

DOM-Q-NET: 構造化言語の接地RL

Sheng Jiaト
ロント大学ベクトル研究
所 sheng.jia@utoronto.ca

Jamie Kiros
Google プレイ
ン kiros@google.com

Jimmy Baト
ロント大学 ベクトル研究
所 jba@cs.toronto.ca

概要

Web と対話するエージェントを構築すると、知識の理解と表現の学習が大幅に改善されます。ただし、現在の深層強化学習 (RL) モデルでは、離散アクション空間が大きく、状態間のアクション数が異なるため、Web ナビゲーション タスクは困難です。この作業では、これらの問題の両方に対処するための RL ベースの Web ナビゲーションの新しいアーキテクチャである DOM-Q-NET を紹介します。DOM 要素のクリックと文字列入力などの、さまざまなアクション カテゴリに対して個別のネットワークを使用して Q 関数をパラメータ化します。このモデルは、グラフ ニューラル ネットワークを利用して、標準的な Web ページのツリー構造の HTML を表します。専門家のデモンストレーションを使用しなくても、既存の作業に匹敵するか、それを上回ることができる MiniWoB 環境でモデルの機能を実証します。さらに、マルチタスク設定でトレーニングすると、サンプル効率が 2 倍向上し、モデルが学習した動作をタスク間で転送できるようになります。

1 はじめに

過去数年間、深層強化学習 (RL) は、アーケード ゲームのプレイ (Mnih et al., 2015) やロボット アームの操作 (Levine et al., 2016) などのタスクの解決に大きな成功を収めてきました。最近のニューラル ネットワークの進歩により、RL エージェントは人間の専門家による特徴量エンジニアリングを行わなくても、生のピクセルから制御ポリシーを学習できるようになりました。ただし、ディープ RL メソッドのほとんどは、エージェントへの入力に関節の角度と速度であるシミュレートされた物理環境、または入力がレンダリングされたグラフィックスであるシミュレートされたビデオ ゲームのいずれかで問題を解決することに重点を置いています。このようなシミュレートされた環境で訓練されたエージェントは、世界の豊かなセマンティクスについてほとんど知識がありません。

World Wide Web (WWW) は、現実世界に関する豊富な知識のリポジトリです。この複雑な Web 環境でゲートをナビゲートするには、エージェントはテキスト、画像、およびそれらの間の関係のセマンティックな意味について学習する必要があります。各アクションは、ツリー構造の HTML からのドキュメント オブジェクト モデル (DOM) との対話に対応しています。ソーシャル ネットワークで友達を見つける、興味深いリンクをクリックする、Google マップで場所を評価するなどのタスクは、特定の DOM 要素にアクセスし、ユーザー入力でその値を変更するものとして組み立てることができます。

Atari ゲームとは対照的に、Web タスクの難しさは、その多様性、広いアクション スペース、およびまばらな報酬シグナルに起因します。エージェントの一般的な解決策は、以前の研究での模倣学習によって専門家のデモを模倣することです (Shi et al., 2017; Liu et al., 2018)。劉ら。 (2018) は、MiniWoB (Shi et al., 2017) ベンチマーク タスクで非常に少数の専門家によるデモンストレーションで最先端のパフォーマンスを達成しましたが、その探索ポリシーには、HTML の専門知識で手作りされた制約付きのアクション セットが必要です。

この作業では、私たちの貢献は、Web ナビゲーション用に因数分解された Q 関数をパラメータ化する新しいアーキテクチャ DOM-Q-NET を提案することです。これは、専門家のデモンストレーションを使用しなくても、MiniWoB での既存の作業に匹敵するか、それを上回るようにトレーニングできます。Graph Neural Network (Scarselli et al., 2009; Li et al., 2016; Kipf & Welling, 2016) は、3 レベルの状態とアクションの表現を提供するためのメイン バックボーンとして使用されます。

特に、私たちのモデルは、ローカル DOM 表現のニューラル メッセージ パッシングと読み出し (Gilmer et al., 2017) を使用して、Web ページの近隣表現とグローバル表現を生成します。

また、3つの個別の多層パーセプトロン (MLP) (Rumelhart et al., 1985) を使用して、さまざまなアクション カテゴリ (「クリック」、「タイプ」、「モード」) の因数分解された Q 関数をパラメータ化することも提案します。アーキテクチャ全体が完全に微分可能であり、そのすべてのコンポーネントが共同でトレーニングされます。

さらに、Web ナビゲーション タスクのマルチタスク学習でモデルを評価し、学習した動作が Web インターフェイス上で伝達可能であることを示します。私たちの知る限り、これは RL エージェントが MiniWoB で複数のタスクを一度に解決する最初の例です。マルチタスク エージェントは、シングルタスク エージェントと比較して平均 2 倍のサンプル効率を達成することを示します。

2背景

2.1 DOMSを使用した Web ページの表現

ドキュメント オブジェクト モデル (DOM) は、HTML ドキュメントのプログラミング インターフェイスであり、そのようなドキュメントの論理構造を定義します。DOM はツリー構造で接続されており、Web ナビゲーションは DOM にアクセスし、オプションでユーザー入力によってそれを変更するものとして構成されています。オブジェクト指向プログラミングのオブジェクトと同様に、基本オブジェクトとして、各 DOM には「タグ」と「クラス」、「フォーカスされている」などのその他の属性があります。ブラウザーはこれらの属性を使用して、ユーザーの Web ページをレンダリングします。

2.2 強化学習

従来の強化学習の設定では、エージェントは、割引された将来の報酬の合計を最大化するために、無限の地平線の割引マルコフ決定プロセス (MDP) と対話します。MDP はタプル (S, A, T, R, γ) として定義されます。ここで、 S と A はそれぞれ状態空間とアクション空間であり、 $T(s|s, a)$ は状態 $s \in S$ に到達する遷移確率です。 $s \in S$ からアクション $a \in A$ をとることにより S を生成する。 R は遷移による即時報酬、 γ は割引係数。アクションのタプルの Q 値関数は $Q_{\pi}(s, a) = E[\sum_{t=0}^{\infty} \gamma^t r_t | s_0 = s, a_0 = a]$ と定義されます。ここで、 T は終了までのタイムステップ数です。この式は、状態 s から始まり、アクション a を実行し、終了までポリシーに従うと予想される将来の割引報酬を表します。最適な Q 値関数 $Q^*(s, a) = \max_{\pi} Q_{\pi}(s, a)$, $\forall s \in S, a \in A$ (Sutton & Barto, 1998) を満たします。最適性方程式 $Q^*(s, a) = \max_{a'} [R(s, a) + \gamma \sum_{s'} T(s'|s, a) Q^*(s', a')]$ 。

2.3 グラフニューラルネットワーク

無向グラフ $G = (V, E)$ の場合、Message Passing Neural Network (MPNN) フレームワーク (Gilmer et al., 2017) は、フォワードパスの2つのフェーズを定式化して、ノードレベルの特徴表現 h_v を更新します。ここで、 $v \in V$ およびグラフレベルの特徴ベクトル y^h 。メッセージパッシングフェーズでは、頂点更新関数 U を現在の隠れ状態とメッセージ $h_{t+1}(mt+1)$ に適用することにより、各ノードの隠れ状態を更新します。ここで、 $h_{t+1}(mt+1) = U(h_t, m_t)$ として計算されます。 $h_{t+1}(mt+1)$ は現在の隠れ状態とメッセージ h_t の関数であり、 m_t はメッセージ m_t の関数です。このプロセスは T タイムステップ実行されます。読み出し

$$h_v^t = U(h_v^{t-1}, \sum_{u \in N(v)} m_{vu}^t) \quad \text{with} \quad m_{vu}^t = M(h_v^{t-1}, h_u^{t-1})$$

の h_v^t は、 U の、 h_v^{t-1} の、 m_{vu}^t の、 M の、 h_v^{t-1} の、 h_u^{t-1} の、 $v \in G$ 。

2.4 グラフニューラルネットワークによる強化学習

グラフニューラルネットワーク (GNN) を使用して身体をモデル化するロボット移動の研究が行われています (Wang et al., 2018; Hamrick et al., 2018)。NerveNet は、GNN で学習したポリシーが、MLP で学習したポリシーよりも他の学習タスクにうまく移行することを示しています (Wang et al., 2018)。GNN を使用してポリシー全体をパラメータ化しますが、DOM-Q-NET は GNN を使用して因数分解された Q 関数の表現モジュールを提供します。ロボットのグラフ構造は静的ですが、Web ページのグラフ構造は時間ステップごとに変化する可能性があることに注意してください。移動ベースの制御タスクは密な報酬を提供しますが、Web ナビゲーションタスクは、エピソードの最後に 0/1 の報酬しかないまばらな報酬の問題です。私たちのタスクでは、モデルは目標指示に対するアクションの依存性も考慮する必要があります。

2.5 Webインターフェイス上のRLに関する以前の作業

シラ。(2017) Web ナビゲーションの多くのおもちゃのタスクで構成されるベンチマーク タスク、Mini World of Bits (MiniWoB) を構築しました。この環境は、Web ページのイメージと HTML の両方を提供します。彼らの研究は、視覚入力を使用するエージェントは、デモンストレーションが与えられたとしても、ほとんどのタスクを解決できないことを示しました。それから劉等。(2018) DOM 要素とゴールの間の一連の注意を使用する DOM-NET アーキテクチャを提案しました。正式な言語を使用してエージェントのアクション スペースを制約するワークフローのガイド付き探索により、デモンストレーションを使用する際に最先端のパフォーマンスとサンプル効率を達成しました。これらの以前のアプローチとは異なり、専門家のデモンストレーションや予備知識なしで Web ナビゲーションに取り組むことを目指しています。

3ニューラルDOM Qネットワーク

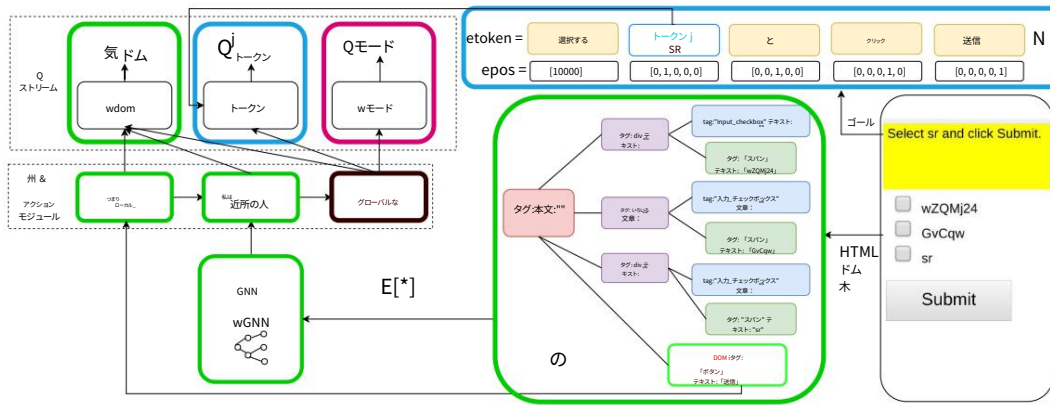


図 1: 右側の Web ページの DOM ツリー表現は、各 DOM が V のノードを表すグラフとして示されています。異なる色は、DOM の異なるタグ属性を示します。

DOM は、ローカル モジュール local として組み込まれ、GNN によって伝播されて隣接モジュール neighbor を生成します。グローバル モジュールの eglobal は、隣接モジュールから集約されます。Qdom ストリームは 3 つのモジュールすべてを使用しますが、Qtoken および Qmode ストリームはグローバルモジュールのみを使用します。ここで、'submit' および 'sr' トークンの Q 値は、それぞれ Qdom および Qtoken によって計算されます。

情報を見つけるために複数の Web ページまたはメニューをナビゲートする問題を考えてみましょう。V を現在の Web ページの DOM のセットとします。多くの場合、同じ Web 環境で達成できる複数の目標があります。自然言語文の形式でエージェントに提示される目標を考慮します。たとえば、図 1 の「sr を選択して [送信] をクリックします」や「テキストボックスを使用して Kanasha と入力し、[検索] を押してから、9 番目の検索結果を見つけてクリックします」などです。図 2。

G は、与えられたゴール センテンスの単語トークンのセットを表します。RL エージェントは、目標を達成した場合のみ報酬を受け取るため、報酬が少ない問題です。ナビゲーションの主な手段は、Web ページのボタンとテキスト フィールドを操作することです。

Web ナビゲーションの状態-アクション値関数を表すには、2 つの大きな課題があります。アクション スペースが膨大であること、アクションの数か状態によって大幅に異なる可能性があることです。以下の両方の問題に対処するために、DOM-Q-NET を提案します。

3.1 Web ナビゲーションのためのアクションスペース

アクション空間 A から 1 つのアクション a のみを選択する必要がある典型的な RL タスクとは対照的に、Atari のコントローラーの関節の動きのすべての組み合わせから 1 つを選択するなど (Mnih et al., 2015)、Web 上でのアクションを 3 つのフレームで構成します。アクションの明確なカテゴリ:

- DOM 選択 adom は、現在の Web ページ adom $\in V$ で単一の DOM を選択します。DOM の選択は、ボタンやチェックボックスのクリック、文字列入力を入力するテキスト ボックスの選択などの典型的なインタラクティブ アクションをカバーします。

- 単語トークンの選択 $a_{token} \in G$ は、指定されたゴール センテンスからワーク トークンを選択して、選択したテキスト ボックスに入力します。型付き文字列がゴール命令に由来するという仮定は、以前の研究 (Liu et al., 2018) と一致しています。
- モード $a_{mode} \in \{\text{click}, \text{type}\}$ は、エージェントが Web ページで行動するとき「クリック」または「入力」することを意図しているかどうかを環境に伝えます。 a_{mode} は 2 項アクションとして表されます。

各タイム ステップで、環境はアクションのタプル、つまり $a = (a_{dom}, a_{token}, a_{mode})$ を受け取りますが、 $a_{mode} = \text{type}$ でない限り a_{token} を処理しません。

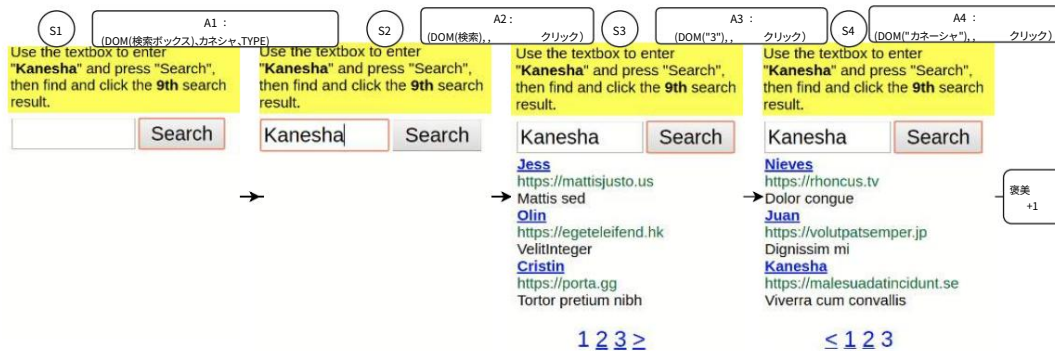


図 2: 検索エンジンのモデルによって実行された成功の軌跡。 S_i は状態であり、 $A_i = (a_{dom}, a_{token}, a_{mode})$ は、タイムステップ i での 3 つの異なるカテゴリのアクションに対するアクションのタプルです。

$DOM(x)$ は、Web ページ内の対応する要素 x のインデックスを表します。

3.2 因数分解された Q 関数

状態-アクション値関数を表す 1 つの方法は、 a_{dom} と a_{token} のすべての順列を考慮することです。たとえば、Mnih et al. (2015) は、Atari のジョイスティックの方向とボタンのクリックの順列を検討しています。 MiniWoB の場合、これにより、サイズ $|V|$ の巨大なアクション スペースが導入されます。 $\times |G|$ 。 DOM とゴール トークンの数 $|V| \times |G|$ は最大 60 と 18 に達し、一部のハード タスクでは合計アクション数が 1,000 を超えます。

アクション空間を縮小するために、 a_{dom} と a_{token} のアクション値が互いに独立している因数分解された状態アクション値関数を考えます。正式には、最適な Q 値関数を 3 つのアクション カテゴリの個々の値関数の合計として定義します。

$$Q(s, a) = Q(s, \text{恩寵, 贈り物, 贈り物}) + Q(s, \text{贈り物}) + Q(s, \text{贈り物}) + Q(s, \text{贈り物}). \quad (1)$$

独立性の仮定の下では、各 Q 値関数に対して貪欲なアクションを個別に選択することで、最適なポリシーを見つけることができます。したがって、因数分解された Q 関数の最適なアクションの計算コストは、2 次ではなく、DOM 要素の数と単語トークンの数に比例します。

$$a = \arg \max_{a_{dom}} Q(s, a_{dom}), \arg \max_{a_{token}} Q(s, \text{攻撃}), \arg \max_{\text{モード}} Q(s, \text{アモード}) \quad (2)$$

3.3 ウェブページのステートアクション埋め込みの学習

さまざまなチェックボックスのクリックや目に見えないタイプのフォームへの入力など、Web 上の多くのアクションは、類似したタグまたはクラス属性を共有しています。私たちの目標は、Web ページのこのような不変性を効果的に捉えながら、さまざまな時間ステップでさまざまな数の DOM 要素とゴール トークンを柔軟に処理できるニューラル ネットワーク アーキテクチャを設計することです。さらに、Web 上の情報を見つける場合、エージェントは、ローカル情報 (ボタンの名前とその周囲のテキストなど) とグローバル情報 (一般的なテーマなど) の両方を認識する必要があります。メニューから特定のボタンをクリックする合図は、おそらく散らばっています。

上記の問題に対処するために、図 1 に示すように、DOM-Q-NET と呼ばれる、現在の Web ページの各 DOM の因数分解された Q 値を計算する GNN ベースの RL エージェントを提案します。これは、ツリー構造の追加情報を使用します。因数分解された Q ネットワーク間で共有される、状態アクション表現、埋め込み e の学習をガイドする HTML。HTML ツリー構造を明示的にモデル化すると、DOM 要素間の関係情報がエージェントに提供されます。Web ページが与えられると、モデルは、GNN のノードレベルおよびグラフレベルの出力に対応する低レベルおよび高レベルのモジュールを使用して、連結された埋め込みベクトル e neighbor, e_{global} を学習します。

ローカル モジュールは、DOM の各埋め込み属性 e の連結する、具体的には、DOM 要素のグラフ間の最大、どのサイン距離を使用して、ソフトを測定します。

ゴールセンテンスの DOM v のアライメント。劉語理 (2018) は正確なアライメントを使用して、ゴール センテンスに現れるトークンを取得しますが、私たちの方法では、完全に一致しない同義語を検出できます。

$$e_{local} = e_{属性, 最大} \cos(\text{と 属性}, e_{goal}) \quad (3)$$

これは、各 DOM をクリックする非伝播アクション表現を提供し、GNN のスキップ接続です。

隣接モジュール隣接 は、グラフ ニューラル ネットワークを使用した の隣接コンテキストを組み込んだノード表現ハウスイン です。モデルは、ノード間でメッセージの受け渡しを実行します。ローカル モジュールは、このプロセスを初期化するため各ステップの中間状態、および非線形頂点更新 (Li 使用 2016) は、Gate Recurrent GNN の (Chen et al., 2014) を採用して、このプロセスを T 回のステップで実行して、最終的な隣接埋め込みを取得します。

$$m_{i,t+1}^{nei} = \sum_{k \in N(i)} w_{GNN}^{nei, k, t} e_{neighbor}^{k, t} \quad (4)$$

$$e_{i,t+1}^{nei} = \text{GRU}(m_{i,t+1}^{nei}, e_{neighbor}^{i, t}) \quad (5)$$

コンテキスト情報を組み込むことにより、このモジュールには現在のページの状態表現と、特定の DOM をクリックするという伝播されたアクション表現が含まれるため、このモジュールのみを使用して Q 値関数を概算できます。

グローバル モジュール e_{global} は、読み出しフェーズ後の Web ページ全体のハイレベルな機能表現です。これは、因子分解された 3 つの Q ネットワークすべてで使用されます。目標情報を明示的に組み込む場合と組み込まない場合で、このようなグローバルな埋め込みを取得するために、2 つの読み出し関数を調査します。

1) max-pooling を使用して、Web ページのすべての DOM 埋め込みを集約します。

$$e_{global} = \max_{v \in V} e_{local} \quad (6)$$

2) ゴールベクトルをアテンションクエリとしてゴールアテンションを使用します。これは、Velickovic らとは対照的です。(2018) 注意はメッセージ パッシング フェーズで使用され、クエリはタスク依存の表現ではありません。ゴール ベクトル h_{goal} を得るために、各ゴール トークン e_{token} は、図 1 に示すように、ワンホット位置エンコーディング ベクトル e_{pos} と連結されます。目標表現をプールの Vaswani らによって動機付けられました。(2017) では、ローカル埋め込みをキーとして、近隣埋め込みを値として、スケールされたドット積の注意を使用します。Elocal と Eneighbor は、それぞれ (e local) と (e neighbor) のパック表現であることに注意してください。ここで、 $E_{local} \in \mathbb{R}(V, dk)$

$$E_{neighbor} = \text{softmax}(\sqrt{dk} \cdot \frac{h_{goal} E_{attn}}{E_{neighbor} \cdot e_{global}}) \cdot E_{neighbor}, \quad e_{global} \cdot \text{attn} = [e_{global}, e_{attn}] \quad (7)$$

説明図を付録 6.2 に示し、ノード レベルの特徴をゴール ベクトルと連結する簡単な方法を付録 6.3 に示します。この方法も目標情報の組み込みに有効であることが分かったが、モデルのサイズが大きくなってしまふ。

学習DOMを選択する Q 値関数は、2 層の MLP によってパラメータ化されます。

$Q_{\text{dom}} = \text{MLP}(e_{\text{dom}}; w_{\text{dom}})$ 、DOM 埋め込み e の連結を取ります

ル、

入力として neighbour、eglobal。同様に、単語トークンとモードを選択するための Q 値関数は、それぞれ MLP(etoken, eglobal; wtoken) と MLP(eglobal;

wmode) を使用して計算されます。図 1 を参照してください。埋め込み行列を含むすべてのモデル パラメータは、ゼロから学習されます。 $\theta = (E, w_{\text{GNN}}, w_{\text{dom}}, w_{\text{token}}, w_{\text{mode}})$ を埋め込み行列、グラフ ニューラル ネットワークの重み、および因数分解された Q 値関数の重みを含むモデル パラメータとします。モデル パラメータは、二乗 TD エラーを最小化することによって更新されます (Sutton, 1988 年)。

$$\min_{\theta} E(s, a, r, s) \text{ リプレイ } [y^{\text{DQN}} - Q(s, \text{恵み}; \theta) - Q(s, \text{攻撃}; \theta) - Q(s, \text{恵み}; \theta)]^2, \quad (8)$$

ゴリズムのように、遷移ペア (s, a, r, s) がリプレイ バッファからサンプリングされ、ターゲット Q は標準の DQN 及び θ を使用して y ターゲット Q 値が因数分解されます。

$$Q^{\text{DQN}} + \gamma \max_y \left[Q(s, a, \text{dom}; \theta) + \max_{a_{\text{token}}} Q(s, \text{トークン}; \theta) + \max_{\text{モード}} Q(s, a, \text{モード}; \theta) \right] \quad (9) = r$$

3.4 学習した行動を伝達するためのマルチタスク学習

学習した行動を伝達し、モデルによって複数のタスクを解決することの有効性を評価するために、複数の環境で動作する単一のエージェントをトレーニングします。さまざまなタスクからの遷移が共有リプレイ バッファに収集され、各環境でアクションを実行した後にネットワークが更新されます。詳細については、Alg. 1 を参照してください。

4 実験

最初に、以前の研究と比較することにより、大きな行動空間に対する提案されたモデルの一般化機能を評価します。付録 6.4 で定義されているように、さまざまな難易度を持つタスクが MiniWoB から選択されます。次に、マルチタスク学習からのモデルを使用して、サンプル効率の向上を調査します。我々は、各代表的なモジュールの有効性を正当化するためにアブレーション研究を行い、続いて、マルチタスクとシングルタスクの設定における目標注意からのサンプル効率の向上を比較します。ハイパーパラメータについては、付録 6.1 で説明しています。

4.1 DOM-Q-NET ベンチマーク MINIWOB

Web ナビゲーション タスクはまばらな報酬の問題であり、リプレイ バッファを使用したポリシー外の学習はよりサンプル効率が高いため、Rainbow (Hessel et al., 2018) の 4 つのコンポーネントを使用して Q ラーニング アルゴリズムを使用してエージェントをトレーニングします。4 つのコンポーネントは、DDQN (Van Hasselt et al., 2016)、Prioritized replay (Schaus et al., 2016)、Multi-step learning (Sutton, 1988)、および NoisyNet (Fortunato et al., 2018) で設定は各タスクで使用され、ここでは、DOM 要素のクリックと文字列の入力のみが必要なタスクを検討しています。タスクが正しく完了した場合、エージェントは +1 の報酬を受け取り、それ以外の場合は 0 の報酬を受け取ります。ソーシャル メディアを除くすべてのタスクに対して、 $T=3$ ステップのニューラル メッセージ パッシングを実行します。ソーシャル メディアでは、 $T=7$ ステップを使用して大きな DOM スペースに対処します。

評価指標: トレーニング中の最後の 100 エピソードの報酬の移動平均をプロットします。

以前の研究 (Shi et al., 2017; Liu et al., 2018) に従い、成功率を報告します。これは、報酬 +1 で終了するテスト エピソードの割合です。報告されたそれぞれの成功率は、4 回の異なる実行の平均に基づいており、付録 6.6 で実験プロトコルについて説明しています。

結果: 図 3 は、Liu らによって選択されたほとんどのタスクで DOM-Q-NET が 100% の成功率に達することを示しています。(2018)、クリック ウィジェット、ソーシャル メディア、メール受信トレイを除く。私たちのモデルは、ソーシャル メディアで 86% の成功率に達しており、ゴール アテンションを使用することで、モデルはクリック ウィジェットとソーシャル メディアを 100% の成功率で解決できます。探索中にアクションセットに制約を与える、事前に定義された目標のフィールドを使用する、専門家のデモンストレーションを表示するなどの事前知識は使用しませんでした。具体的には、私たちのモデルは、デモンストレーションを伴う以前の作業では解決できなかった、長期的なタスクである日付の選択を解決します。このタスクは多くの同様のアクションを想定していますが、大きなアクション スペースがあります。模倣学習やガイド付き探索を使用しても、ニューラル ネットワークは目に見えない多様な DOM の状態とアクションを一般化する表現を学習する必要があり、私たちのモデルはそれを証明しています。

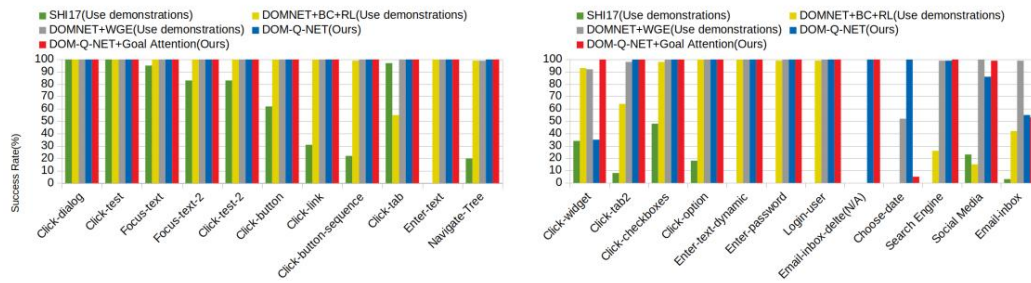


図 3: DOM-Q-NET と Shi らのパフォーマンスの比較。(2017);劉ら。(2018)

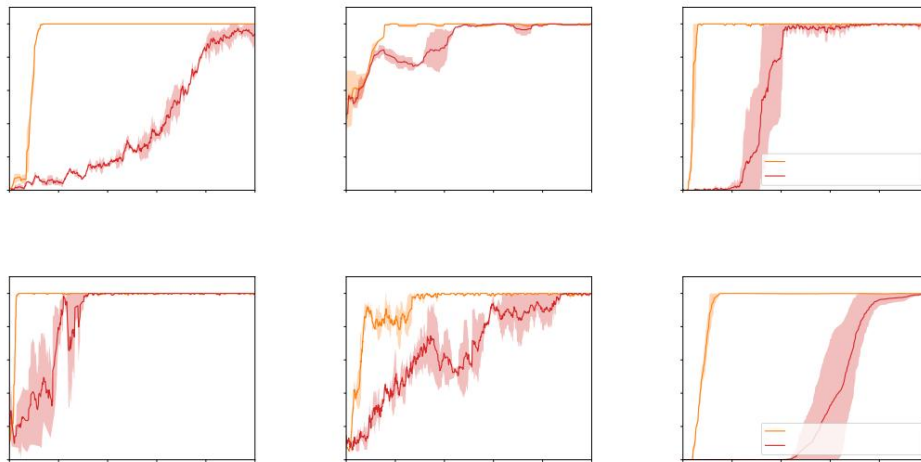


図 4: マルチタスクの比較: 9 つのマルチタスク DOM-Q-NET とゴールアテンションは一貫してサンプル効率が悪く、他のタスクの結果は、付録 6.7.1 に示されています。ga = 目標注意。

4.2 マルチタスク

マルチタスク エージェントとシングルタスク エージェントのサンプル効率を比較するために、2 つのメトリックが使用されます。

- M_{total} マルチタスク エージェント: すべてのタスクを解決したときに観測されたフレームの総数。
 M_{total} シングルタスク エージェント: 各タスクを解決するために観測されたフレーム数の合計。
- M_{task} : 特定のタスクを解くために観測されたフレーム数。

約 $M_{total} = 63000$ フレームを使用して、2 倍のサンプル効率で 9 つのタスクを解決するマルチタスク エージェントをトレーニングしましたが、シングルタスク エージェントは $M_{total} = 127000$ フレームを組み合わせ使用します。図 4 は、9 つのタスクのうち 6 つのプロットを示しています。特に、login-user と click-checkbox は、マルチタスク学習を使用して 40000 少ないフレームで解決されます。次に、図 5 に示す 2 つのハード タスクを含めました。シングルタスク エージェントが 11 のタスクを解決するために $M_{total} = 477000$ フレームを観察したサンプル効率と比較すると、マルチタスク エージェントは、最後のソーシャル メディアがタスクは、図 5 に示すように解決されます。さらに、プロットは、より単純なタスクを使用したマルチタスク学習が、ハード タスクに観測されたフレームを使用する際により効率的であることを示しており、これら 2 つのタスクのみを使用したマルチタスク学習よりも優れた M_{task} を達成しています。これらの結果は、私たちのモデルが、学習した行動を異なるタスク間で積極的に伝達できることを示しています。

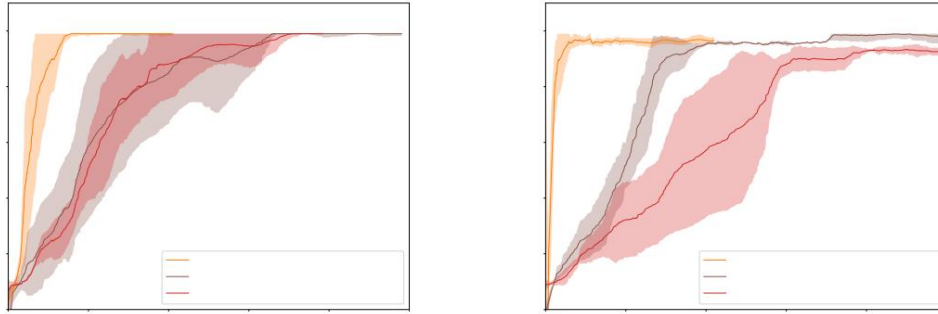


図 5: マルチタスク学習による 2 つのハード タスク、ソーシャル メディア (左) と検索エンジン (右) のサンプル効率の比較。9 マルチタスクとは、図 4 で説明したタスクを指します。

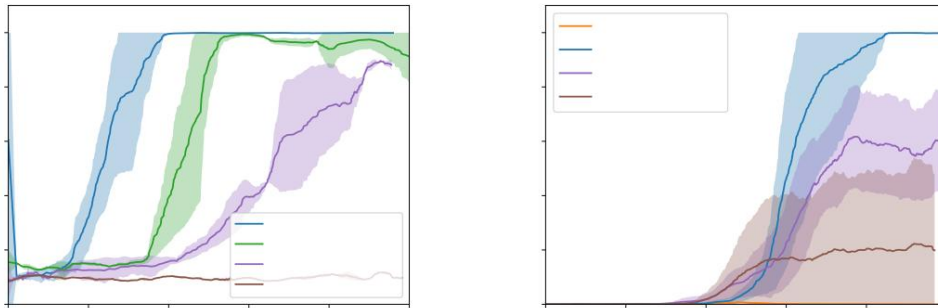


図 6: l = ローカル、 n = ネイバー、 g = グローバル モジュールのアブレーション実験。dom q net - g は、グローバル モジュールのない DOM-Q-NET です。dom q net - l - g は、隣接モジュールのみを持つ DOM-Q-NET です。dom q net - ng はローカルモジュールのみの DOM-Q-NET です。

4.3 DOM表現モジュールに関するアブレーション研究

Qdomストリームに各モジュールを使用することの有効性を正当化するために、アブレーション実験を行います。提案されたモデルを、Qdom を計算するためのいくつかのモジュールを省略した 3 つの割引バージョンと比較します: (a) $edom = \frac{e}{n}$, (b) $edom = \frac{e}{neighbor}$, (c) $edom = \frac{e}{[e]}$

図 6 は選択された 2 つのタスクを示しています。クリック チェックボックスの失敗例は、多くの DOM が同じ属性を持ち、コンテキストの違いにもかかわらずまったく同じ表現を持つため、neighbor モジュールなしの DOM 選択は単純に機能しないことを示しています。劉ら。(2018) は、メッセージ パッシングを手作りすることでこの問題に対処しました。最適な動作への DOM-Q-NET のより速い収束は、隣接モジュールの制限と、グローバルおよびローカル モジュールが Web ページの高レベルおよび低レベル表現へのショートカットをどのように提供するかを示しています。

4.4 ゴール・アテンションの有効性

ほとんどの MiniWoB タスクには、クエリの単語トークンとリンクが DOM と一致する「クエリ ワードを検索に入れ、一致するリンクを見つける」などの 1 つの望ましい制御ポリシーしかありません。したがって、私たちのモデルは、クリックウィジェットなどの例外を除いて、目標表現をネットワークに供給することなく、ほとんどのタスクを解決します。付録 6.7 は、ゴール・アテンションを含むさまざまなゴール・エンコーディング方法を使用したモデルの比較を示しています。いくつかのタスクで見られるように、ゴール・アテンションの効果は明白ではありません。ただし、図 7 は、サンプル効率の向上を示しています。

マルチタスクの学習設定では、目標注意を使用することはかなりの効果があり、このゲインは、シングルタスクの設定でのゲインよりもはるかに大きくなります。これは、エージェントが、複数のタスクを解決する際に、さまざまな目標指示が与えられた場合に、DOMツリーのさまざまな部分に注意を払うことをうまく学習したことを示しています。

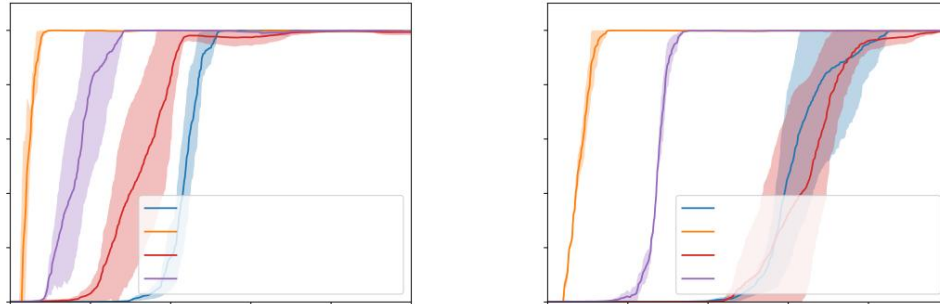


図 7: シングルおよびマルチタスク学習に対する目標注意の効果 (ga=目標注意)

5 ディスカッション

ゴール アテンション、ローカル ワード埋め込み、およびグラフ ニューラル ネットワークを使用して、因数分解された Q 関数をパラメーター化するための新しいアーキテクチャを提案します。このモデルで Web ナビゲーションの構築に貢献します。デモンストレーションがなくても、大きなアクション スペースで比較的難しいタスクを解決し、マルチタスク学習から学習した動作を転送します。これは、Web ナビゲーションの 2 つの重要な要素です。将来の作業のために、エージェントが学習した行動を一般化するために使用できるタスクの単純なインスタンスが環境にない、電子メールの受信トレイなどのタスクの探索戦略を調査します。劉ら。 (2018) は、探索を導く興味深い方法を示しました。

もう 1 つの作業は、各 DOM 要素の Q 値を評価するための計算コストを削減することです。

最後に、検索エンジンの使用にメソッドを適用する予定です。質問応答などのタスクは、エージェントが検索をクエリし、結果ページをナビゲートし、目的の目標を解決するための関連情報を取得する機能から恩恵を受ける可能性があります。検索をクエリしてナビゲートする機能は、現実的な環境でエージェントをブートストラップして、タスク指向の知識を取得し、サンプル効率を向上させるためにも使用できます。

謝辞:このプロジェクトでは、Dopamine (Castro et al., 2018) によるセグメント ツリーの実装を使用したことを認めます。

再現性:コードとデモは、それぞれ<https://github.com/Sheng-J/DOM-Q-NET>と<https://www.youtube.com/channel/UCrGsYub9lKCYO8dlREC3dnQ>で入手できます。

参考文献

パブロ サミュエル カストロ、サブホディーブ モイトラ、カルレス ゲラダ、サウラブ クマール、マーク G. ベルマーレ。

ドーパミン: 深層強化学習の研究フレームワーク。 CoRR,abs/1812.06110,2018。

URL <http://arxiv.org/abs/1812.06110>。

Kyunghyun Cho, Bart van Merriënboer, C. Aglar Gulc, ehre, Fethi Bougares, Holger Schwenk, および Yoshua Bengio。

統計的機械翻訳のための RNN エンコーダー/デコーダーを使用したフレーズ表現の学習。 CoRR,abs/1406.1078,2014。

URL <http://arxiv.org/abs/1406.1078>。

メイレ・フォルトウナート、モハマド・ゲシュラギ・アザール、ビラル・ピオット、ジェイコブ・メニック、イアン・オスバンド、アレックス・グレイブス、ヴラド・ムニー、レミ・ムノス、デミス・ハサビス、オリヴィエ・ピエキン 他探索のためのノイズの多いネットワーク。
ICLR, 2018 年。

ジャスティン・ギルマー、サムエル・S・シェーンホルツ、パトリック・F・ライリー、オリオール・ピニャルス、ジョージ・E・ダール。ニューラル量子化学のメッセージパッシング。 ICML,2017年。

ジェシカ・B・ハムリック、ケルシー・R・アレン、ピクター・バブスト、ティナ・チュー、ケビン・R・マッキー、ジョシュア・B・テネン・バウム、ピーター・W・バツタリア。人間と機械の物理的構造の関係誘導バイアス。 arXiv プレプリント arXiv:1806.01203, 2018.

Matteo Hessel, Joseph Modayil, Hado Van Hasselt, Tom Schout, Georg Ostrovski, Will Dabney, Dan Horgan, Bilal Piot, Mohammad Azar, David Silver. Rainbow: 深層強化学習の改善を組み合わせます。 2018年AAAIにて。

ディーデリック P. キングマとジミー パ。 Adam: 確率的最適化の手法です。 CoRR, abs/1412.6980, 2014. URL <http://arxiv.org/abs/1412.6980>。

トーマス・N・キップとマックス・ウェリング。グラフ畳み込みネットワークによる半教師付き分類。 CoRR, abs/1609.02907, 2016. URL <http://arxiv.org/abs/1609.02907>。

セルゲイ・レヴィーン、チェルシー・フィン、トレヴァー・ダレル、ピーター・アビール。深層視覚運動ポリシーのエンド ツー エンド トレーニング。 JMLR, 2016年。

Yujia Li, Daniel Tarlow, Marc Brockschmidt, Richard Zemel。ゲートッド グラフ シーケンス ニューラル ネットワーク。 ICLR, 2016年。

Evan Zheran Liu, Kelvin Guu, Panupong Pasupat, Tianlin Shi, および Percy Liang。ワークフローに沿った探索を使用して Web インターフェイスで学習します。 ICLR, 2018 年。

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Belle mare, Alex Graves, Martin Riedmiller, Andreas K Fijeland, Georg Ostrovski 他深層強化学習による人間レベルの制御。ネイチャー, 2015年。

David E Rumelhart, Geoffrey E Hinton, および Ronald J Williams。エラー伝播による内部表現の学習。テクニカル レポート、カリフォルニア大学サンディエゴ ラ ホーヤ研究所認知科学, 1985 年。

フランコ・スカルセリ、マルコ・ゴリ、アー・チュン・ツォイ、マルクス・ハーゲンブフナー、ガブリエレ・モンファルディーニ。グラフ ニューラル ネットワーク モデル。ニューラル ネットワークに関する IEEE トランザクション, 2009 年。

トム・ショール、ジョン・クワン、イオアニス・アントノグロウ、デビッド・シルバー。優先体験リプレイ。の ICLR, 2016 年。

Tianlin Shi, Andrej Karpathy, Linxi Fan, Jonathan Hernandez, および Percy Liang。ビットの世界: Web ベースのエージェント向けのオープン ドメイン プラットフォーム。 ICML, 2017年。

リチャード・S・サットン。時間差の方法による予測の学習。機械学習では、1988年。

リチャード・S・サットンとアンドリュー・G・バルト。強化学習入門、第 135 巻。MIT プレス ケンブリッジ, 1998 年。

ハド・ヴァン・ハッセルト、アーサー・ゲス、デヴィッド・シルバー。 double q による深層強化学習 学ぶ。 2016年AAAIにて。

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Lion Jones, Aidan N Gomez, ウカシユ・カイザー、イリヤ・ポロスキ。必要なのは注意だけです。 NIPS, 2017年。

ベタル・ヴェリコピッチ、ギエム・ククルル、アランチャ・カサノバ、アドリアナ・ロメロ、ピエトロ・リオ、ヨシュア・ベンジオ。注意ネットワークをグラフ化します。 ICLR, 2018 年。

Tingwu Wang, Renjie Liao, Jimmy Ba, Sanja Fidler. Nervenet: 構造化されたポリシーを学習する ニューラル ネットワークをグラフ化します。 ICLR, 2018 年。

6付録

6.1ハイパーパラメータ

以下のハイパーパラメータは、このホワイトペーパーで紹介するすべての実験で使用されます。

表 1: Rainbow DQN を使用したトレーニングのハイパーパラメーター (4 つのコンポーネント)

ハイパーパラメータ	価値
最適化アルゴリズム	Adam (Kingma & Ba, 2014) 0.00015
学習率	128 0.99 200 オンライン ネットワーク
バッチサイズ	ク アップデート 1
割引率	
DQN対象ネットワーク更新期間	
1 フレームあたりの更新数	50
探査ステップ数	8
N ステップ (マルチステップ) ブートストラップ	0.5
ノイズ ネットσ0	トゥル
DDQN を使用する	- 5000
簡単なタスク: トレーニングのステップ数	
ミディアム タスク: トレーニングのステップ数 50000	
Hard Tasks: トレーニングのステップ数 2000000	

表 2: DOM-Q-NET のハイパーパラメータ

ハイパーパラメータ	価値
語彙サイズ: タグ	80
語彙サイズ: テキスト	400
語彙サイズ :クラス	80
埋め込みディメンション: タグ	16
埋め込みディメンション: テキスト	32
埋め込み次元: クラス	16
完全接続 (FC) 層の次元	128
因子分解された 3 つの Q ネットワークの FC 層の数 各 2	
隠れ層の活性化	履歴書
ニューラル メッセージ パッシングのステップ数 3 (ソーシャル メディア	タスクの場合は 7)
DOMS の最大数 160	
ゴールトークンの最大数 18	
語彙外ランダムベクトル生成	オプションの選択、チェックボックスのクリック

表 3: リブレイ バッファのハイパーパラメータ

ハイパーパラメータ値 α : 重要度サンプリング重みを計算するための優先	
度指数 0.5 β 0 シングル タスク バッファ サイズ 15000 マルチ	タスクが
サイズ 100000	

6.2目標注意出力モデル

図 8 は、ゴールアテンションを使用したグラフ ニューラル ネットワークの読み出しフェーズを示しています。グラフ レベルの特徴ベクトル h_{global} は、T ステップのメッセージ パッシングで処理されたノード レベルの表現 $\{h_1, \dots, h_V\}$ の加重平均によって計算されます。重み $\{\alpha_1, \dots, \alpha_V\}$ は、ゴール ベクトルをクエリとして、ノード レベルの特徴をキーとしてメッセージ パッシングなしで計算されます。DOM-Q-NET では、3.3 に示すように、ローカル埋め込みをキーとして、近隣埋め込みを値として、スケールリングされた内積注意 (Vaswani et al., 2017) を使用します。

GNN の一連の作業で目標注意を組み立て (Scarselli et al., 2009), GNN を目標指向 RL のパラメーター化された状態として使用できるようにします。

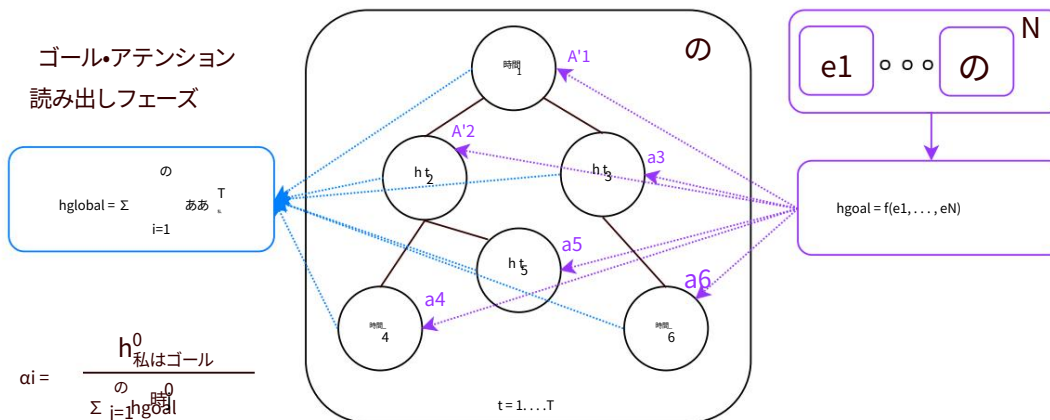


図 8: ゴール・アテンション

6.3 ゴールエンコーダー

グローバルモジュールのための 3 つのタイプのゴールエンコーディングモジュールが調査されます。

1. ノードレベルの機能を使用したゴール ベクトルの連結
2. 6.2 に示すように、目標の注意
3. 図 9 に示すように、目標ベクトルの連結と注意の両方

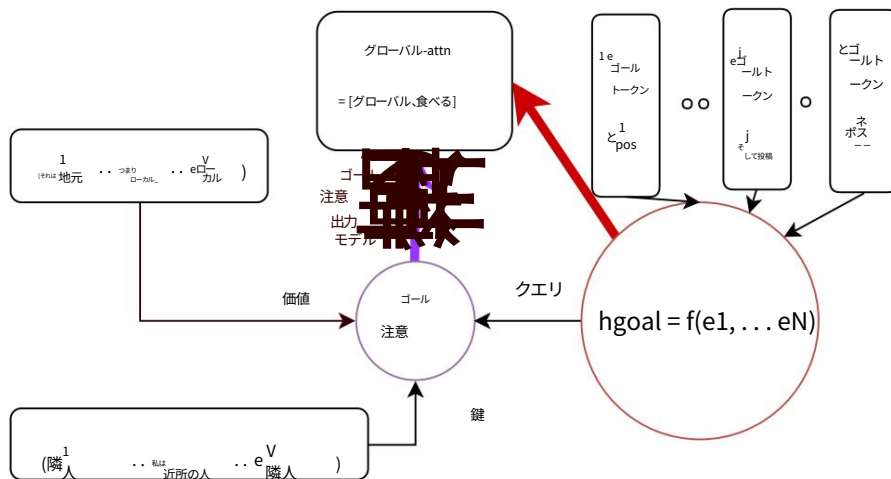


図 9: ゴールエンコーダー

付録 6.7 のマルチタスクと 23 タスクのベンチマーク結果も、さまざまなゴール エンコーディング モジュールを使用した場合のパフォーマンスを比較しています。

6.4 MINIWOB タスクの難しさの定義

- 簡単なタスク: 単一タスク DOM-Q-NET で 5000 タイムステップ未満で解決可能な任意のタスク
{click-dialog, click-test, focus-text, focus-text-2, click-test-2, click-button, click-link, click-button-sequence, click-tab, click-tab-2, Navigate-木}

- ミディアム タスク:単一タスク DOM-Q-NET で 50000 タイムステップ未満で解決可能な任意のタスク
{enter-text, click-widget, click-option, click-checkboxes, enter-text-dynamic, enter password, login-user, email-inbox-delete} •ハード タスク:シングル タスク DOM で 200000 タイムステップ未満で解決可能なタスク-Q-NET,または
エージェントが 100% の成功率に達しないタスク。 {choose-date, search-engine, social-media, email-inbox}

6.5 マルチタスク学習

アルゴリズム 1 共有リプレイ バッファを使用したマルチタスク学習

1:与えられた:

- ポリシー外の RL アルゴリズム A、•複数数のタスクのための一連の環境 K 、

例: DQN, DDPG, NAF, SDQN 例: クリック
チェックボックス、ソーシャル メディア

2:共有リプレイ バッファ R を初期化する
3:共有ネットワーク パラメーター $\theta(k)$ を初期化して A を初期化する
4:各環境の初期化とサンプル s 5: for $i = 1$ to M do for 0
each $k \in K$ do (k) (k)
6:
7: A から行動ポリシーを使用してアクション a をサンプリングします: $a(k) \leftarrow \pi(s) (k)$
8: アクションを実行します t 状態 $s_{t+1}(k)$ (k) (k) a s_t, t, t を観察します $\rightarrow k$ そして、報酬 $r_{t+1}(k)$
遷移を格納 (s R からミニバッチ $t+1$) R で
ッチ B をサンプリングし、 k のエピソードが終了した場合に θ に関して 1 ステップの最適
化を実行し、(k) k をリセットして s をサンプリングする
9:10:11:
12: 0

6.6 実験プロトコル

エージェントが最高のパフォーマンスに収束したら、トレーニングの最後に 100 個のテスト エピソードの成功率を報告します。図 3 で報告されている最終的な成功率は、4 つの異なるランダム シード/実行からの成功率の平均に基づいています。詳しくは、付録 6.4 に示すように、タスクの難易度に応じて一定数のフレームをトレーニングした後、RL エージェントを評価します。表 4 に示すように、このホワイト ペーパーで提示されている結果は、表 1, 2, 3 の一連のハイパーパラメーターに対する合計 536 の実験に基づいています。

表 4: 実験の統計

タスク数	23
マルチタスクの同時実行タスク数	94
比較されたゴール エンコーディング モジュールの数	
$N1 = (23 + 9) * 4 = 128$	
アブレーション研究のタスク数	2
アブレーション研究で比較した割引モデルの数	3
$N2 = 2 * 3 = 6$	
結果の平均を計算するための実験の数 4	
11 マルチタスク学習の実験回数 11	
$N_{total} = (128 + 6 + 11) * 4 = 580$	

6.7 ベンチマーク結果

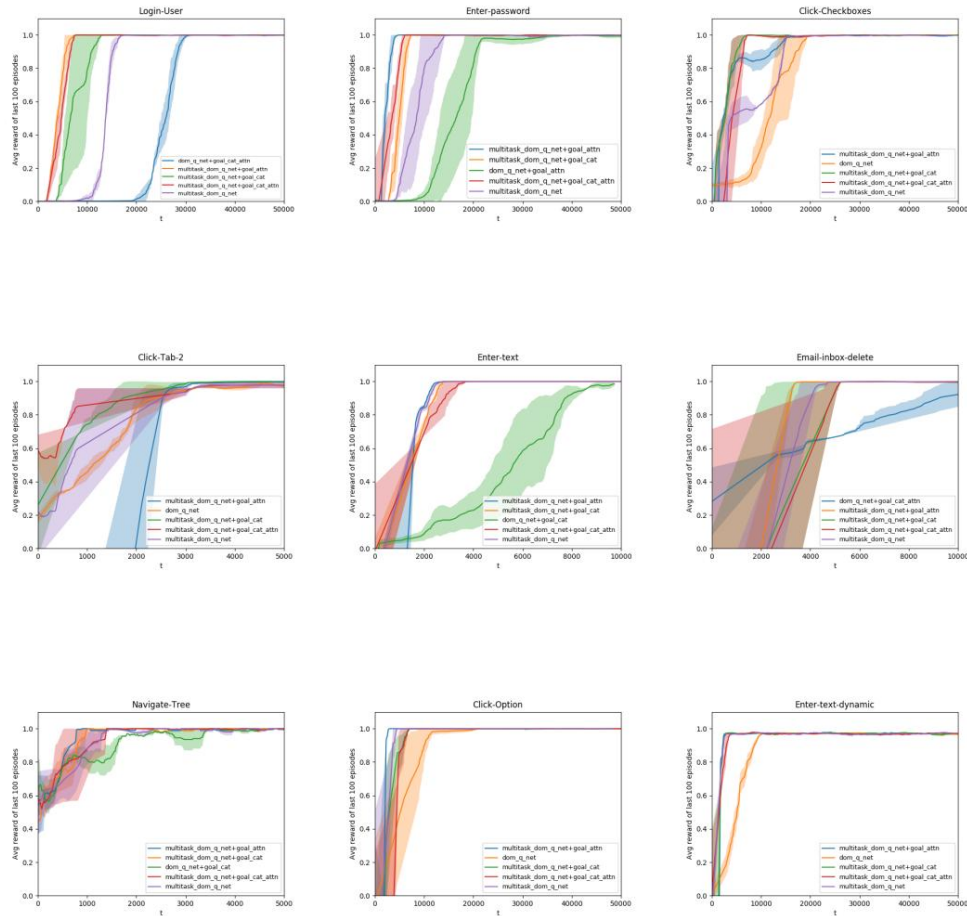
この論文で報告された結果を提供するシングルタスクエージェントとマルチタスクエージェントの両方の学習曲線を提示します。各プロットには、異なるゴール エンコーディング モジュールを持つ DOM-Q-NET エージェントの学習曲線が付随しています。X 軸と Y 軸は、それぞれ過去 100 の報酬のタイムステップと移動平均を表します。ミディアムおよびハード タスクについては、トランジションの割合も示します。

リプレイ バッファ内の正の/ゼロ以外の報酬、およびトレーニング全体でサンプリングされた一意の正の遷移の数。これは、各タスクの報酬の希薄性を実証し、実際の問題が探索の欠如によるものかどうかを調査するためです。

(multistep-bootstrap (Sutton, 1988) を使用していることに注意してください。そのため、直接報酬につながらないいくつかの遷移は、ここでは依然として肯定的であると分析されています)

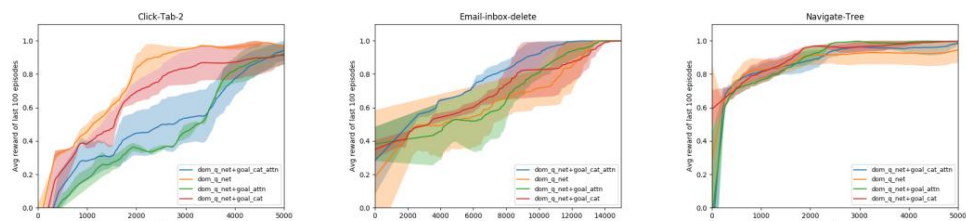
6.7.1 マルチタスク(9タスク)の結果

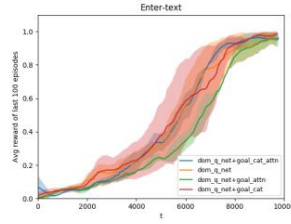
以下は、異なるゴール エンコーダー モジュールを使用したマルチタスクで使用される 9 つのタスクの結果を示しています。



6.7.2 いくつかの簡単なタスクと中程度のタスク

1000 ステップ未満の非常に単純なタスクのプロットは省略されています。





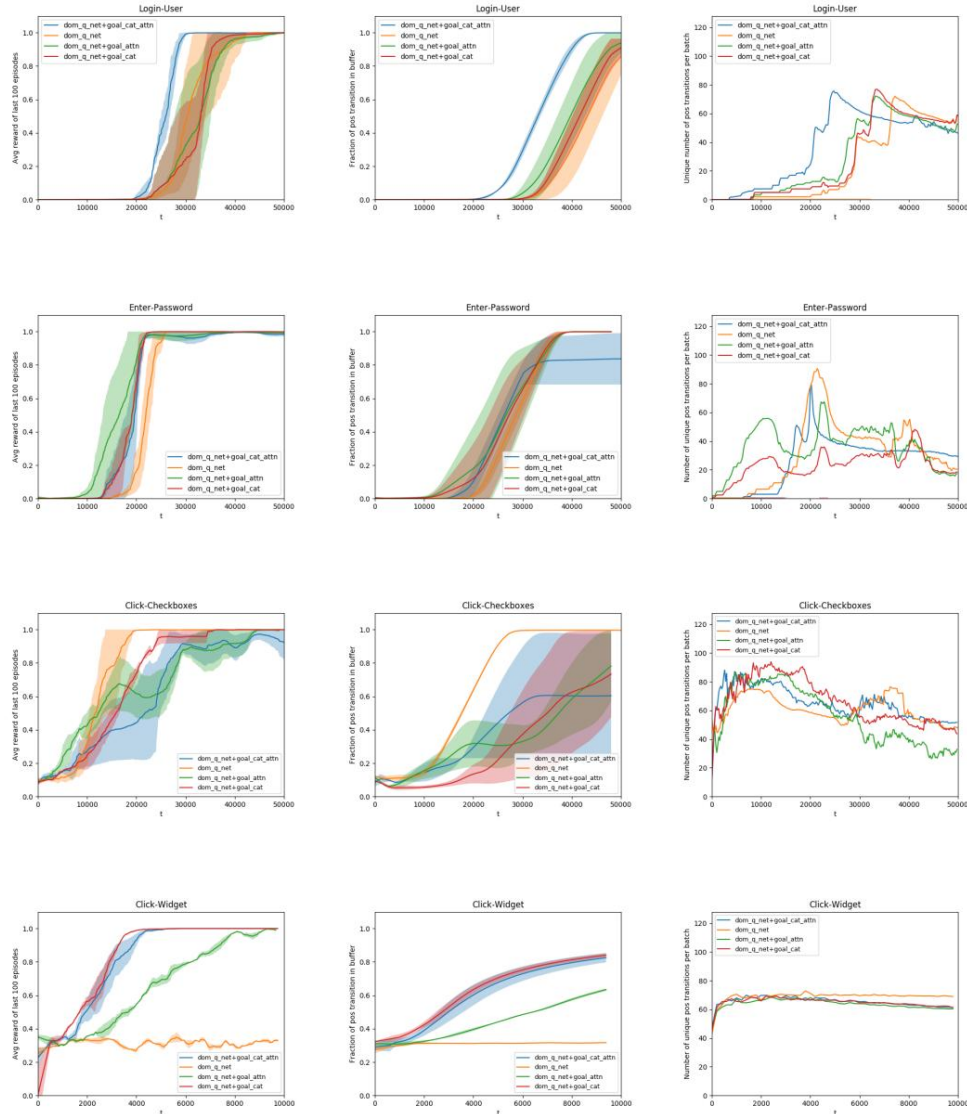
6.7.3 リプレイバッファ情報を含む中程度のタスク

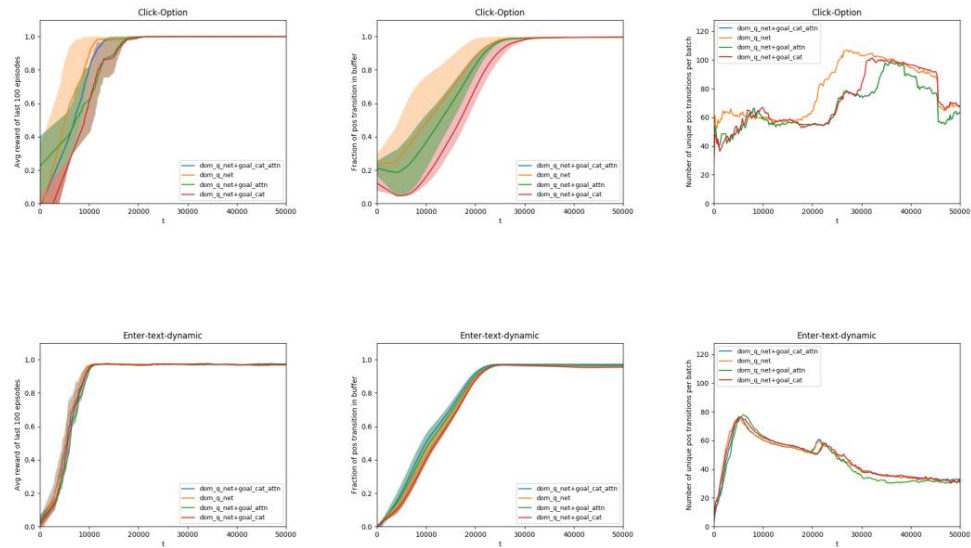
DOM-Q-NET パフォーマンスと DOM-Q-NET + グローバル モジュールのさまざまな目標エンコーディング モジュールのベンチマーク。

左側のプロットは、過去 100 エピソードの平均移動報酬を示しています。

中央のプロットは、リプレイ バッファ内のポジティブ トランジションの割合を示しています。

右側のプロットは、サンプリングごとにサンプリングされた一意の数の正の遷移を示しています。





6.7.4ハードタスク

左側のプロットは、過去 100 エピソードの平均移動報酬を示しています。

中央のプロットは、リプレイ バッファ内のポジティブ トランジションの割合を示しています。

右側のプロットは、サンプリングごとにサンプリングされた一意の数の正の遷移を示しています。

