人物・物体・動作デコーダの分離による HOI 検出

陳 俊文[1,a)]　王 瀛成[1,b)]　柳井 啓司[1,c)]

概要

最近の one-stage HOI 検出手法は，物体デコーダの検出ターゲットを変更し，ボックスターゲットがクエリ埋め込みから明示的に分離されていないため，学習収束が遅い．本研究では人物デコーダ，物体デコーダ，動作デコーダからなる新しい one-stage フレームワークを提案する．さらに，学習効率を向上させるために，学習可能な物体と動作のラベル埋め込みを用いた事前知識を導出するノイズ除去学習方法を提案する．HICO-DET で本手法は学習エポックの 3 分の 1 で最先端手法より高い精度を達成した．

1. はじめに

最近の HOI（Human-Object Interaction）検出の研究は，主に物体検出のフレームワークに基づいて構築されている．HOI インスタンス $\{B_h, (B_o, O), V\}$ は，人物ボックス B_h，クラス O を持つ物体ボックス B_o，動作クラス V のトリプレットの定義に従い，検出方法は one-stage と two-stage に分かれる．One-stage アプローチでは，検出効率が高く，トレーニングが容易であるため，近年注目されている．

最近，Transformer [6] ベースの HOI 検出手法 [4], [9], [17], [22] は，物体検出器 DETR [1] を採用することにより，アテンションメカニズムのメリットを示した．QPIC [17] は，one-stage および two-stage の CNN ベースの手法におけるマッチング処理を行わず，エンコーダ・デコーダのアーキテクチャを採用し，インタラクションヘッドを用いて HOI インスタンスを直接予測する．しかし，QPIC の単一のデコーダは，人物と物体の位置関係やインタラクション認識の特徴が混ざっているため，HOI の予測精度が低下する．物体検出とインタラクション認識をカスケード的に分離した one-stage 手法 [8], [11], [18], [19], [21] は QPIC を改善したが，インスタンスデコーダでは人物と物体の検出はまだ混ざっているため，物体検出タスクで事前学習したモデルの性能を活用していない．

[1] 電気通信大学　大学院情報理工学研究科　情報学専攻
[a)] chen-j@mm.inf.uec.ac.jp
[b)] wang-y@mm.inf.uec.ac.jp
[c)] yanai@cs.uec.ac.jp

2. 関連研究

最近の研究では，QAHOI [3] と MSTR [10] は，Deformable Transformer デコーダの参照点を HOI インスタンスのアンカーと見なし，アンカーを用いて人物と物体検出を誘導する．しかし，QAHOI と MSTR のアンカーまたクエリ埋め込みは各 HOI 要素の予測に共用されているため，学習の収束が遅い．

3. 手法

特定の用途のためのクエリ埋め込みを明確にするために，本論文では，デコーダを分離した SOV（**S**ubject **O**bject **V**erb）フレームワークを提案した．また，学習効率を向上させるために，学習可能な物体と動作のラベル埋め込みを用いた事前知識を導入する **S**pecific **T**arget **G**uilded ノイズ除去学習方法 STG を提案した．図 1 は，SOV-STG のフレームワークを示している．SOV は特徴抽出器と SOV デコーダから構成される．学習可能なアンカーボックスとラベル埋め込みは，推論とノイズ除去学習のために HOI に特化した事前知識を提供する．

3.1 アンカーボックスによる HOI インスタンスの予測

クエリ埋め込みのデコーディングターゲットを明確にするために，SOV フレームワークは DAB-Deformable-DETR [13] のアテンションメカニズムを活用し，学習可能な subject と object のアンカーボックスを直接使用して人物と物体のボックスを予測する．また，adaptive shifted minimum bounding rectangle（ASMBR）を提案し，人物ボックスと物体ボックスの空間的な関係を考慮しながら動作ボックスを生成する．図 2 に示すように，デコーダの最終層で予測された人物ボックス $\boldsymbol{B}_s = (x_s, y_s, w_s, h_s)$ と物体ボックス $\boldsymbol{B}_o = (x_o, y_o, w_o, h_o)$（$(x, y)$：ボックス中心）を与えると，ASMBR（動作ボックス）は，次のように定義される：

$$\boldsymbol{B}_v = \left(\frac{x_s + x_o}{2}, \frac{y_s + y_o}{2}, w_v, h_v \right) \tag{1}$$

$$w_v = \frac{w_s + w_o}{2} + |x_s - x_o|, h_v = \frac{h_s + h_o}{2} + |y_s - y_o| \tag{2}$$

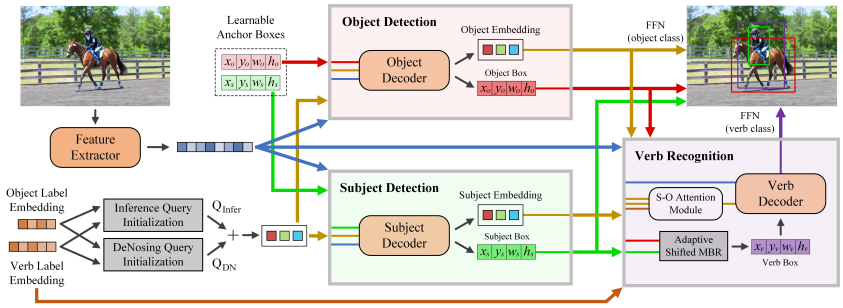

図 1　SOV-STG のフレームワークの全体図

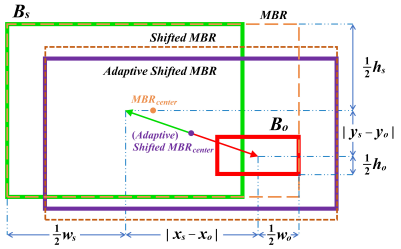

図 2　ASMBR のデザイン

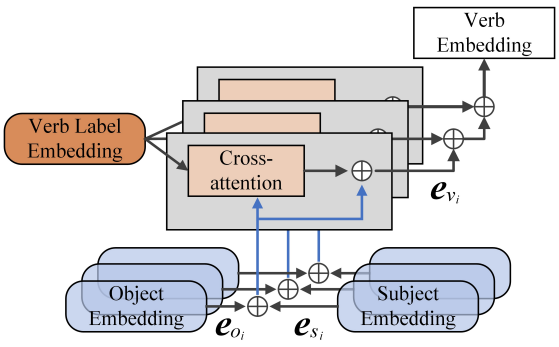

図 3　S-O アテンションモジュール

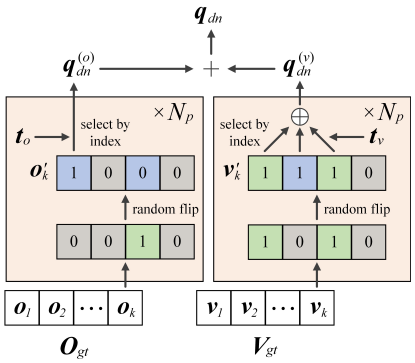

図 4　DN クエリの生成

3.2　SOV デコーダ

デコーディングターゲットを明確にするために，分離されたデコーダの設計が重要である．物体検出器の検出能力を維持するため，物体デコーダは検出タスクで学習された物体検出器から重みを初期化する．さらに，物体デコーダの重みを用いて，人物デコーダの初期化を行い，人物デコーダの学習負担を軽減させることができる．人物デコーダと物体デコーダは，人物アンカーボックス $\boldsymbol{B}_s$ と物体アンカーボックス $\boldsymbol{B}_o$ とクエリ埋め込み $\boldsymbol{e}$ を層ごとに並列に更新する．次に，物体と人物の埋め込みを Subject-Object（S-O）アテンションモジュール（セクション 3.3）に入れ，動作埋め込みを融合させる．最後に，動作埋め込みと動作ボックスを動作デコーダに与え，動作クラスを予測する．

3.3　動作デコーダと S-O アテンションモジュール

提案した動作ボックス（ASMBR）は人物と物体のボックスから直接生成されるため，動作デコーダは動作ボックスの予測学習をすることなく，動作認識に集中することができる．図 1 に示すように，動作認識部分は主に S-O アテンションモジュールと動作デコーダの 2 つの部分から構成されている．特徴量融合時に動作ラベルの知識を統合するために，S-O アテンションで動作ラベル埋め込みを融合させる．さらに，S-O アテンションに bottom-up path を設計し，下層から上層への情報を強化させる．図 3 で，S-O アテンションモジュールの計算を示している．i 番目の層 ($i > 1$) から，人物の埋め込み $\boldsymbol{e}_{s_i} \in \mathbb{R}^{N_q \times D}$ と物体の埋め込み $\boldsymbol{e}_{o_i} \in \mathbb{R}^{N_q \times D}$ を与え，N_q はクエリ数であると仮定する（D は Transformer の潜在次元）．まず，人物と物体の埋め込みを加算することで融合する．そして，融合した埋め込み量 $\boldsymbol{e}_{so_i} = (\boldsymbol{e}_{o_i} + \boldsymbol{e}_{s_i})/2$ を用いて，動作ラベル埋め込み量 $\boldsymbol{t}_v$ とのクロスアテンションの計算を行う．動作ラベルの基礎知識として学習可能な動作ラベル埋め込み $\boldsymbol{t}_v \in \mathbb{R}^{N_q \times D}$ については，次のセクション 3.4 で紹介する．さらに，レイヤーの情報を強化するために，bottom-up path を追加する．最後に，bottom-up path を追加した後の動作埋め込み $\boldsymbol{e}_{v_i}$ は次のように定義できる：

$$\begin{aligned}\boldsymbol{e}_{v_i} =&((\mathrm{CrossAttn}(\boldsymbol{e}_{so_{i-1}}, \boldsymbol{t}_v) + \boldsymbol{e}_{so_{i-1}}) \\ &+ (\mathrm{CrossAttn}(\boldsymbol{e}_{so_i}, \boldsymbol{t}_v) + \boldsymbol{e}_{so_i}))/2\end{aligned} \tag{3}$$

3.4　分離されたラベル埋め込み

図 1 に示すように，SOV デコーダのクエリ埋め込みを初期化するために，2 種類の学習可能なラベル埋め込みを使用している．D 次元の C_o 個ベクトルからなる（C_o は物体クラス数）物体ラベル埋め込み $\boldsymbol{t}_o \in \mathbb{R}^{C_o \times D}$ は，物体ラベルの事前知識として定義する．同様に，動作ラベル埋め込み $\boldsymbol{t}_v \in \mathbb{R}^{C_v \times D}$ を動作ラベル事前知識として定義する．物体ラベルと動作ラベルの事前知識を用いて，物体ラベル埋め込み $\boldsymbol{q}_o \in \mathbb{R}^{N_q \times D}$ と動作ラベル埋め込み $\boldsymbol{q}_v \in \mathbb{R}^{N_q \times D}$ を線形結合により初期化する．次に，物体ラベルと動作ラベルの埋め込みを加算して，推論クエリ埋め込み $\boldsymbol{q}_{ov} \in \mathbb{R}^{N_q \times D}$ が得られる．線形結合は 2 つの学習可能な行列 $\boldsymbol{A}_o \in \mathbb{R}^{N_q \times C_o}$ と $\boldsymbol{A}_v \in \mathbb{R}^{N_q \times C_v}$ を用いて以下のように定義される：

Method	Epoch	Backbone	Default			Known Object		
			Full	*Rare*	*Non-Rare*	*Full*	*Rare*	*Non-Rare*
Two-stage								
CATN [5]	12	ResNet-50	31.86	25.15	33.84	34.44	27.69	36.45
UPT [20]	20	ResNet-101-DC5	32.62	28.62	33.81	36.08	31.41	37.47
Liu *et al.* [14]	129	ResNet-50	33.51	30.30	34.46	36.28	33.16	37.21
One-stage								
QAHOI [3]	150	ResNet-50	26.18	18.06	28.61	-	-	-
AS-Net [4]	90	ResNet-50	28.87	24.25	30.25	31.74	27.07	33.14
QPIC [17]	150	ResNet-50	29.07	21.85	31.23	31.68	24.14	33.93
MSTR [10]	50	ResNet-50	31.17	25.31	32.92	34.02	28.83	35.57
Zhou *et al.* [21]	80	ResNet-50	31.75	27.45	33.03	34.50	30.13	35.81
CDN-B [19]	100	ResNet-50	31.78	27.55	33.05	34.53	29.73	35.96
GEN-VLKT-S [11]	90	ResNet-50	33.75	29.25	35.10	36.78	32.75	37.99
GEN-VLKT-M [11]	90	ResNet-101	34.78	31.50	35.77	38.07	34.94	39.01
GEN-VLKT-L [11]	90	ResNet-101	34.95	**31.18**	36.08	**38.22**	**34.36**	**39.37**
QAHOI-Swin-L [3]	150	Swin-Large-22K	35.78	29.80	37.56	37.59	31.36	39.36
FGAHOI-Swin-L [16]	150	Swin-Large-22K	37.18	30.71	39.11	38.93	31.93	41.02
SOV-STG-S	30	ResNet-50	33.80	29.28	35.15	36.22	30.99	37.78
SOV-STG-M	30	ResNet-101	34.87	30.41	36.20	37.35	32.46	38.81
SOV-STG-L	30	ResNet-101	**35.01**	30.63	**36.32**	37.60	32.77	39.05
SOV-STG-Swin-L	30	Swin-Large-22K	**43.35**	**42.25**	**43.69**	**45.53**	**43.62**	**46.11**

表 1 HICO-DET での結果

Method	Backbone	AP^{S1}_{role}	AP^{S2}_{role}
QPIC [17]	ResNet-50	58.8	61.0
UPT [20]	ResNet-101	61.3	67.1
MSTR [10]	ResNet-50	62.0	65.2
Liu *et al.* [14]	ResNet-50	63.0	65.2
CDN-L [19]	ResNet-101	63.9	65.9
GEN-VLKT-M [11]	ResNet-101	63.3	65.6
GEN-VLKT-L [11]	ResNet-101	63.6	65.9
SOV-STG-M	ResNet-101	63.7	65.2
SOV-STG-L	ResNet-101	63.9	65.4

表 2 V-COCO での結果

#	Verb Box	Default		
		Full	*Rare*	*Non-Rare*
(1)	Object Box	33.16	27.21	34.94
(2)	Subject Box	32.78	28.01	34.21
(3)	MBR	33.44	27.84	35.11
(4)	SMBR	33.41	28.22	34.97
(5)	ASMBR	**33.80**	**29.28**	**35.15**

表 3 動作のボックスのデザイン

Denoising Strategies				Default		
#	Box	Obj	Verb	*Full*	*Rare*	*Non-Rare*
(1)				32.99	28.28	34.40
(2)	✓			33.27	29.07	34.53
(3)	✓		✓	33.28	28.57	34.69
(4)		✓	✓	33.39	28.82	34.76
(5)	✓	✓		33.51	29.05	34.84
(6)	✓	✓	✓	**33.80**	**29.28**	**35.15**

表 4 ノイズ除去の実験

#	oDec	sDec	vDec	STG	Default		
					Full	*Rare*	*Non-Rare*
(1)	✓		✓	✓	32.68	28.21	34.02
(2)	✓	✓		✓	32.35	27.64	33.63
(3)	✓				30.14	22.82	32.32
(4)	✓	✓			30.62	24.60	32.42
(5)	✓	✓	✓		31.90	25.92	33.69
(6)	✓	✓	✓	✓	**33.80**	**29.28**	**35.15**

表 5 各モジュールの貢献

$$\boldsymbol{q}_o = \boldsymbol{A}_o \boldsymbol{t}_o, \quad \boldsymbol{q}_v = \boldsymbol{A}_v \boldsymbol{t}_v \tag{4}$$

$$\boldsymbol{q}_{ov} = \boldsymbol{q}_o + \boldsymbol{q}_v \tag{5}$$

3.5 Specific Target Guided Denoising

図 4 では，DN（DeNoising）クエリの初期化と，ground-truth HOI インスタンスにノイズを追加するプロセスを示している．Ground-truth の物体ラベル集合 $\boldsymbol{O}_{gt} = \{\boldsymbol{o}_i\}_{i=1}^{K}$ と動作ラベル集合 $\boldsymbol{V}_{gt} = \{\boldsymbol{v}_i\}_{i=1}^{K}$ を与えると，2 種類のラベル DN クエリが初期化されている．ここで，$\boldsymbol{o}_i$ と $\boldsymbol{v}_i$ は物体クラスと動作クラスの one-hot ラベルであり，k は ground-truth の HOI インスタンス数である．k 番目の ground-truth の HOI インスタンスに対して，物体ラベル $\boldsymbol{o}_k$ の ground-truth のインデックスを他の物体クラスのインデックスにランダムに反転させ，ノイズ物体ラベル $\boldsymbol{o}'_k$ を得て，N_p グループのノイズラベルが生成される．次に，物体 DN クエリ $\boldsymbol{q}^{(o)}_{dn} \in \mathbb{R}^{N_p \cdot K \times D}$ が，物体ラベル埋め込み $\boldsymbol{t}_o$ から，ノイズ物体ラベル $\boldsymbol{O}'_{gt}$ のインデックスによって収集される．動作ラベルは co-occurrence ground-truth クラスがあるため，co-occurrence ground-truth インデックスがノイズ動作ラベルに現れるように，ground-truth 動作ラベルの他のインデックスをランダムに反転してノイズ動作ラベル $\boldsymbol{v}'_k$ を生成する．物体 DN クエリと同じように，動作ラベル DN クエリ $\boldsymbol{q}^{(v)}_{dn} \in \mathbb{R}^{N_p \cdot K \times D}$ は，動作ラベル埋め込み $\boldsymbol{t}_v$ の中から，ノイズ動作ラベル $\boldsymbol{V}'_{gt}$ のインデックスによって選択された動作ラベル DN 埋め込みを合計したものである．最後に，物体 DN クエリと動作 DN クエリを連結し，ノイズ除去学習用の DN クエリ $\boldsymbol{q}_{dn} \in \mathbb{R}^{2N_p \cdot K \times D}$ を形成する．ノイズ除去学習により分割した事前知識を学習し，SOV の推論を誘導することができる．

4. 実験

4.1 実験設定

データセット HICO-DET[2]（トレーニングセット 38,118 枚，テストセット 9,658 枚）と V-COCO [7]（トレーニングセット 5,400 枚，テストセット 4,946 枚）データセットで実験を行った．HICO-DET では，600 種類の HOI クラス（117 種類のアクションクラスと 80 種類の物体クラスの組合せ）のインスタンス数によって，3 つのカテゴリ *Full*（全ての HOI クラス），*Rare*（インスタンスが 10 個未満の 138 クラス），*Non-Rare*（インスタンスが 10 個以上の 462 クラス）に分けられる．V-COCO では，COCO [12] と同じ 80 種類の物体クラスと 29 種類の動作クラスがアノテーションされており，29 種類の動作クラスがあるシナリオ 1 と 25 種類の動作クラスがあるシナリオ 2 の 2 つのシナリオ設定がある．

評価指標 評価指標は mAP（mean average precious）を採用する．True Positive の HOI インスタンスでは，予測された人物ボックスと ground-truth の人物ボックスの間の IoU が 0.5 より高く，予測された物体と ground-truth の物体のボックスの間の IoU も 0.5 より高くなる必要がある．

学習設定 GEN-VLKT [19] と同様に，全てのデコーダの層数を調整することにより，SOV-STG の 2 つのバリエーションを実装し，3 層デコーダの SOV-STG-S，6 層デコーダの SOV-STG-M と SOV-STG-L と表記する．Transformer の潜在次元は $D = 256$，クエリ数は $N_q = 64$ とする．DN 部分では，各 ground-truth HOI インスタンスに対して，$2N_p = 6$ グループのノイズラベルを生成する．HICO-DET データセットに対して，AdamW オプティマイザーで学習率 2e-4（バックボーンは 1e-5），重み減衰 1e-4 でモデルを学習する．バッチサイズは 32（GPU あたり 4 枚画像），学習エポックは 30（20 エポックで学習率減衰）と設定する．全ての実験は 8 枚の NVIDIA A6000 GPU で行っている．

4.2 最先端手法との比較

表 1 では，HICO-DET データセットにおいて，提案した SOV-STG と最近の SOTA 手法を比較した．ResNet-50 をバックボーンとする SOV-STG-S は，Default 設定の *Full* カテゴリで 33.80mAP を達成した．Deformable Transformer

を用いた one-stage の手法である QAHOI や MSTR と比較すると，アンカーポイントを用いた手法と比較して，SOV-STG はアンカーボックス事前知識とラベル事前知識の知識を受け，それぞれ 29.11%と 8.44%の mAP 改善を達成した．さらに，言語モデルの知識を使用せずに，SOV-STG-M は GEN-VLKT-M の 1/3 の学習エポックで 0.26%の mAP で上回る．V-COCO において，表 2 に示すように，SOV-STG-L は AP^{S1}_{role} で 63.9 mAP を達成し，GEN-VLKT-L を 0.47%で上回っていることが示されている．また，Swin Transformer [15] を用いてベストモデルの SOV-STG-Swin-L を学習した，43.62 mAP で新しい SOTA を達成した．

4.3 アブレーション実験

SOV-STG-S モデルを用いて，HICO-DET データセットでアブレーション実験を行った．

各モジュールの貢献 SOV-STG は柔軟なアーキテクチャと学習パイプラインで構成されている．各提案モジュールの貢献度を明らかにするために，表 5 では，提案モジュールを一つずつ削除し，HICO-DET データセットでアブレーション実験を行う．行 (5) は，STG を削除し，S-O アテンションモジュールを加算融合に置き換えた実験を行う．その結果，STG と S-O アテンションにより，”Full ”カテゴリで 5.96%の性能向上が見られた．次に，(4) において，(5) の動作デコーダを削除する．その結果，(4) と (5) を比較すると，動作デコーダがない場合は，性能が 4.01%低下している．次に，(3) では，人物デコーダと和融合モジュールを削除し，人物と物体ボックスの両方を物体デコーダで更新する．検出のデコード負担をバランスさせることなく，(4) と比較すると，1.57%性能が低下している．さらに，(1) と (2) では，それぞれ人物デコーダと動作デコーダで drop-one-out 実験を行った．(1) と (2) を比較すると，動作デコーダを使わないモデルの方が，人物デコーダを使わないモデルよりも性能が悪くなっており，動作デコーダがより重要な役割を担っていることが分かった．

動作ボックス 提案する ASMBR は，人物ボックスと物体ボックスの空間的関係を動的に考慮し，対応する領域から意味的特徴を抽出するよう動作デコーダを誘導する適応的な動作ボックスである．ASMBR の有効性を検証するために，他の動作ボックスを用いたアブレーション実験を行い，その結果を Table 3 に示す．(3)〜(5) の結果から，MBR の調整とシフトにより， 動作ボックスの性能が促進され，Full カテゴリで 1.08%，Rare カテゴリで 5.17%向上した．さらに，(1) と (2) では，動作ボックスとして物体または人物ボックスを直接使用し，その結果，物体の領域が動作予測においてより重要な役割を果たすことが示された．

ノイズ除去学習方法 表 4 では，ボックス座標，物体ラベル，動作ラベルの 3 つの部分について，ノイズ除去学習方法を実験した．(6) の結果は，SOV-STG-S の結果を示す．

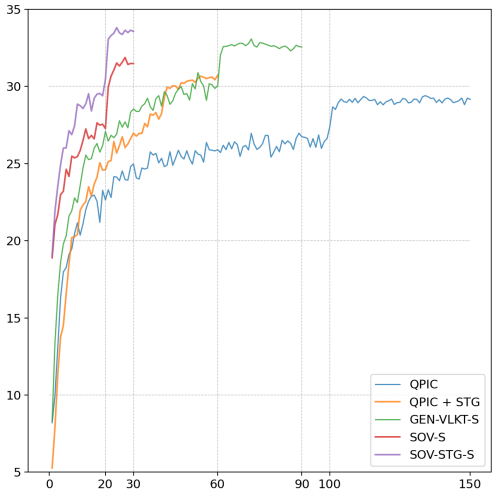

図 5 SOTA との学習中の精度の比較．

(1) において，ground-truth のボックス座標，物体ラベル，動作ラベルをノイズなしで直接モデルに入力する．その結果，(6) の完全ノイズ除去学習と比較して，精度は 2.40%低下した．(3)，(4)，(5) では，drop-one-out 実験を行い，ノイズ除去学習の各部が有効であることを示した．(2) と (3)，(4) と (3) の結果では，動作のノイズ除去は物体のノイズ除去との併用で性能が向上していることが分かった．

学習コストの軽減 提案手法が学習コストを軽減することを示すために，提案手法と SOTA モデル，QPIC と GEN-VLKT の学習プロセスを可視化して比較した．図 5 に示すように，SOV は，デコーディングのバランスを取るために，学習最初から高い AP を達成し，QPIC と GEN-VLKT よりも早く収束することができる．また，STG の学習パイプラインは，DETR ベースの HOI 検出モデルのクエリ埋め込みを構築するために，他の DETR ベースのモデルの改善に容易に適用することができる．図 5 に示すように，STG を QPIC に実装し，その結果，STG のノイズ除去学習により，学習の収束を早め，性能を向上させることができることを示した．

5. おわりに

本論文では，ターゲットに特化した分離されたデコーダ SOV とノイズ除去学習方法 STG を用いた新たな one-stage のフレームワークを提案する．提案したフレームワーク SOV-STG は，HOI インスタンスをボックスで表現する新しい形式を採用し，デコーディングに特化した事前知識を学習できる．また，設計されたアーキテクチャと効率的な学習方法により，より少ない学習コストで最先端の性能を達成することができる．SOV-STG は，HOI の検出を特定の事前知識とデコーダで分離しているため，それらのいずれかを改良することも容易である．今後は，言語モデルから初期化された物体ラベルや動作ラベルの事前知識を導入し，性能向上をさらに向上させることを目指す．

参考文献

[1] Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A. and Zagoruyko, S.: End-to-end object detection with transformers, *ECCV* (2020).

[2] Chao, Y.-W., Liu, Y., Liu, X., Zeng, H. and Deng, J.: Learning to detect human-object interactions, *WACV* (2018).

[3] Chen, J. and Yanai, K.: QAHOI: Query-Based Anchors for Human-Object Interaction Detection, *arXiv preprint arXiv:2112.08647* (2021).

[4] Chen, M., Liao, Y., Liu, S., Chen, Z., Wang, F. and Qian, C.: Reformulating HOI detection as adaptive set prediction, *CVPR* (2021).

[5] Dong, L., Li, Z., Xu, K., Zhang, Z., Yan, L., Zhong, S. and Zou, X.: Category-Aware Transformer Network for Better Human-Object Interaction Detection, *CVPR* (2022).

[6] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J. and Houlsby, N.: An image is worth 16x16 words: Transformers for image recognition at scale, *ICLR* (2021).

[7] Gupta, S. and Malik, J.: Visual Semantic Role Labeling, *arXiv preprint arXiv:1505.04474* (2015).

[8] Iftekhar, A., Chen, H., Kundu, K., Li, X., Tighe, J. and Modolo, D.: What to look at and where: Semantic and Spatial Refined Transformer for detecting human-object interactions, *CVPR* (2022).

[9] Kim, B., Lee, J., Kang, J., Kim, E.-S. and Kim, H. J.: HOTR: End-to-end human-object interaction detection with transformers, *CVPR* (2021).

[10] Kim, B., Mun, J., On, K.-W., Shin, M., Lee, J. and Kim, E.-S.: MSTR: Multi-Scale Transformer for End-to-End Human-Object Interaction Detection, *CVPR* (2022).

[11] Liao, Y., Zhang, A., Lu, M., Wang, Y., Li, X. and Liu, S.: GEN-VLKT: Simplify Association and Enhance Interaction Understanding for HOI Detection, *CVPR* (2022).

[12] Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P. and Zitnick, C. L.: Microsoft COCO: Common objects in context, *ECCV* (2014).

[13] Liu, S., Li, F., Zhang, H., Yang, X., Qi, X., Su, H., Zhu, J. and Zhang, L.: DAB-DETR: Dynamic Anchor Boxes are Better Queries for DETR, *ICLR* (2022).

[14] Liu, X., Li, Y.-L., Wu, X., Tai, Y.-W., Lu, C. and Tang, C.-K.: Interactiveness Field in Human-Object Interactions, *CVPR* (2022).

[15] Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S. and Guo, B.: Swin transformer: Hierarchical vision transformer using shifted windows, *ICCV* (2021).

[16] Ma, S., Wang, Y., Wang, S. and Wei, Y.: FGAHOI: Fine-Grained Anchors for Human-Object Interaction Detection, *arXiv preprint arXiv:2301.04019* (2023).

[17] Tamura, M., Ohashi, H. and Yoshinaga, T.: QPIC: Query-based pairwise human-object interaction detection with image-wide contextual information, *CVPR* (2021).

[18] Yuan, H., Wang, M., Ni, D. and Xu, L.: Detecting Human-Object Interactions with Object-Guided Cross-Modal Calibrated Semantics, *AAAI* (2022).

[19] Zhang, A., Liao, Y., Liu, S., Lu, M., Wang, Y., Gao, C. and Li, X.: Mining the Benefits of Two-stage and One-stage HOI Detection, *NeurIPS* (2021).

[20] Zhang, F. Z., Campbell, D. and Gould, S.: Efficient Two-Stage Detection of Human-Object Interactions With a Novel Unary-Pairwise Transformer, *CVPR* (2022).

[21] Zhou, D., Liu, Z., Wang, J., Wang, L., Hu, T., Ding, E. and Wang, J.: Human-Object Interaction Detection via Disentangled Transformer, *CVPR* (2022).

[22] Zou, C., Wang, B., Hu, Y., Liu, J., Wu, Q., Zhao, Y., Li, B., Zhang, C., Zhang, C., Wei, Y. et al.: End-to-end human object interaction detection with hoi transformer, *CVPR* (2021).