth}}^{\langle \mathrm{targ} \rangle}$ は正規化後，224×224 の大きさにリサイズした．本研究では $\boldsymbol{x}_{\mathrm{rgb}}^{\langle \mathrm{dest} \rangle}$ および $\boldsymbol{x}_{\mathrm{depth}}^{\langle \mathrm{dest} \rangle}$ に対して ResNet18 の “conv4_x” の出力から特徴量を抽出した．$\boldsymbol{x}_i^{\langle \mathrm{obst} \rangle}$ は，$\boldsymbol{x}_{\mathrm{rgb}}^{\langle \mathrm{dest} \rangle}$ を Faster R-CNN に入力し，検出された各矩形領域の特徴量ベクトルを表す．この際，Faster-RCNN の事前学習および fine-tuning にはそれぞれ COCO データセット [15] および BILA-Sim データセットを用いた．また，Region Proposal Network には ResNet50 を用い，RoI pooling layer の “fc6” の出力を矩形領域の特徴量とした．$\boldsymbol{x}_i^{\langle \mathrm{obstloc} \rangle} = [x_1,\ y_1,\ x_2,\ y_2,\ w,\ h,\ w \times h]$ は各矩形領域の座標値をエンコードして得られるベクトルである．ここで，各矩形領域の左上と右下の頂点の座標を $(x_1,\ y_1)$，$(x_2,\ y_2)$，幅と高さをそれぞれ w，h とする．

4.2 Attention Branch

Attention Branch の入力は Feature Extractor の出力である配置領域の特徴量マップ $\boldsymbol{h}_{\cdot}^{\langle \mathrm{dest} \rangle} \in \mathbb{R}^{14 \times 14 \times 256}$ および Target Embedder の出力である対象物体の特徴量マップ $\boldsymbol{h}_{\cdot}^{\langle \mathrm{targ} \rangle} \in \mathbb{R}^{14 \times 14 \times 256}$ である．Attention Branch では，まず，$\boldsymbol{h}_{\cdot}^{\langle \mathrm{dest} \rangle}$ を 2 つの ResBlock 層に入力し，$\boldsymbol{h}_{\cdot}^{\langle \mathrm{targ} \rangle}$ は 1 つ目の ResBlock 層の後に結合する．その後，バッチ正規化および畳み込み層を適用する．畳み込みの後，sigmoid 関数に入力することで Attention map $\boldsymbol{a}_{\cdot} \in \mathbb{R}^{14 \times 14 \times 1}$ を生成する．次に，$\boldsymbol{y}_{\cdot}^{\langle \mathrm{att} \rangle} \in \mathbb{R}^2$ を GAP および softmax 関数の出力から得る．最後に，Transformer Perception Branch の出力 $\boldsymbol{w}_{\cdot} = (1 + \boldsymbol{a}_{\cdot}) \odot \boldsymbol{f}_{\cdot} \in \mathbb{R}^{14 \times 14 \times 256}$ を得る．ここに，$\odot$ はアダマール積を表す．

4.3 Structural Causal Encoder

Structural Causal Encoder は $\boldsymbol{x}_{\cdot}^{\langle \mathrm{obst} \rangle}$ および $\boldsymbol{x}_{\cdot}^{\langle \mathrm{obstloc} \rangle}$ を入力とし，配置領域における構造的知識の特徴量を出力するモジュールである．Structural Causal Encoder は埋め込み処理および Multi-Layer Transformer 層で構成される．

まず，埋め込み処理では，これらのベクトルをそれぞれ全結合層に入力し，得られた出力を連結した後，再び全結合層に入力することで，埋め込み処理の出力 $\boldsymbol{h}_i^{\langle \mathrm{obst} \rangle} \in \mathbb{R}^{1024}$ を得る．次に，$\boldsymbol{h}_i^{\langle \mathrm{obst} \rangle}$ を連結することで，$\boldsymbol{h}_{\mathrm{obstemb}} \in \mathbb{R}^{512 \times N_{\mathrm{obst}}}$ を得て，$\boldsymbol{h}_{\mathrm{obstemb}}$ に対して，Multi-Layer Transformer を適用する．はじめに，Self-Attention のための query $\boldsymbol{Q}^{(j)}$，key $\boldsymbol{K}^{(j)}$，value $\boldsymbol{V}^{(j)}$ を生成する．続いて，[3] と同様の Multi-Head Attention の計算式に基づき，Attention スコア $\boldsymbol{S}_{\mathrm{attn}}$ を算出する．$\boldsymbol{S}_{\mathrm{attn}}$ に全結合層，ドロップアウト層，正規化層を適用した後，全結合層と活性化関数による処理を行う．最後に，再び全結合層，ドロップアウト層，正規化層を適用する．この一連の処理を 1 つの transformer 層の encoder と定義する．そして，最後の transformer 層の encoder からの出力を $\boldsymbol{h}_{\mathrm{obst}} \in \mathbb{R}^{512}$ とする．

4.4 Transformer Perception Branch

Transformer Perception Branch は ResNet18 の後半部分と transformer で構成される．Transformer Perception Branch について，入力は $\boldsymbol{w}_{\cdot}$ および $\boldsymbol{h}_{\mathrm{obst}}$ であり，出力は $\boldsymbol{m}_{\cdot} \in \mathbb{R}^{512}$ である．

Transformer Perception Branch の transformer 層は encoder，decoder のそれぞれの役割をする部分に分かれている．まず，GAP 層の出力 $\boldsymbol{o}_{\cdot} \in \mathbb{R}^{512}$ に対して，transformer 層の encoder の処理を行った．これらの変形および処理を N_{e} 回繰り返す．この encoder の出力を $\boldsymbol{\alpha}_{\cdot} \in \mathbb{R}^{512}$ とする．decoder として，$\boldsymbol{h}_{\mathrm{obst}}$ に対して encoder と同様に Attention スコアを求めたのち，全結合層，ドロップアウト層，正規化層を適用し，さらに全結合層と活性化関数による処理を行った．これらの

処理で得られる値を $\boldsymbol{h}_{\text{obstmha}} \in \mathbb{R}^{512}$ とする．さらに，$\boldsymbol{h}_{\text{obstmha}}$ および $\boldsymbol{\alpha}_{\cdot}$ に対して Multi-Layer Transformer を適用する．はじめに，以下の式によって，query $\boldsymbol{Q}^{(j)}$，key $\boldsymbol{K}^{(j)}$，value $\boldsymbol{V}^{(j)}$ をそれぞれ Attention の Head 数 A だけ生成する．ただし，$j = 1, \ldots, A$ である．

$$\boldsymbol{Q}^{(j)} = \boldsymbol{W}_q^{(j)} \boldsymbol{\alpha}_{\cdot}$$

$$\boldsymbol{K}^{(j)} = \boldsymbol{W}_k^{(j)} \boldsymbol{h}_{\text{obstmha}}, \boldsymbol{V}^{(j)} = \boldsymbol{W}_v^{(j)} \boldsymbol{h}_{\text{obstmha}}$$

続いて，Multi-Head Attention の計算式に基づき Attention スコアを求めたのち，transformer 層の encoder と同様に計算結果に対する処理を行った．

Transformer Perception Branch の出力 $\boldsymbol{m}_{\text{rgb}}$ および $\boldsymbol{m}_{\text{depth}}$ から，$\boldsymbol{h}_{\text{total}} \in \mathbb{R}^2$ を求め，モデル全体の最終的な出力 $p\left(\hat{\boldsymbol{y}}_{\text{total}}\right)$ を得る．損失関数は [2] に示す式と同じ式を用いる．

5. 実験設定

本論文では新たに BILA-S データセットをシミュレーション環境上で作成した．用いたシミュレーション環境は [2] と同様である．

配置位置の決め方は以下の通りである．ロボットが配置可能な範囲は，前後 2cm・左右 10cm である．まず，[2] を用いて，ロボットのヘッドカメラで撮影した画像から衝突する確率の予測値の計算と配置領域の attention map の作成を行う．次に，これらを用いて安全に物体を置けると予測される位置を決める．衝突しないと予測した場合には配置可能範囲内で一番注目している部分から，衝突すると予測した場合には一番注目していない部分から，ランダムに選んだ位置を配置位置とする．本データセットの作成手順は [2] と同様である．

データセットでは各サンプルに次のように “衝突” または “接触” のラベルを自動的に付与した．シーン内の物体 i と j の相対速度 v_{ij} の最大値 $\max_{ij} |v_{ij}|$ が $V_c = 1.0 m/s$ より大きければ “衝突” と判断し $y = 1$ とした．それ以外は “接触” と判断し $y = 0$ とした．

データの事前処理については [2] と同様の処理を行った．BILA-S データセット統計情報に関しては以下の通りである．サイズは 11940 である．1 サンプルは配置領域の RGBD 画像，対象物体の RGBD 画像，正解ラベルからなる．ここで，対象物体の RGBD 画像は重複を除くと合計 14 種類からなる．また，正解ラベルが “衝突”，“接触” クラスであるサンプル数はそれぞれ 5226, 6714 である．

ハイパーパラメータの設定を次に示す．最適化手法は Adam($\beta_1 = 0.9$ $\beta_2 = 0.999$) を用い，学習率は 0.003 とした．また，Attention Branch の畳み込み層は $14 \times 14 \times 2$ とした．Transformer Perception Branch において，$N_e = 2$, $N_d = 2$，Attention の Head 数は 16 とし，Structural Causal Encoder において，$L = 2$, Attention の Head 数は 8 とした．さらに，損失関数の重みは $\lambda_{\text{rgb}}^{\langle \text{att} \rangle} = 1$, $\lambda_{\text{depth}}^{\langle \text{att} \rangle} = 1$, $\lambda_{\text{rgb}}^{\langle \text{tra} \rangle} = 1$, $\lambda_{\text{depth}}^{\langle \text{tra} \rangle} = 1$, $\lambda_{\text{total}} = 2$ とした．提案手法における訓練可能なパラメータ数は約 6700 万であり，積和演算数は 6.61×10^9 である．学習は GeForce RTX 3090（メモリ 24GB），64GB- RAM，Intel Corei9 10900K を搭載した計算機上で行った．学習および 1 サンプルあたりの推論にかかる時間はそれぞれは $30min$ および $10.3ms$ 程度であった．また，最大エポック数を 35 とし，検証集合において損失関数が最小となった時のテスト集合における精度を最終的な学習の精度とした．

表 1　BILA-S データセットにおける定量的結果

手法	TPB の decoder 層	Destination 画像サイズ	精度
[2]		384×384	77.64±2.32
Type1		384×384	80.43±0.66
Type2	✓	192×192	80.59±0.55
Ours	✓	384×384	**80.74±0.53**

6. 実験結果

各実験条件での定量的結果，定性的結果を以下に示す．ベースライン手法は Transformer PonNet [2] とした．本実験で使用した評価尺度は精度を用いた．

6.1 定量的結果

各手法の BILA-S データセットにおける定量的結果を表 1 に示す．なお実験は 5 回行い，その精度の平均値および標準偏差を示す．

表 1 より，ベースライン手法および提案手法の精度はそれぞれ 77.64 % および 80.74 % であり，提案手法がベースライン手法を 3.10 ポイント上回った．したがって，ベースライン手法と比較して提案手法の方が優れた結果が得られた．

6.2 定性的結果

図 3 に予測成功の例を示す．(a)，(b) の 1 列目の $\boldsymbol{x}_{\text{rgb}}^{\langle \text{dest} \rangle}(i)$ に対して，提案手法はそれぞれ $p(\hat{y} =$ “衝突”$) = 0.968$，$p(\hat{y} =$ “接触”$) = 0.944$ と予測した．

(a) に関して，ベースライン手法は，図 3(a) の 3 列目および 4 列目より，RGB 画像では画像全体に，Depth 画像では配置領域以外の部分に注目していることが分かる．一方で提案手法は，図 3(a) の 5 列目および 6 列目より，RGB 画像と Depth 画像の両方がウサギに注目していることが分かる．また，図 3(a) の 7 列目に示すように，このとき，配置領域内に存在しているすべての障害物は正しく検出されていた．この結果から，提案手法は配置領域内にあるウサギと対象物体間の “衝突” を正しく予測したと考えられる．

(b) に関して，ベースライン手法は，図 3(b) の 3 列目および 4 列目より，RGB 画像では右端に，Depth 画像では中央および右側に注目していることが分かる．一方で，提案手法は配置領域の箱の手前と，箱と缶の間 (図 3(b) の 6 列目) に注目している．また，図 3(b) の 7 列目に示すように，このとき，配置領域内にある箱と缶は正しく検出されていた．この結果から，提案手法は配置領域内の箱の手前および箱と缶の間を安全な領域として “接触” を正しく予測したと考えられる．

図 4 に予測失敗の例を示す．(a)，(b) の 1 列目の $\boldsymbol{x}_{\text{rgb}}^{\langle \text{dest} \rangle}$ に対して，提案手法はそれぞれ $p(\hat{y} =$ “衝突”$) = 0.823$，$p(\hat{y} =$ “接触”$) = 0.925$ と予測した．まず，(a) に関して，実際に対象物体を配置した領域はカップの左手前の安全領域であった．しかし，図 4(a) の 3 列目より，RGB 画像ではカップに注目していることが分かる．よって，カップに “衝突” すると予測したと考えられる．次に，(b) に関して，実際には対象物体を配置する際にロボットのアームと左側のアヒルが “衝突” していた．しかし，図 4(b) の 3 列目および 4 列目より，RGB 画像と Depth 画像の両方でアヒルの右側に注目していることが分かる．よって，アヒルの右側を安全な領域として判断し，“接触” と予測したと考えられる．また，Depth 画像では少しアヒルにも注目しているが，大部分が安全領域に注目していること、RGB

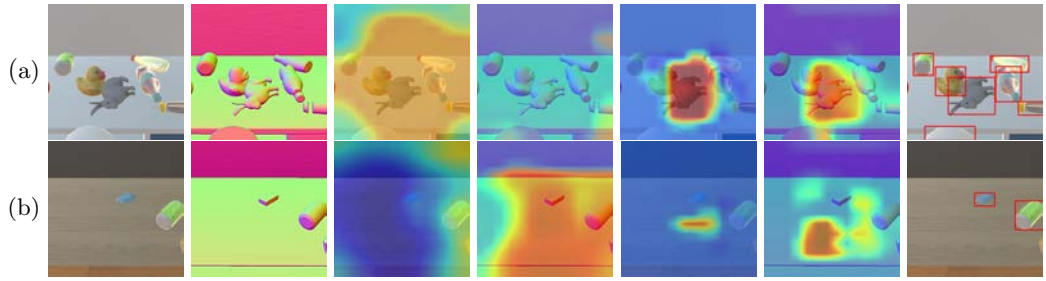

図 3 定性的結果（成功例）．左列から順番に $\boldsymbol{x}_{\mathrm{rgb}}^{\langle \mathrm{dest} \rangle}$，$\boldsymbol{x}_{\mathrm{depth}}^{\langle \mathrm{dest} \rangle}$，ベースライン手法での $\boldsymbol{a}_{\mathrm{rgb}}$，ベースライン手法での $\boldsymbol{a}_{\mathrm{depth}}$，提案手法での $\boldsymbol{a}_{\mathrm{rgb}}$，提案手法での $\boldsymbol{a}_{\mathrm{depth}}$，各障害物の矩形領域である．(a)True Positive，(b)True Negative.

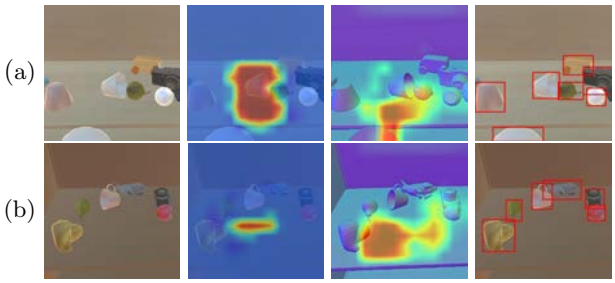

図 4 定性的結果（失敗例）．左列から順番に $\boldsymbol{x}_{\mathrm{rgb}}^{\langle \mathrm{dest} \rangle}$，$\boldsymbol{a}_{\mathrm{rgb}}$，$\boldsymbol{a}_{\mathrm{depth}}$，各障害物の矩形領域．(a)False Positive，(b)False Negative.

画像では安全領域にのみ注目していることから，予測に影響しなかったと考えられる．

6.3 Ablation Study

Ablation study として以下の 2 つの条件について検証を行った．

(1) Transformer Perception Branch 内の decoder 層の有無：Transformer Perception Branch 内の decoder 層を取り除くことで性能にどの程度の差が生じるかを調査した．ここで，$\boldsymbol{h}_{\mathrm{obst}}$ および $\boldsymbol{m}$. から $\boldsymbol{h}_{\mathrm{total}}$ を求めて用いた．

(2) $\boldsymbol{x}_{\mathrm{rgb}}^{\langle \mathrm{dest} \rangle}$ および $\boldsymbol{x}_{\mathrm{depth}}^{\langle \mathrm{dest} \rangle}$ のサイズの変更：$\boldsymbol{x}_{\mathrm{rgb}}^{\langle \mathrm{dest} \rangle}$ および $\boldsymbol{x}_{\mathrm{depth}}^{\langle \mathrm{dest} \rangle}$ のサイズを 384×384 から 192×192 に変更させた場合に，性能にどの程度の差が生じるかを調査した．

表 1 より，Type1 と提案手法を比較すると，Transformer Perception Branch 内の decoder 層を取り除くことで精度が 0.31 ポイント低下した．また，Type2 と提案手法を比較すると，$\boldsymbol{x}_{\mathrm{rgb}}^{\langle \mathrm{dest} \rangle}$ および $\boldsymbol{x}_{\mathrm{depth}}^{\langle \mathrm{dest} \rangle}$ のサイズを小さくすることで精度が 0.15 ポイント低下した．一方，Transformer PonNet に Structural Causal Encoder を加えることで精度が 3.10 ポイント向上した．以上より，Transformer Perception Branch 内の decoder 層の有無や，$\boldsymbol{x}_{\mathrm{rgb}}^{\langle \mathrm{dest} \rangle}$ および $\boldsymbol{x}_{\mathrm{depth}}^{\langle \mathrm{dest} \rangle}$ のサイズと比較して，Transformer PonNet に Structural Causal Encoder を加えることが最も重要であると考えられる．

7. 結論

本研究では，対象物体を指定した場所に配置する際の衝突予測タスクを扱った．提案手法の貢献は以下である．

- Transformer PonNet [2] を拡張し，新たに構造的知識を扱うために，Structural Causal Encoder モジュールを導入した．
- [3] を元にしたモデルが予測した安全領域に基づく配置方策を導入した．
- 提案手法は主要尺度である精度において，BILA-S データセット上でベースライン手法である Transformer PonNet [2] を上回った．

謝辞

本研究の一部は，JSPS 科研費 20H04269，JST ムーンショット，NEDO の助成を受けて実施されたものである．

参 考 文 献

[1] A. Bakhtin, L. van derMaaten, J. Johnson, L. Gustafson, and R. Girshick, “Phyre: A new benchmark for physical reasoning,” NeurIPS, vol.32, pp.5082–5093, 2019.

[2] 植田有咲，M. Aly，平川翼他，“生活支援ロボットによる物体配置タスクにおける Transformer PonNet に基づく危険性予測および可視化,” JSAI2021，p.2J1GS8a03，2021.

[3] 畑中駿平，上田雄斗他，“生活支援ロボットによる物体配置タスクにおける危険性予測および視覚的説明生成,” RSJ，2021.

[4] A. Saxena, J. Driemeyer, and A.Y. Ng, “Learning 3-d object orientation from images,” ICRA, pp.794–800, 2009.

[5] M.J. Schuster, J. Okerman, H. Nguyen, J.M. Rehg, et al., “Perceiving clutter and surfaces for object placement in indoor environments,” Humanoids, pp.152–159, 2010.

[6] K. Harada, et al., “Validating an object placement planner for robotic pick-and-place tasks,” Robotics and Autonomous Systems, vol.62, no.10, pp.1463–1477, 2014.

[7] Y. Jiang, M. Lim, C. Zheng, and A. Saxena, “Learning to place new objects in a scene,” The International Journal of Robotics Research, vol.31, no.9, pp.1021–1043, 2012.

[8] C. Wang and X. Guo, “Plane-based optimization of geometry and texture for rgb-d reconstruction of indoor scenes,” 3DV, pp.533–541, 2018.

[9] A. Magassouba, et al., “A multimodal classifier generative adversarial network for carry and place tasks from ambiguous language instructions,” IEEE RAL, vol.3, no.4, pp.3113–3120, 2018.

[10] J.A. Haustein, K. Hang, J. Stork, and D. Kragic, “Object placement planning and optimization for robot manipulators,” IROS, pp.7417–7424, 2019.

[11] P.S. Schmitt, W. Neubauer, W. Feiten, K.M. Wurm, G.V. Wichert, and W. Burgard, “Optimal, sampling-based manipulation planning,” ICRA, pp.3426–3432, 2017.

[12] M. Gualtieri, A. tenPas, et al., “Pick and place without geometric object models,” ICRA, pp.7433–7440, 2018.

[13] A. Magassouba, K. Sugiura, A. Nakayama, T. Hirakawa, et al., “Predicting and attending to damaging collisions for placing everyday objects in photo-realistic simulations,” Advanced Robotics, vol.35, no.12, pp.1–13, 2021.

[14] S. Ren, K. He, R. Girshick, and J. Sun, “Faster r-cnn: Towards real-time object detection with region proposal networks,” IEEE transactions on pattern analysis and machine intelligence, vol.39, no.6, pp.1137–1149, 2016.

[15] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, et al., “Microsoft coco: Common objects in context,” European conference on computer visionSpringer, pp.740–755 2014.