# 逆学習による言語モデルの解析

磯沼 大 [1,2]　　イヴァン チトフ [1]
[1] エディンバラ大学　[2] 東京大学
m.isonuma@ed.ac.uk　ititov@inf.ed.ac.uk

## 概要

大規模言語モデルの高い能力を発揮させ、かつ有害な生成を抑えるには、その原因となる学習データの特定が不可欠である。理想的には、各学習データを除いて再学習したモデルを評価することで各データの影響を測ることができるが、計算コストが膨大になり現実的ではない。本稿では学習データを除く代わりに、学習済モデルから各データを逆学習してその影響を測る手法を提案する。本手法は極めて単純で、勾配上昇法で学習データを逆学習し、逆学習後のモデルの性能が悪化するほど、その学習データの影響が大きいと推定する。本手法は既存手法に比べ膨大なメモリやチェックポイントの保存を必要とせず、かつ評価実験で高い性能を示した。

## 1　はじめに

大規模言語モデル（LLM）は驚異的な汎化・推論能力を示し、社会に衝撃を与えている。一方で、差別やバイアスを含む文章を生成するなど、その安全性もまた憂慮されている。LLM の高い能力を引き出しつつ、有害性を除去するには、それらの原因となる学習データの特定が不可欠である。しかし LLM は大量のコーパスで学習されており、どの学習データに起因するのか特定するには困難を極める。

ある評価データに対する学習データの影響を知るための理想的な方法は、各学習データを除いてモデルを再学習し、その評価データにおける性能変化を測ることである（leave-one-out）。しかし、この方法は計算コストが膨大で現実的ではない。そこで学習データの影響を推定する手法として、Hessian-based influence functions (HIF) [1, 2, 3] や、TracIn [4] が提案されてきた。しかし HIF はヘッセ行列の逆行列計算に多大な計算量を要し、TracIn もまた多数のチェックポイントを必要とする。また、これらの手法は一次近似を伴うため、大量のデータを学習したモデルには有効でないことが報告されている [5, 6, 7]。

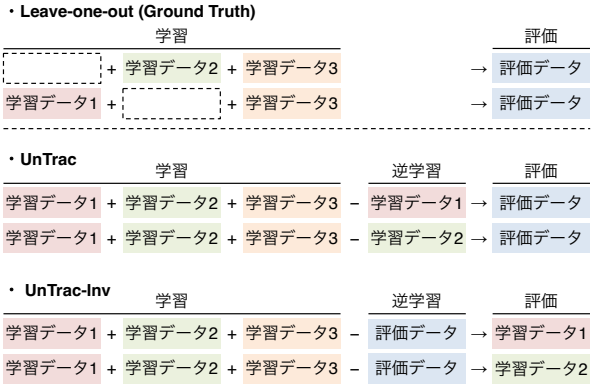


**図 1**　提案する UnTrac/UnTrac-Inv と leave-one-out の対比。

本稿では、学習データを除く代わりに、学習済モデルから逆学習して、学習データの影響を推定する UnTrac を提案する（図 1）。Leave-one-out は、学習データを除く前後でのモデルの性能を比較し、その影響を推定する。同様に、UnTrac は学習データを逆学習し、逆学習前後でのモデルの性能を比較して、その影響を推定する。逆学習はモデルから有害な学習データを忘却させる方法として注目されており [8, 9]、本稿はこれを学習データの影響推定に応用する。本手法で要する計算量は通常の学習と同等で、HIF や TracIn に必要な多大な計算量やチェックポイントを要しない。また一次近似を伴わないため、大量のデータを学習する場合にも有効である。

UnTrac は、複数の学習データの影響を測る場合、その数だけ逆学習を実行しなくてはならず、計算コストが嵩む。そこで、本研究ではよりスケーラブルな方法として UnTrac-Inv を提案する。UnTrac-Inv は学習データの代わりに評価データを逆学習し、逆学習前後のモデルを学習データで評価する。UnTrac-Inv は幾つかの仮定・条件のもとで UnTrac と等価である。また、UnTrac と UnTrac-Inv は、HIF などの既存手法の一般化としてみなせることを示す。

実験では、複数のモデル（エンコーダデコーダ/デコーダ）及び実験設定（事前学習/ファインチューニング）にて、提案手法の有効性を検証する。

## 2 問題設定

本研究の目標は、ある評価データセットにおけるモデルの性能に対し、学習データセットが与える影響を推定することである。これを定式化するため、全データセットで学習されたモデル $\boldsymbol{\theta}_0$ と、ある学習データセット $Z$ を抜いて学習されたモデル $\boldsymbol{\theta}_{-Z}$ を考える（leave-one-out）。学習データセット $Z$ が評価データセット $Z$ に与える実際の影響値を式 (1) で定義する。各学習データセット $Z$ について、この影響値の相対的な大きさを推定することを目指す。

$$I_{\text{truth}}(Z', Z) = \sum_{j=1}^{N'} L(z'_j, \boldsymbol{\theta}_0) - L(z'_j, \boldsymbol{\theta}_{-Z}) \tag{1}$$

ここで、$z'_j$ は評価データセット $Z'$ 中のバッチを、$N'$ はバッチの数を、$L$ は損失関数を表す。本稿では、$\boldsymbol{\theta}_{-Z}$ の学習に際し、どのデータセットを抜く場合でも同じ学習ステップ数を用いる。以下では逆学習により $I_{\text{truth}}$ を推定する方法を詳説する。

## 3 提案手法

### 3.1 UnTrac

学習済モデルのパラメータを $\boldsymbol{\theta}_0$、$i$ ステップ逆学習後のモデルを $\boldsymbol{\theta}_i$ と表す。UnTrac は、学習データセット $Z$ の評価データセット $Z'$ に対する影響を、評価データセットの損失関数 $L$ の差分で推定する:

$$\begin{aligned} I(Z', Z) &= \sum_{j=1}^{N'} L(z'_j, \boldsymbol{\theta}_T) - L(z'_j, \boldsymbol{\theta}_0) \\ &= \sum_{j=1}^{N'} \sum_{i=1}^{T} L(z'_j, \boldsymbol{\theta}_i) - L(z'_j, \boldsymbol{\theta}_{i-1}) \end{aligned} \tag{2}$$

ただし $T$ は逆学習のステップ数である。既存研究を踏襲し [8]、$\boldsymbol{\theta}_i$ は勾配上昇法により、学習データセット $Z$ 中のバッチ $z_i$ の損失関数を最大化するように更新される。確率的勾配法 (SGD) を用いた場合の更新式は学習率 $\eta$ を用いて式 (3) で表される。

$$\boldsymbol{\theta}_i = \boldsymbol{\theta}_{i-1} + \eta \nabla_{\boldsymbol{\theta}} L(z_i, \boldsymbol{\theta}_{i-1}) \tag{3}$$

### 3.2 UnTrac-Inv

実務では、ある特定の評価データセットに対して、どの学習データセットの影響が大きいか関心を持つ場合が多い。本節では、学習データセットの数に対してスケーラブルな方法として、UnTrac-Inv を導入する。UnTrac-Inv は学習データセットの代わりに評価データセットを逆学習し、学習データセットにて損失関数の差分を測ることで影響を推定する：

$$\begin{aligned} I'(Z', Z) &= \sum_{i=1}^{N} L(z_i, \boldsymbol{\theta}_{T'}) - L(z'_j, \boldsymbol{\theta}_0) \\ &= \sum_{i=1}^{N} \sum_{j=1}^{T'} L(z_i, \boldsymbol{\theta}_j) - L(z_i, \boldsymbol{\theta}_{j-1}) \end{aligned} \tag{4}$$

ここで $N$ は学習データセット内のバッチ数を、$T'$ は逆学習のステップ数を表す。

いくつかの仮定・条件の下、UnTrac-Inv と UnTrac は等価であるとみなせる。式 (3) 及び一次近似 $L(z, \boldsymbol{\theta}_i) - L(z, \boldsymbol{\theta}_{i-1}) \approx \nabla_{\boldsymbol{\theta}} L(z, \boldsymbol{\theta}_{i-1})^\top (\boldsymbol{\theta}_i - \boldsymbol{\theta}_{i-1})$ により、式 (2) と式 (4) はそれぞれ下記のように近似できる。

$$I(Z', Z) \approx \sum_{i=1}^{T} \sum_{j=1}^{N'} \eta \nabla_{\boldsymbol{\theta}} L(z_i, \boldsymbol{\theta}_{i-1})^\top \nabla_{\boldsymbol{\theta}} L(z'_j, \boldsymbol{\theta}_{i-1}) \tag{5}$$

$$I'(Z', Z) \approx \sum_{i=1}^{N} \sum_{j=1}^{T'} \eta \nabla_{\boldsymbol{\theta}} L(z_i, \boldsymbol{\theta}_{j-1})^\top \nabla_{\boldsymbol{\theta}} L(z'_j, \boldsymbol{\theta}_{j-1}) \tag{6}$$

逆学習のステップ数が 1 で ($T = T' = 1$)、バッチがデータセット内の全サンプルを含む時、$I(Z', Z)$ は $I'(Z', Z)$ に一次近似の下で一致する。即ち、逆学習のステップ数が少なく、バッチサイズが大きい場合、UnTrac-Inv は有効であることが示唆される。

### 3.3 既存手法との関連

**TracIn, GradDot & GradCos** TracIn [4] は各チェックポイント $t \in \mathcal{T}_{cp}$ に保存されたパラメータ $\boldsymbol{\theta}_t$ を用いて、学習データセットの影響を式 (7) で推定する。

$$\text{TracIn}(Z', Z) = \sum_{t \in \mathcal{T}_{cp}} \sum_{i=1}^{N} \sum_{j=1}^{N'} \eta \nabla_{\boldsymbol{\theta}} L(z_i, \boldsymbol{\theta}_t)^\top \nabla_{\boldsymbol{\theta}} L(z'_j, \boldsymbol{\theta}_t) \tag{7}$$

最終チェックポイントのみ（学習済モデルのパラメータ $\boldsymbol{\theta}_0$）が使われることも多く、GradDot と呼ばれる。また、学習データが外れ値の場合は勾配が大きくなるため、勾配を正規化した（即ち内積をコサイン類似度で置換した）GradCos も頻用される。

**Hessian-based influence Functions** HIF [2] は、学習済モデル $\boldsymbol{\theta}_0$ について、影響を式 (8) で推定する。

$$\text{HIF}(Z', Z) = \sum_{i=1}^{N} \sum_{j=1}^{N'} \nabla_{\boldsymbol{\theta}} L(z_i, \boldsymbol{\theta}_0)^\top \boldsymbol{H}^{-1} \nabla_{\boldsymbol{\theta}} L(z'_j, \boldsymbol{\theta}_0) \tag{8}$$

ここで、$\boldsymbol{H}$ は学習データの損失関数に関するヘッセ行列である：$\boldsymbol{H} = 1/N \sum_{i=1}^{N} \nabla^2_{\boldsymbol{\theta}} L(z_i, \boldsymbol{\theta}_0)$。ヘッセ行列の逆行列計算には莫大な計算量を要するため、本稿では LISSA [2] 及び Arnoldi [10] により近似する。

**表 1**　ファインチューニングに用いるデータセットのサンプル例。波括弧 {} で囲まれた部分はサンプル毎に異なる。

| データセット | 入力 | 出力 |
|---|---|---|
| 評価 | What is the number that comes after {2}? | {3} |
| 学習 1（類似タスク/同じ出力形式） | Determine the number that succeeds {zero}. Provide your answer in numerical form. | {1} |
| 学習 2（類似タスク/異なる出力形式） | Determine the number that succeeds {one}. Provide your answer in words. | {two} |
| 学習 3（非類似タスク/同じ出力形式） | Determine the length of '{friend}'. Provide your answer in numerical form. | {6} |
| 学習 4（非類似タスク/異なる出力形式） | Determine the length of '{problem}'. Provide your answer in words. | {seven} |

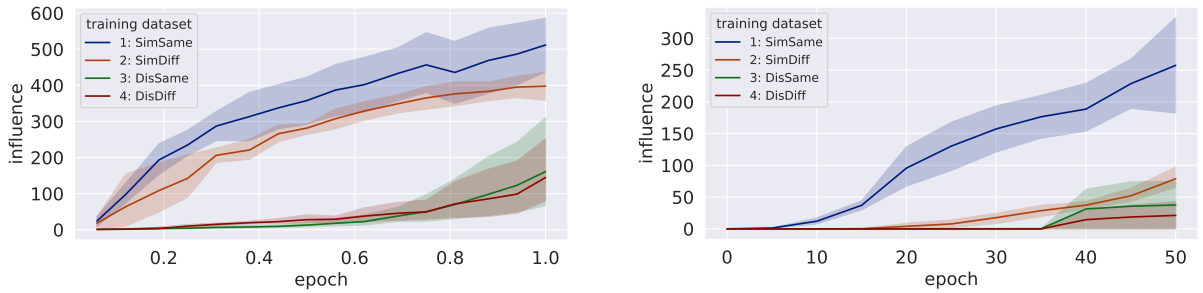


**図 2**　UnTrac により推定された影響値（左）と UnTrac-Inv により推定された影響値（右）。横軸は逆学習のエポック数。

**逆学習に基づく解釈**　GradDot、GradCos 及び HIF は、UnTrac の特殊ケースとみなせる。UnTrac により 1 ステップで学習データセットの全サンプルを逆学習した時、式 (2) は一次近似で下記のように表せる：

$$\begin{aligned} I(Z', Z) &= \sum_{j=1}^{N'} L(z'_j, \theta_1) - L(z'_j, \theta_0) \\ &\approx \sum_{j=1}^{N'} \nabla_\theta L(z'_j, \theta_0)^\top (\theta_1 - \theta_0) \end{aligned} \tag{9}$$

SGD を用いた場合、$\theta_1 - \theta_0 = \eta \sum_{i=1}^{N} \nabla_\theta L(z_i, \theta_0)$ より、式 (9) は GradDot と一致する。同様に、RMSProp や Adam を用いた場合は GradCos と一致し、ニュートン法を用いた場合は HIF と一致する。同様の議論は UnTrac-Inv についても成り立つ。

# 4　評価実験

本節では、逆学習により学習データの影響を推定できるか検証する。学習データの影響推定では、検証データは通常存在しない。従って、本実験では 4.2 節の Toxigen データセットによる検討に基づき、同一のハイパーパラメータを他のデータセットでも使用した。勾配法は Adam [11]（学習率：5e-5、$\beta_1 = 0.9$、$\beta_2 = 0.999$）を用いた。UnTrac のバッチサイズは 1 で、1 エポック逆学習を行った。3.2 節の議論から、UnTrac-Inv はステップ数が少なく、バッチサイズが大きい場合に有効であることが示唆される。従って、バッチサイズは 256（評価データセットサイズ）に設定し、ステップ数（エポック数）は 5 に設定した。4.2 節にてこれらの影響を議論する。

## 4.1　ファインチューニングの影響推定

まず、事前学習モデルを複数のタスクでファインチューニングした場合に、学習タスクの影響を正確に推定できるか検証する。本手法に対する懸念として、評価タスクと出力形式は同じだが異なるタスクを逆学習すると、逆学習後のモデルはいかなる入力に対しても当該形式で出力しなくなり、当該タスクの影響が過大評価されることを危惧した。そこで、本実験では評価タスクと類似する/しない、出力形式が同じ/異なる 4 つの学習タスクを用意し、各学習タスクの影響を適切に推定できるか検証する。

**モデル**　本実験では事前学習済エンコーダデコーダモデルの T5 [12]（30 億パラメータ）を使用する。

**データセット**　表 1 に示すように、評価データセットと、それに対応する 4 つの学習データセットを作成した。学習/評価データセットはそれぞれ 256 サンプルで構成される。この 4 つの学習データセットを混合し、モデルを 1 エポック学習した。

**結果**　図 2 は UnTrac（左）及び UnTrac-Inv（右）について、逆学習を進めたときの影響値の変化である（4 回試行した平均と 95%信頼区間を表示）。両手法とも、類似タスク（1, 2）の影響は非類似タスク（3, 4）より高く推定された。実際に leave-one-out においても、データセット 1 と 2 の実際の影響は 3 と 4 より大きいことを確認している（付録 A.1 節）。従って、UnTrac と UnTrac-Inv はタスクの類似性に基づいて適切に学習データの影響を推定でき、出力形式の類似性に大きく影響されないことがわかる。

**表 2** 各手法により推定された影響と実際の影響とのピアソン相関係数。4 回の試行における平均と標準偏差を表示。

| | 各学習データセットサイズが均一 | | | 各学習データセットサイズが不均一 | | |
|---|---|---|---|---|---|---|
| 推定手法 | ToxiGen | WinoBias | TruthfulQA | ToxiGen | WinoBias | TruthfulQA |
| GradDot | -0.123 ± 0.008 | 0.418 ± 0.018 | 0.156 ± 0.022 | -0.250 ± 0.007 | 0.446 ± 0.015 | -0.524 ± 0.003 |
| GradCos | -0.050 ± 0.008 | 0.524 ± 0.014 | 0.447 ± 0.015 | -0.337 ± 0.007 | 0.496 ± 0.012 | -0.401 ± 0.004 |
| HIF (Arnoldi) | -0.068 ± 0.023 | 0.559 ± 0.010 | 0.250 ± 0.024 | -0.343 ± 0.005 | 0.584 ± 0.014 | -0.362 ± 0.006 |
| HIF (LISSA) | -0.040 ± 0.328 | 0.389 ± 0.117 | -0.178 ± 0.173 | 0.071 ± 0.091 | -0.092 ± 0.042 | -0.098 ± 0.058 |
| TracIn | 0.207 ± 0.010 | 0.082 ± 0.013 | 0.591 ± 0.014 | -0.187 ± 0.005 | 0.183 ± 0.019 | 0.081 ± 0.010 |
| UnTrac | 0.419 ± 0.063 | 0.743 ± 0.086 | 0.314 ± 0.223 | 0.403 ± 0.033 | 0.518 ± 0.122 | 0.246 ± 0.082 |
| UnTrac-Inv | 0.372 ± 0.008 | 0.813 ± 0.012 | 0.582 ± 0.016 | 0.393 ± 0.037 | 0.275 ± 0.125 | 0.360 ± 0.017 |

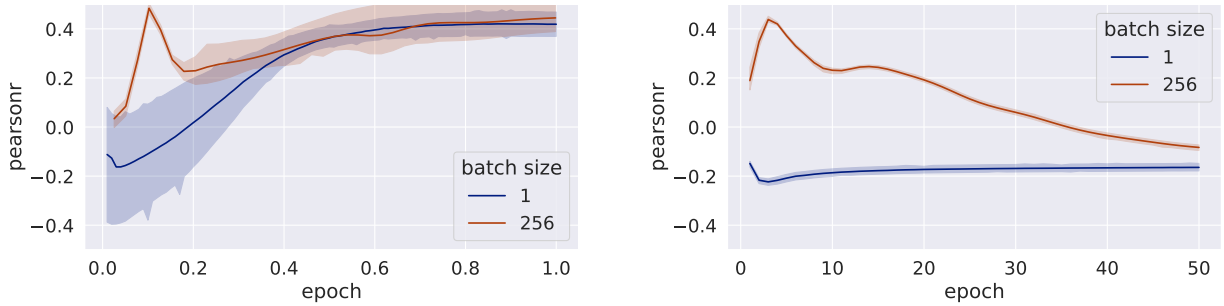


**図 3** UnTrac（左）と UnTrac-Inv（右）で推定した影響と実際の影響とのピアソン相関係数。横軸は逆学習のエポック数。4 回試行した平均と 95%信頼区間を表示。各学習データセットサイズは均一で、評価データセットには ToxiGen を使用。

## 4.2 事前学習における影響推定

LLM は差別やバイアス、不正確な内容を含む生成を行うことがある。本節では、そうした有害なコンテンツの生成に対する事前学習データセットの影響を、提案法が正確に推定できるか検証する。

**モデル** 本実験では open pre-trained transformer (OPT) [13]（125 万パラメータ）を用いた。下記の事前学習データセットで 1 エポック学習した。

**データセット** OPT の事前学習で用いられた 8 つのデータセットを学習に使用した。提案法がデータセットの混合比率によらず有効であることを示すため、各データセットサイズが均一な場合と、不均一な場合のそれぞれで実験した。学習データセットの詳細は A.2 節を参照されたい。評価データセットには、ToxiGen [14]、WinoBias [15] 及び TruthfulQA [16] を採用した。ToxiGen はマイノリティに対する差別的な文章を、WinoBias はある職業とそれに対する代名詞（he/she）を含む文章を、TruthfulQA は様々な分野にわたる質問とそれに対する不正確な回答を格納したデータセットである。差別/バイアスを含む文章や不正確な回答の対数尤度を損失関数として測ることで、各学習データセットの影響を推定する。

**結果** 表 2 に、leave-one-out により算出された実際の影響と、各手法で推定された影響とのピアソン相関係数を示す. 全データセットを通じて、UnTrac と UnTrac-Inv は実際の影響を比較的高い精度で推定できている。GradDot、GradCos 及び HIF (Arnoldi) は Winobias では高い精度を示すものの、Toxigen や TruthfulQA では特に学習データセットサイズが不均一の場合に精度が低い。TracIn は特にデータセットサイズが不均一の場合に有効でないことがわかる。

図 3 に、逆学習のステップ数と、影響値の推定精度（ピアソン相関係数）との関係を示す。UnTrac はバッチサイズに関わらず、逆学習が進むと安定した精度を示す一方、3.2 節の議論の通り、UnTrac-Inv はバッチサイズが小さい場合やステップ数が大きい場合有効でない。また 3.3 節の議論によれば、GradDot、GradCos 及び HIF は、逆学習のステップ数が 1 の場合に UnTrac と UnTrac-Inv の 1 次近似とみなせる。UnTrac と UnTrac-Inv はステップ数が 1 のとき共に精度が低く、逆学習の観点からみると、既存手法は 1 ステップでの推定に限界があったといえる。

# 5 おわりに

本稿では学習済モデルから学習データや評価データを逆学習することで、評価データに対する学習データの影響を推定できることを示した。本研究が、Chain-of-thought など、LLM の特異な能力を引き出す学習データの解明に役立つことを期待する。

## 謝辞

本研究は、NEDO JPNP20006 及び JST CREST JPMJCR21D1 の支援を受けたものである。

## 参考文献

[1] Frank R Hampel. The influence curve and its role in robust estimation. **Journal of the american statistical association**, Vol. 69, No. 346, pp. 383–393, 1974.

[2] Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. In **International conference on machine learning**, pp. 1885–1894. PMLR, 2017.

[3] Pang Wei W Koh, Kai-Siang Ang, Hubert Teo, and Percy S Liang. On the accuracy of influence functions for measuring group effects. **Advances in neural information processing systems**, Vol. 32, , 2019.

[4] Garima Pruthi, Frederick Liu, Satyen Kale, and Mukund Sundararajan. Estimating training data influence by tracing gradient descent. **Advances in Neural Information Processing Systems**, Vol. 33, pp. 19920–19930, 2020.

[5] Samyadeep Basu, Phil Pope, and Soheil Feizi. Influence functions in deep learning are fragile. In **International Conference on Learning Representations**, 2021.

[6] Anders Søgaard. Revisiting methods for finding influential examples. **arXiv preprint arXiv:2111.04683**, 2021.

[7] Andrea Schioppa, Katja Filippova, Ivan Titov, and Polina Zablotskaia. Theoretical and practical perspectives on what influence functions do. **arXiv preprint arXiv:2305.16971**, 2023.

[8] Joel Jang, Dongkeun Yoon, Sohee Yang, Sungmin Cha, Moontae Lee, Lajanugen Logeswaran, and Minjoon Seo. Knowledge unlearning for mitigating privacy risks in language models. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, **Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)**, pp. 14389–14408, Toronto, Canada, July 2023. Association for Computational Linguistics.

[9] Jiaao Chen and Diyi Yang. Unlearn what you want to forget: Efficient unlearning for LLMs. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, **Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing**, pp. 12041–12052, Singapore, December 2023. Association for Computational Linguistics.

[10] Andrea Schioppa, Polina Zablotskaia, David Vilar, and Artem Sokolov. Scaling up influence functions. In **Proceedings of the AAAI Conference on Artificial Intelligence**, Vol. 36, pp. 8179–8186, 2022.

[11] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. **arXiv:1412.6980v9**, 2014.

[12] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, Peter J Liu, et al. Exploring the limits of transfer learning with a unified text-to-text transformer. **Journal of machine learning research**, Vol. 21, No. 140, pp. 1–67, 2020.

[13] Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. Opt: Open pre-trained transformer language models. **arXiv preprint arXiv:2205.01068**, 2022.

[14] Thomas Hartvigsen, Saadia Gabriel, Hamid Palangi, Maarten Sap, Dipankar Ray, and Ece Kamar. ToxiGen: A large-scale machine-generated dataset for adversarial and implicit hate speech detection. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio, editors, **Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)**, pp. 3309–3326, Dublin, Ireland, May 2022. Association for Computational Linguistics.

[15] Jieyu Zhao, Tianlu Wang, Mark Yatskar, Vicente Ordonez, and Kai-Wei Chang. Gender bias in coreference resolution: Evaluation and debiasing methods. In Marilyn Walker, Heng Ji, and Amanda Stent, editors, **Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)**, pp. 15–20, New Orleans, Louisiana, June 2018. Association for Computational Linguistics.

[16] Stephanie Lin, Jacob Hilton, and Owain Evans. TruthfulQA: Measuring how models mimic human falsehoods. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio, editors, **Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)**, pp. 3214–3252, Dublin, Ireland, May 2022. Association for Computational Linguistics.

[17] Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In **Proceedings of the IEEE international conference on computer vision**, pp. 19–27, 2015.

[18] Trieu H Trinh and Quoc V Le. A simple method for commonsense reasoning. **arXiv preprint arXiv:1806.02847**, 2018.

[19] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. **arXiv preprint arXiv:1907.11692**, 2019.

[20] Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, et al. The pile: An 800gb dataset of diverse text for language modeling. **arXiv preprint arXiv:2101.00027**, 2020.

**表 3**　各手法により推定された各学習データセットの影響値。手法ごとに値域が異なるため、影響値を標準化して表示。

| データセット | 1: 類似タスク/同出力形式 | 2: 類似タスク/異出力形式 | 3: 非類似タスク/同出力形式 | 4: 非類似タスク/異出力形式 |
|---|---|---|---|---|
| GradDot | 0.864 | -0.107 | 0.836 | -1.594 |
| GradCos | 1.456 | -0.508 | 0.291 | -1.240 |
| HIF (LISSA) | -1.294 | 1.517 | -0.111 | -0.111 |
| TracIn | -0.331 | 1.690 | -0.443 | -0.916 |
| UnTrac | 1.330 | 0.600 | -0.913 | -1.018 |
| UnTrac-Inv | 1.688 | -0.212 | -0.651 | -0.826 |
| 実際の影響 | 1.416 | 0.462 | -1.031 | -0.847 |

**表 4**　各ハイパーパラメータにおける提案法の精度（提案法により推定された影響と実際の影響とのピアソン相関係数）。4 回の試行における平均と標準偏差を表示。各学習データセットサイズは均一で、評価データセットに ToxiGen を使用。

| 勾配法 | SGD | SGD w/ momentum | RMSProp | Adam | Adafactor |
|---|---|---|---|---|---|
| UnTrac | -0.147 ± 0.014 | -0.239 ± 0.011 | 0.418 ± 0.063 | 0.419 ± 0.063 | 0.345 ± 0.179 |
| UnTrac-Inv | -0.100 ± 0.069 | -0.099 ± 0.070 | -0.231 ± 0.012 | 0.376 ± 0.008 | 0.313 ± 0.003 |
| 学習率 | 5e-06 | 1e-05 | 5e-05 | 1e-04 | 5e-04 |
| UnTrac | -0.127 ± 0.302 | 0.312 ± 0.311 | 0.419 ± 0.063 | 0.377 ± 0.040 | 0.329 ± 0.015 |
| UnTrac-Inv | 0.100 ± 0.084 | 0.197 ± 0.067 | 0.376 ± 0.008 | 0.137 ± 0.019 | 0.027 ± 0.015 |

# A　付録

## A.1　ファインチューニングの影響推定

**実験結果の詳細**　表 3 に、各手法で推定された各学習データセットの影響値と、leave-one-out により推定された実際の影響を示す。leave-one-out によれば、類似タスクである学習データセット 1,2 が評価データセットに与える実際の影響は、非類似タスクである学習データセット 3,4 より大きい。提案手法である UnTrac 及び UnTrac-Inv も同様の傾向を示しており、タスクの類似性に基づいて影響を推定できており、出力形式には大きく影響されないことが示唆される。一方、既存手法はいずれも学習データセット 1,2 の影響を過小評価、または学習データセット 3 の影響を過大評価しており、実際の影響を正しく推定できていないことがわかる。

## A.2　事前学習における影響推定

**実験設定の詳細**　本実験では OPT の学習に用いられた BookCorpus [17], CC-Stories [18], CCNewsV2 [19] 及び Pile [20] に含まれる 5 つのデータセット: PJ Gutenberg, HackerNews, OpenWebText2, Pile-CC, Wikipedia を事前学習に使用した。各データセットサイズが均一の場合、各データセットに含まれるサンプル数は 40,000 件である。一方、データセットサイズが均一でない場合の各サンプル数は、Pile-CC: 96,000, OpenWebText2: 64,000, CCNewsV2: 48,000, BookCorpus: 32,000, Stories: 32,000, PJ Gutenberg: 16,000, HackerNews: 16,000, Wikipedia: 16,000 である。事前学習データセットの全サンプルを用いて影響を推定すると計算コストが膨大になるため、本実験では無作為に抽出した 10,000 件を各手法の影響推定に用いた。実験結果が抽出されたサンプルに依らないことを示すため、異なるサンプルについて 4 回試行を行い、その平均と標準偏差を報告している。

## A.3　ハイパーパラメータに対する頑健性

表 4 に、他のハイパーパラメータを固定しつつ、勾配法と学習率を変化させた時の提案法の精度を示す。UnTrac は、RMSProp、Adam 及び Adafactor を使用した場合に良い性能を示しており、勾配を考慮した preconditoner が重要な役割を果たしている。UnTrac-Inv についても同様の傾向がみられるが、RMSProp を使用した場合には精度が低くなる。学習率を変化させた場合、UnTrac は低い学習率では収束せず推定精度が低いものの、高い学習率を使用した場合には安定して良い性能を示す。一方、UnTrac-Inv は比較的性能が学習率に影響されやすいが、いずれの学習率においても推定された影響は実際の影響と正の相関を示す。