

# 海馬の記憶インデックスに着想を得たリプレイによる 言語処理タスクの継続学習

前川 在 上垣外 英剛 船越 孝太郎 奥村 学  
東京工業大学

{maekawa, kamigaito, funakoshi, oku}@lr.pi.titech.ac.jp

## 概要

継続学習は過去に学習したタスクへの破滅的忘却を防ぎつつ新しいタスクを学習することを目的とする。その手法の一つである生成的リプレイは過去のタスクのデータを生成モデルにより生成し、新しいタスクのデータに加えることで破滅的忘却を防ぐ。しかし、この手法には生成サンプルの質や偏りによって継続学習の性能が低下するという問題がある。本研究では、脳の高馬の記憶インデックスに着想を得て、学習サンプルの圧縮表現を用いた条件付けにより、過去の特定の学習サンプルをリプレイ時に生成可能な生成的リプレイの手法を提案する。SST, QA-SRL, WOZ の3つの異なる言語処理タスクのデータセットを対象とした実験の結果、提案手法は継続学習における性能の向上を示した。

## 1 はじめに

新しいタスクを次々と学習する継続学習の能力は、人間のような知能システムを実現する上での基本要件の一つである [1]。また、汎用的な継続学習システムの実現は、新しいタスクに対する学習コストを大幅に減らすため、実応用においても重要であると考えられる。しかしながら、ニューラルネットワークで新しいタスクを学習すると、過去に学習したタスクに対する性能が著しく低下する破滅的忘却と呼ばれる現象が起こる [2]。

破滅的忘却は、新しいタスクと同時に過去のタスクを再学習することで対処可能であるが、そのためには過去の全タスクのデータを保持しておく必要があり、ストレージ容量やプライバシーの観点から問題が生じる場合がある。これに対し、実サンプルの代わりに、生成モデルによる擬似サンプルを利用する生成的リプレイという手法がある [3]。一般に生成的リプレイでは、過去の学習サンプルだけでな

く、タスクに相応しい未知のサンプルの生成も想定するが、実際に適切な未知のサンプルを生成することは困難であり、質の低いサンプルを生成する原因になる。また、生成されるサンプルが直前に学習したタスクに偏ってしまうという問題がある。

継続学習を実現している生物の脳では、海馬インデックス理論 [4] によると、海馬が新皮質の活動パターンを圧縮表現に符号化して記憶し、それをインデックス情報として新皮質に投射することで過去の活動パターンが想起されると考えられている。本研究では、この海馬の働きに着想を得て、学習サンプルの圧縮表現による記憶インデックスを用いた生成的リプレイの手法を提案する。各学習サンプルを圧縮表現に符号化する Encoder と、それを記憶インデックスとして保持する海馬モジュールを導入し、記憶インデックスを用いた条件付けによって、リプレイとして過去の特定の学習サンプルを生成することを可能にする。

SST, QA-SRL, WOZ の3つの異なる言語処理タスクのデータセットを対象とした実験の結果、提案手法は生成サンプルの偏りを防ぎ、継続学習の性能を向上させることが可能であることを示した。

## 2 関連研究

ニューラルネットワークを用いた継続学習の研究は大きく、正則化ベース、アーキテクチャベース、リプレイベースの3つの手法に分類される。

正則化ベースの手法では、過去のタスクに対して重要なパラメータが変化しないように制約を加えて破滅的忘却を防ぐ。各パラメータの重要度の推定には、フィッシャー情報行列を利用する手法 [5] や、勾配の大きさを利用する手法 [6] がある。

アーキテクチャベースの手法は、ネットワーク構造の工夫によってタスク間の知識の干渉を防ぐ。タスクごとにネットワークを拡張する手法 [7] や、

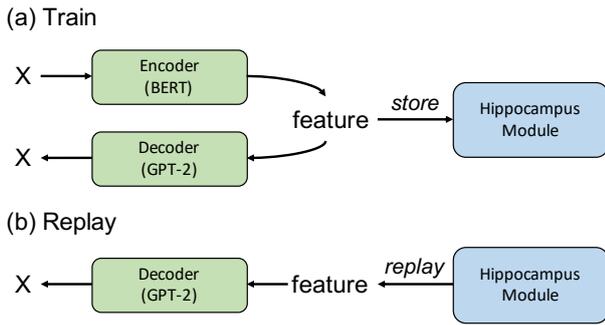


図1 提案手法の概要

フィルタを用いてネットワークをプルーニングする手法 [8] がある。

リプレイベースの手法では、過去に学習したデータを新しいタスクとともに再学習するリプレイを用いる。d’Autume ら [9] は、学習データを保持するエピソード記憶モジュールを導入し、リプレイと推論時のパラメータの局所適応に利用した。Shin ら [3] は、生成モデルによる擬似サンプルを利用する生成的リプレイを提案した。言語処理タスクを対象とした生成的リプレイの手法には、LAMOL [10] がある。LAMOL は、タスクとサンプル生成を言語モデルで同時に学習し、単一のモデルのみで継続学習を行う。LAMOL の派生手法として、知識蒸留を組み合わせた手法 [11, 12] や、生成サンプルのうち忘却度が高いものをリプレイとする手法 [13] がある。

一般に生成的リプレイの手法では、過去の学習サンプルを必要としない一方で、生成サンプルの質や偏りによる継続学習の性能低下が生じる。そこで本研究では、LAMOL における生成サンプルの質や偏りを改善するために、記憶インデックスによる条件付けを用いて、リプレイとして過去の特定の学習サンプルを生成できる手法を提案する。

### 3 提案手法

提案手法の概要を図 1 に示す。提案手法ではタスクの学習と同時に Autoencoder としてサンプル生成を学習する。その際に得られる特徴ベクトルを記憶インデックスとして海馬モジュールで保持し、リプレイ時に条件付けとして Decoder に与えて、対応する学習サンプルを生成する。また、タスク自体も同じ Decoder に学習させる。

#### 3.1 モデル

提案手法は、学習サンプルを符号化する Encoder、タスクとサンプル生成を行う Decoder、記憶イン

デックスを保持する海馬モジュールから構成される。以下では各モジュールの詳細を説明する。

##### 3.1.1 Encoder

BERT [14] を用いて学習サンプルを固定長の特徴ベクトルに符号化する。通常行われる BERT の学習と同様に、先頭につけた [CLS] トークンの埋め込みベクトル  $\mathbf{h}_{[\text{CLS}]} \in \mathbb{R}^H$  を入力全体を表す特徴表現として利用し、重み行列  $\mathbf{W}_E \in \mathbb{R}^{P \times H}$  と活性化関数  $\tanh$  を用いて、 $P$  次元の特徴ベクトル  $\mathbf{h} \in \mathbb{R}^P$  に変換する (式 1)。

$$\mathbf{h} = \tanh(\mathbf{W}_E \mathbf{h}_{[\text{CLS}]}) \quad (1)$$

Encoder のパラメータは、3.2 節で述べる事前学習でのみ最適化を行い、継続学習時は固定する。これは、継続学習の過程で学習時と生成時とで特徴ベクトルにズレが生じ、記憶インデックスによる制御可能性が著しく低下することを防ぐためである。

##### 3.1.2 Decoder

LAMOL [10] のアイデアに基づき、GPT-2 [15] を用いてタスクとサンプル生成を同時に学習する。

提案手法では、サンプル生成時に、Encoder で符号化した特徴ベクトル  $\mathbf{h}$  による条件付けを行う。固定長のベクトルを用いて GPT-2 のテキスト生成を制御する方法については、先行研究 [16, 17] によっていくつかの手法が提案されており、そのうち本研究では、以下で説明するように、Embedding 層と Self-Attention 層を利用する手法を用いる。

**Embedding** 特徴ベクトル  $\mathbf{h}$  から、重み行列  $\mathbf{W}_D \in \mathbb{R}^{H \times P}$  を用いて GPT-2 の入力埋め込みと同じ  $H$  次元のベクトルに変換し、各入力トークンの単語埋め込み  $\mathbf{h}_{\text{WE}}$  と位置埋め込み  $\mathbf{h}_{\text{PE}}$  を足し合わせ、新たな入力埋め込み  $\mathbf{h}_{\text{Emb}} = \mathbf{h}_{\text{WE}} + \mathbf{h}_{\text{PE}} + \mathbf{W}_D \mathbf{h}$  とする。

**Self-Attention** 特徴ベクトル  $\mathbf{h}$  を重み行列  $\mathbf{W}_M \in \mathbb{R}^{LH \times P}$  による線形変換で  $\mathbf{h}_{\text{Mem}} \in \mathbb{R}^{LH}$  に変換し、 $L$  個の  $H$  次元ベクトル  $[\mathbf{h}_{\text{Mem},1}, \dots, \mathbf{h}_{\text{Mem},L}]$  に分割する。ただし、 $L$  は GPT-2 の層数である。各  $l$  層目の Self-Attention において、 $\mathbf{h}_{\text{Mem},l}$  を Key, Value ベクトルに変換して追加の参照情報として利用し、 $x_{<l}$  と  $\mathbf{h}_{\text{Mem},l}$  の両方で条件づける。

##### 3.1.3 海馬モジュール

海馬モジュールは、記憶インデックスとして各学習サンプルの特徴ベクトルを保持する。具体的に

は、新しいタスクの学習時、Encoder で符号化した特徴ベクトルをタスクごとに分けて保存する。リプレイ時に、特徴ベクトルを過去のすべてのタスクから同じ数ずつランダムに取り出して、Decoder のサンプル生成の条件付けに使用する。

### 3.2 事前学習

記憶インデックスとして利用する特徴ベクトルは、Decoder で元の学習サンプルを復元できるだけの十分な情報を含んでいることが望ましい。しかし、BERT の事前学習によって獲得された [CLS] の埋め込みと、ランダムに初期化された  $W_E$  による線形変換では不十分であると考えられる。そこで、テキストコーパスを用いて教師なしで Autoencoder を事前学習することで、十分な情報を持った特徴ベクトルへの符号化を実現する。

また、言語モデルを Autoencoder と同時に学習する。これは、Autoencoder の学習により、GPT-2 が元々持つ言語モデルの情報を失うことを防ぐためである。したがって、事前学習では、Autoencoder の損失と言語モデルの損失の和  $L = L_{AE} + L_{LM}$  を最小化するように、Encoder と Decoder のパラメータを最適化する。

### 3.3 継続学習

#### 3.3.1 データ形式

LAMOL と同様に、decaNLP フォーマット [18] に基づき、Context と Question から Answer を予測する質問応答として、すべてのタスクを Decoder の GPT-2 で学習する (図 2(a))。サンプル生成は、Encoder で Context, Question, Answer 全体を符号化した特徴ベクトルから、元の入力を復元する Autoencoder として学習する (図 2(b))。

#### 3.3.2 学習の流れ

継続学習の問題として、タスク列  $\{T_1, T_2, \dots\}$  を考える。新しいタスク  $T_i$  ( $i > 1$ ) の学習前に、海馬モジュールから特徴ベクトルを  $\gamma|T_i|$  個取り出し、Decoder で対応するサンプルを生成する。ただし、 $|T_i|$  はタスク  $T_i$  の学習サンプルの数であり、 $\gamma$  はリプレイの割合である。生成サンプルのうち、[ANS] が 1 つでないものと、BERT および GPT-2 の最大系列長を超えているものは棄却し、残りをリプレイとして新しいタスクのデータに加える。

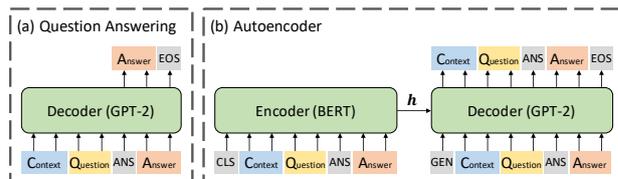


図 2 (a) [ANS] を生成の開始位置を示す特殊トークンとし、質問応答としてタスクを学習。(b) Encoder で符号化した特徴ベクトルによる条件付けを用いて、Decoder で開始トークン [GEN] からサンプル全体を生成。

タスクの損失  $L_{QA}$  と Autoencoder の損失  $L_{AE}$  を、サンプル生成の損失の重み  $\lambda$  を用いて  $L = L_{QA} + \lambda L_{AE}$  として、Decoder を学習する。

## 4 実験

### 4.1 実験設定

**データセット** decaNLP [18] から、感情分析 (SST)、意味役割付与 (QA-SRL)、対話状態追跡 (WOZ) の 3 つの異なるタスクのデータセットを使用し<sup>1)</sup>、提案手法の事前学習には Wiki-40B [19] のテストセットを利用した。

**比較手法** 全手法で GPT-2<sup>2)</sup> を使用し、各タスクを 9 epoch ずつ学習した。以下に各手法を示す。

**提案手法:** Encoder には BERT<sub>BASE</sub><sup>3)</sup> を使用し、サンプル生成の損失の重みは  $\lambda = 0.25$ 、特徴ベクトルの次元数  $P$  は 128 とした。先行研究 [10] に合わせ、リプレイの割合は  $\gamma = 0.05, 0.2$  とした。

**LAMOL:** 提案手法によるリプレイの条件付けの有効性を検証するためのベースライン。言語モデルとして生成を学習し、[GEN] から top- $k$  サンプル ( $k = 20$ ) でリプレイサンプルを生成する。

**Task-specific tokens:** Sun ら [10] が提案した、タスクごとに開始トークンを変え、生成サンプルが過去の全タスクで均等になるよう制御する手法。提案手法と LAMOL のそれぞれに適用した。

**Real sample:** 提案手法のリプレイを実サンプルに置き換える。提案手法の性能上限と考えられる。

**Fine-tuning:** タスクを 1 つずつファインチューニングする。継続学習の性能下限と考えられる。

**Multitask:** 全タスクのデータセットを混合し同時に学習する。継続学習の性能上限と考えられる。

**評価指標** 継続学習の性能評価のために、以下の 2 つの評価指標を用いる。ここで  $a_{j,i} \in [0, 100]$  はタ

1) 各タスクの詳細は付録の表 2 を参照。

2) <https://huggingface.co/gpt2>

3) <https://huggingface.co/bert-base-uncased>

**表 1** 各手法の全 6 パターンの学習順序に対する実験結果とそれらの平均及び標準偏差. 各スコアは 2 回実行した結果の平均. Proposal および LAMOL の上付き文字はリプレイの割合  $\gamma$  を表し, 下付き文字はそれぞれ共通の開始トークン (GEN), Task-specific tokens (TASK), Real sample (REAL) を表す.

| Methods                                  | SST SRL WOZ                          |                   | SST WOZ SRL     |                   | SRL SST WOZ     |                   | SRL WOZ SST     |                   | WOZ SST SRL     |                   | WOZ SRL SST     |                   | Average         |                   | Std  |      |
|--|--------------------------------------|-------------------|-----------------|-------------------|-----------------|-------------------|-----------------|-------------------|-----------------|-------------------|-----------------|-------------------|-----------------|-------------------|------|------|
|  | Avg. $\uparrow$                      | Fgt. $\downarrow$ | Avg. $\uparrow$ | Fgt. $\downarrow$ | Avg. $\uparrow$ | Fgt. $\downarrow$ | Avg. $\uparrow$ | Fgt. $\downarrow$ | Avg. $\uparrow$ | Fgt. $\downarrow$ | Avg. $\uparrow$ | Fgt. $\downarrow$ | Avg. $\uparrow$ | Fgt. $\downarrow$ | Avg. | Fgt. |
| Fine-tuning                              | 45.5                                 | 54.9              | 26.2            | 83.5              | 63.9            | 26.5              | 32.6            | 73.5              | 44.2            | 56.3              | 38.8            | 64.5              | 41.9            | 59.9              | 11.9 | 17.9 |
| LAMOL <sub>GEN</sub> <sup>0.05</sup>     | 78.1                                 | 5.8               | 75.5            | 10.2              | 71.1            | 15.4              | 72.5            | 13.6              | 66.7            | 22.0              | 73.4            | 11.9              | 72.9            | 13.1              | 3.6  | 5.1  |
| LAMOL <sub>TASK</sub> <sup>0.05</sup>    | 77.9                                 | 6.0               | 73.3            | 12.2              | 76.9            | 7.2               | 75.6            | 8.4               | 65.5            | 23.9              | 74.0            | 11.0              | 73.9            | 11.4              | 4.1  | 6.1  |
| Proposal <sub>GEN</sub> <sup>0.05</sup>  | 79.7                                 | 3.3               | 77.8            | 6.3               | 78.7            | 5.6               | 78.1            | 5.7               | 75.9            | 9.4               | 77.0            | 7.7               | 77.8            | 6.3               | 1.3  | 2.0  |
| Proposal <sub>TASK</sub> <sup>0.05</sup> | 79.1                                 | 4.5               | 77.6            | 6.4               | 78.1            | 5.7               | 77.7            | 6.8               | 75.8            | 9.6               | 76.5            | 8.7               | 77.5            | 6.9               | 1.1  | 1.9  |
| LAMOL <sub>GEN</sub> <sup>0.2</sup>      | 79.6                                 | 3.7               | 79.5            | 3.7               | 78.0            | 4.6               | 78.2            | 4.6               | 77.4            | 5.8               | 77.6            | 5.8               | 78.4            | 4.7               | 0.9  | 0.9  |
| LAMOL <sub>TASK</sub> <sup>0.2</sup>     | 79.7                                 | 3.3               | 79.3            | 3.9               | 78.3            | 4.5               | 78.3            | 4.3               | 77.9            | 5.6               | 77.3            | 6.2               | 78.5            | 4.6               | 0.9  | 1.1  |
| Proposal <sub>GEN</sub> <sup>0.2</sup>   | 80.2                                 | 2.1               | 79.4            | 3.9               | 79.4            | 3.5               | 78.8            | 4.8               | 79.7            | 3.6               | 79.6            | 3.8               | 79.5            | 3.6               | 0.7  | 1.1  |
| Proposal <sub>TASK</sub> <sup>0.2</sup>  | 80.0                                 | 2.5               | 80.2            | 3.2               | 80.0            | 2.8               | 79.1            | 4.3               | 79.6            | 3.9               | 80.1            | 3.1               | 79.8            | 3.3               | 0.4  | 0.7  |
| Proposal <sub>REAL</sub> <sup>0.05</sup> | 79.8                                 | 3.0               | 76.4            | 8.2               | 78.6            | 4.8               | 78.7            | 5.1               | 76.6            | 8.0               | 77.8            | 6.5               | 78.0            | 5.9               | 1.3  | 2.0  |
| Proposal <sub>REAL</sub> <sup>0.2</sup>  | 80.8                                 | 1.6               | 79.9            | 2.9               | 80.4            | 2.5               | 79.8            | 2.7               | 80.2            | 2.5               | 80.6            | 2.3               | 80.3            | 2.4               | 0.6  | 0.7  |
| Multitask                                | 81.8 (SST: 90.8 SRL: 70.1 WOZ: 84.5) |                   |                 |                   |                 |                   |                 |                   |                 |                   |                 |                   |                 |                   |      |      |



**図 3** 2 回目のリプレイに含まれる各タスクの生成サンプルの割合

スク  $T_i$  を学習後のタスク  $T_j$  の評価結果を表す.

**Average Score**  $\in [0, 100]$ : 最終的なモデルの各タスクの評価結果  $a_{T,i}$  の算術平均.

**Forgetting Measure**  $\in [-100, 100]$ : Chaudhry ら [20] が提案した, 過去のタスクの忘却度合い. ただし, decaNLP タスクの評価指標に合わせて  $-100 \sim 100$  の値をとる. 値が小さいほど望ましい.

$$F_T = \frac{1}{T-1} \sum_{i=1}^{T-1} f_{T,i} \quad (f_{T,i} = \max_{j < T} a_{j,i} - a_{T,i})$$

## 4.2 実験結果

6 パターンすべての学習順序について実験を行った結果を表 1 に示す. まず, Fine-tuning の結果から, 新しいタスクの学習によって, 過去のタスクに対する破滅的忘却が生じることが確認できる. 提案手法の結果は, すべての条件で LAMOL の性能を大きく上回っており, タスクの学習順序に対する性能のばらつきも抑えられている. これは, 継続学習に対して提案手法が有効であることを示している.

各実験における 2 回目のリプレイ時の生成サン

プルをタスクごとに分類<sup>4)</sup>した結果を図 3 に示す. LAMOL では生成サンプルの多くが直前のタスクのものになっており, Task-specific tokens の効果も学習順序によって大きく差がある. これに対して, 提案手法は生成サンプルの偏りを大幅に抑えられており, 記憶インデックスによる条件付けがうまく機能していると考えられる. また, リプレイの偏りがほとんど同じ場合についても, 継続学習の性能で提案手法が LAMOL を上回っていることから, 生成サンプルの質が改善されていることが示唆される.

## 5 おわりに

本稿では, 言語処理タスクの継続学習に対して, 海馬の記憶インデックスに着想を得た制御可能な生成リプレイの手法を提案した. 複数の異なる言語処理タスクを対象とした実験の結果, 提案手法は生成サンプルの偏りを防ぎ, 継続学習の性能を向上させることを示した. 今後は, 海馬モジュールの記憶容量を削減する方法についても取り組みたい.

4) BERT による分類モデルを作成して使用した.

## 参考文献

- [1] German I Parisi, Ronald Kemker, Jose L Part, Christopher Kanan, and Stefan Wermter. Continual lifelong learning with neural networks: A review. **Neural Networks**, Vol. 113, pp. 54–71, 2019.
- [2] Michael McCloskey and Neal J. Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. **Psychology of Learning and Motivation**, Vol. 24, pp. 109–165, 1989.
- [3] Hanul Shin, Jung Kwon Lee, Jaehong Kim, and Jiwon Kim. Continual learning with deep generative replay. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, **Advances in Neural Information Processing Systems**, Vol. 30, pp. 2990–2999. Curran Associates, Inc., 2017.
- [4] Timothy J Teyler and Jerry W Rudy. The hippocampal indexing theory and episodic memory: updating the index. **Hippocampus**, Vol. 17, No. 12, pp. 1158–1169, 2007.
- [5] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, and Raia Hadsell. Overcoming catastrophic forgetting in neural networks. **Proceedings of the National Academy of Sciences**, Vol. 114, No. 13, pp. 3521–3526, 2017.
- [6] Rahaf Aljundi, Francesca Babiloni, Mohamed Elhoseiny, Marcus Rohrbach, and Tinne Tuytelaars. Memory aware synapses: Learning what (not) to forget. In **Proceedings of the European Conference on Computer Vision (ECCV)**, pp. 139–154, 2018.
- [7] Andrei A Rusu, Neil C Rabinowitz, Guillaume Desjardins, Hubert Soyer, James Kirkpatrick, Koray Kavukcuoglu, Razvan Pascanu, and Raia Hadsell. Progressive neural networks. **arXiv preprint arXiv:1606.04671**, 2016.
- [8] Arun Mallya and Svetlana Lazebnik. Packnet: Adding multiple tasks to a single network by iterative pruning. In **Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)**, pp. 7765–7773, June 2018.
- [9] Cyprien de Masson d'Autume, Sebastian Ruder, Lingpeng Kong, and Dani Yogatama. Episodic memory in lifelong language learning. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, **Advances in Neural Information Processing Systems**, Vol. 32, pp. 13122–13131. Curran Associates, Inc., 2019.
- [10] Fan-Keng Sun, Cheng-Hao Ho, and Hung-Yi Lee. Lamol: Language modeling for lifelong language learning. In **International Conference on Learning Representations**, 2019.
- [11] Yung-Sung Chuang, Shang-Yu Su, and Yun-Nung Chen. Lifelong language knowledge distillation. In **Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)**, pp. 2914–2924, Online, November 2020. Association for Computational Linguistics.
- [12] Jingyuan Sun, Shaonan Wang, Jiajun Zhang, and Chengqing Zong. Distill and replay for continual language learning. In **Proceedings of the 28th International Conference on Computational Linguistics**, pp. 3569–3579, Barcelona, Spain (Online), December 2020. International Committee on Computational Linguistics.
- [13] Heejeong Choi and Pilsung Kang. Lifelong language learning with the most forgotten knowledge. **IEEE Access**, Vol. 9, pp. 57941–57948, 2021.
- [14] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In **Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)**, pp. 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.
- [15] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. **OpenAI blog**, Vol. 1, No. 8, p. 9, 2019.
- [16] Chunyuan Li, Xiang Gao, Yuan Li, Baolin Peng, Xiujuan Li, Yizhe Zhang, and Jianfeng Gao. Optimus: Organizing sentences via pre-trained modeling of a latent space. In **Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)**, pp. 4678–4699, Online, November 2020. Association for Computational Linguistics.
- [17] Le Fang, Tao Zeng, Chaochun Liu, Liefeng Bo, Wen Dong, and Changyou Chen. Transformer-based conditional variational autoencoder for controllable story generation. **arXiv preprint arXiv:2101.00828**, 2021.
- [18] Bryan McCann, Nitish Shirish Keskar, Caiming Xiong, and Richard Socher. The natural language decathlon: Multitask learning as question answering. **arXiv preprint arXiv:1806.08730**, 2018.
- [19] Mandy Guo, Zihang Dai, Denny Vrandečić, and Rami Al-Rfou. Wiki-40B: Multilingual language model dataset. In **Proceedings of the 12th Language Resources and Evaluation Conference**, pp. 2440–2452, Marseille, France, May 2020. European Language Resources Association.
- [20] Arslan Chaudhry, Puneet K Dokania, Thalaiyasingam Ajanthan, and Philip HS Torr. Riemannian walk for incremental learning: Understanding forgetting and intransigence. In **Proceedings of the European Conference on Computer Vision (ECCV)**, pp. 532–547, 2018.

## A 付録

### A.1 継続学習に使用したタスクの詳細

表 2 使用したタスク，データセット，評価指標

| Task                   | Dataset | Context   | Question                             | Answer         | #Train | #Test | Metric |
|------------------------|---------|---|--------------------------------------|----------------|--------|-------|--------|
| Sentiment Analysis     | SST     | It's a very valuable film...                                    | Is this review negative or positive? | positive       | 6920   | 1821  | EM     |
| Semantic Role Labeling | QA-SRL  | COG is a 6-piece metal band based in Metro Manila, Philippines. | Who is based somewhere?              | COG            | 6414   | 2201  | nF1    |
| Goal-Oriented Dialogue | WOZ     | I am looking for African food                                   | What is the change in state ?        | food: African; | 2536   | 1646  | dsEM   |

### A.2 忘却の過程

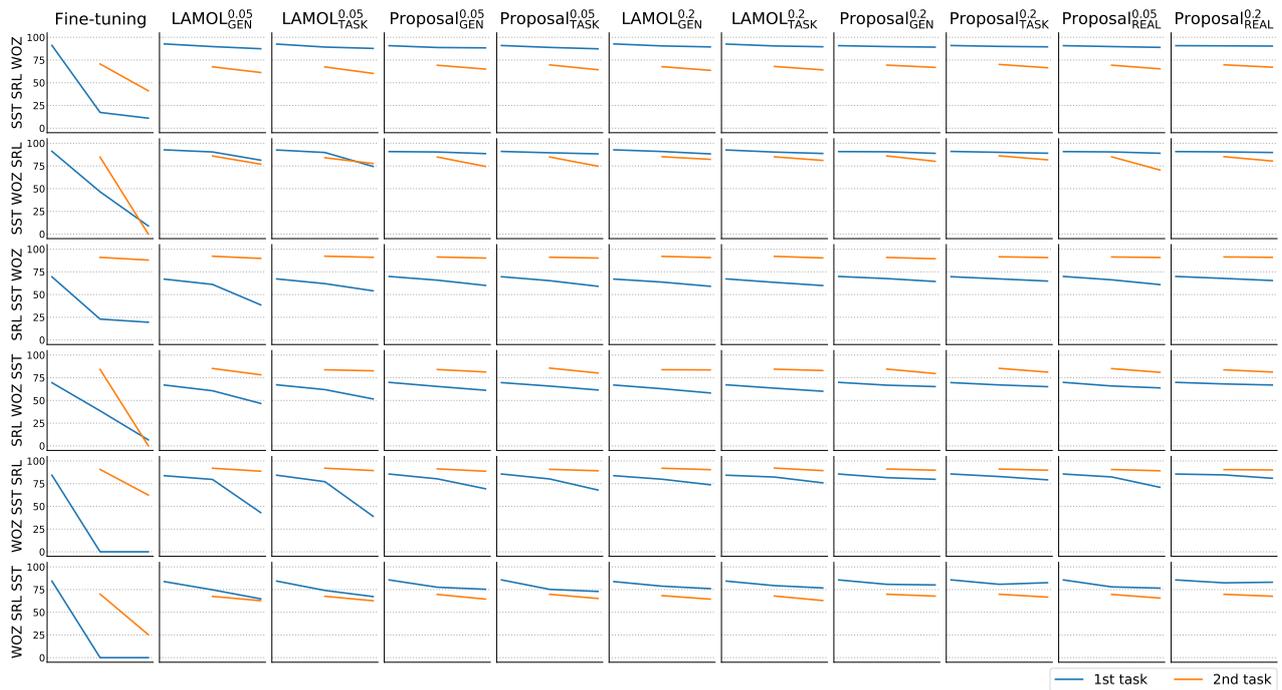


図 4 それぞれの手法の各学習順序における忘却の過程