# 継続編集における知識保持と知識グラフ構造の関係について

石垣龍馬 [1]　関口正登 [1]　前田英作 [1]
[1] 東京電機大学
{24amj02@ms, 25amj16@ms, maeda.e@mail}.dendai.ac.jp

## 概要

LLM に対する知識編集は特定事実の更新を可能にする一方で，編集の累積に伴い，既編集知識および未編集知識の保持が困難になることが知られている．その要因として編集対象の次数や編集対象間距離などの構造が関与すると考えられるが，自然言語コーパス由来の事前学習知識を編集対象とした既存研究では，知識構造の不透明さや文脈依存性が混在し，要因の切り分けが難しい．本研究では，人工知識グラフを学習した小型 LLM を「(subject, relation)→object」を返す推論非依存の知識ベースとして構築し，継続知識編集における知識保持をグラフ構造に基づいて分析した．具体的には，Barabási–Albert グラフ由来のトリプルで学習したモデルに対し，編集 subject の次数および編集 subject 間の hop 距離を制御した編集対象のサンプリングを行い，各ステップで既編集集合および未編集集合の正解率を算出した．その結果，次数や距離といった KG 上の構造変数では説明しきれない挙動が一部条件で観測され，構造に基づく予測と評価の対応が必ずしも一貫しなかった．これは，LLM が学習元 KG をそのまま内部表現しているとは限らず，条件により異なる表現を形成する可能性を示唆する．本研究は，継続編集の不安定性を構造要因に還元して体系的に議論するための実験基盤を提供する．

## 1　はじめに

大規模言語モデル（LLM）は膨大な知識を内部に保持する一方で，hallucination による誤情報の生成や，時間経過に伴う事実の陳腐化は避けられない．実運用では，これらに起因して出力の正確性や最新性が損なわれ得るため，誤った事実の訂正や最新情報への更新を，事前学習の再実行なしに実現するニーズが高まっている．このニーズに対し，局所的なパラメータ更新によって特定知識のみを変更する知識編集（knowledge editing）が提案され，ROME[1] や MEMIT[2] など多様な手法が発展してきた．しかし近年，同一モデルに複数回の編集を継続的に適用する設定では，編集の累積に伴い編集の信頼性が急速に崩れることが大きな課題として顕在化している．具体的には，編集回数の増加に伴って，編集済み知識の保持が難しくなるだけでなく，未編集知識の保持や下流タスク性能も低下することが報告されている [3].

一方で，これらの問題の要因を突き詰めようとすると，現行の評価設定には難しさがある．多くの研究は，自然言語コーパス由来の事前学習知識を対象に編集を行うため，知識の結びつきや文脈依存が混ざり合い，知識の保持を構造要因として切り分けて理解することが難しい．そこで，人工知識グラフ（KG）を設計してモデルに学習させ，モデルが獲得する知識構造そのものを制御した実験が提案されている [4]．特に，Barabási–Albert（BA）グラフのように次数分布が偏る構造では，entity の次数と副作用の大きさが強く結びつくことが報告されている．

本研究はこの発想を継承し，管理された KG を学習した小型 LLM を，推論能力を前提としない「(subject, relation)→object」を返す知識ベース（KB）として構築し，継続知識編集における知識保持を，KG 構造に基づいて分解することを目的とする．この設計は，評価対象をトリプルの予測に限定することで，推論・プロンプト依存などの誤差要因を抑えつつ，編集の影響を精密に追跡できる利点がある．その上で，継続知識編集における編集対象の選び方を「各編集 subject の次数」と「編集 subject 同士の hop 距離」で制御し，各編集ステップにおける知識の保持を分析する（図 1）．評価は編集ステップを時間軸とみなし，これまでに既知識集合に対する正解率と，未編集知識集合に対する正解率を追跡する．本研究は継続知識編集で観測される不安定性を，自然言語由来の複雑要因から切り離し，次数・距離という構造で説明可能な形に還元する実験基盤を提供する．

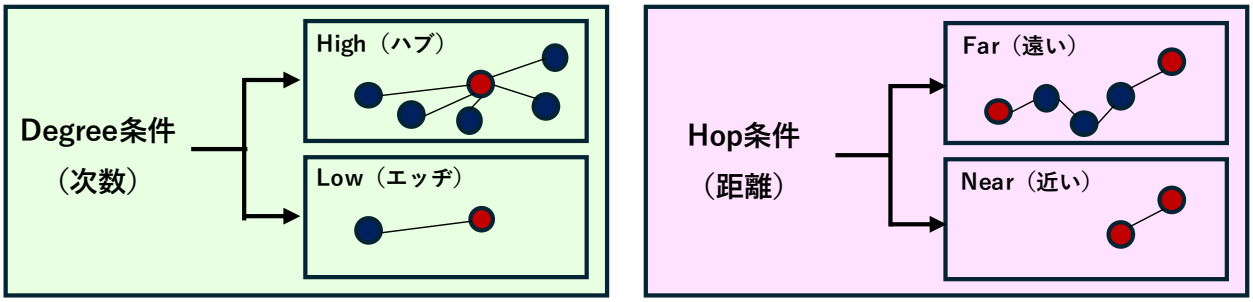


**図 1**　編集対象サンプリング条件の概念図．subject の次数に基づく degree 条件（high: ハブ，low: エッヂ）と，編集 subject 間の hop 距離に基づく hop 条件（far: 遠い，near: 近い）の各軸で編集候補を構成する．

## 2　提案

継続知識編集に伴う忘却（既編集知識の保持低下と，未編集知識への副作用）を，知識グラフ（KG）構造の観点から分解・可視化する評価枠組みを提案する．編集の対象は，(subject, relation) から object を出力する知識ベース（KB）的な振る舞いに限定した小型モデルとし，推論やプロンプト設計の影響を抑えた条件で，編集の累積による忘却の傾向を観測する．本提案の狙いは，KG 上の構造を LLM 内部表現に仮定することではなく，むしろその対応関係を観測から帰納的に検証できるようにする点にある．すなわち，次数・距離を制御した学習・編集実験を通じて，LLM 表現が KG とどの程度似た性質を示すのかを分析できる．

## 3　実験設定

Barabási–Albert（BA）グラフから生成した人工 KG（entity 数 1200，relation 種類 250，triple 数 30000）を用いる．各トリプル $(s, r, o)$ は $E_i\ R_j\ E_k$ の 3 トークン列として表現し，モデルには入力 $(s, r)$ に対して対応する $o$ を出力する知識ベース（KB）的ふるまいを学習させる．モデルは decoder-only の GPT（12 層，8 ヘッド，隠れ次元 512，MLP 次元 2048）とする．

継続知識編集では，編集ステップを $k = 1, \ldots, N$ とし，各ステップで 1 つの知識を編集する．編集手法は ROME に固定し，各ステップ終了時点のモデルで評価を行う．編集候補は，subject 次数に基づく degree 条件（high/low）と，編集対象同士の subject 起点 hop 距離に基づく hop 条件（far/near）の 2 つの軸でそれぞれ構成する．

評価では，編集対象集合 $E = \{(s_i, r_i)\}_{i=1}^{N}$ と未編集集合 $U = \{(s_j, r_j)\}_{j=1}^{M}$ を $E \cap U = \varnothing$ となるように構成し，$E$ は N=100 個，$U$ は M=1000 個を選択する．各ステップ $k$ で，固定した未編集集合に対する正解率（Unedited-ACC）と，時点 $k$ までに編集した知識に対する正解率（Edited-ACC）を算出してステップ方向に追跡し，編集の累積に伴う性能低下を分析する．乱数シードを変えた同一条件を 10 回試行し，各ステップ $k$ における試行平均と，95%信頼区間を可視化する．

## 4　結果

### 4.1　degree 条件

degree 条件で編集候補を high と low のそれぞれをサンプリングした際の継続編集の結果を図 2 に示す．各条件について 10 回試行を行い，薄線は各試行の推移，太線は試行平均，淡色の帯は平均の 95% 信頼区間を表す．左図を見ると，high 条件の方が，Unedited-ACC がステップの進行に伴って大きく低下していることがわかる．一方で，右図を見ると，high 条件の方が Edited-ACC はやや低いものの，条件間の差は Unedited-ACC の差と比べて小さいことがわかる．また，分散については，いずれの条件においても，Edited-ACC の方が，Unedited-ACC より大きいことがわかる．

### 4.2　hop 条件

hop 条件で編集候補を far と near のそれぞれをサンプリングした際の継続編集の結果を図 3 に示す．こちらも各条件について 10 回試行を行い，薄線は各試行の推移，太線は試行平均，淡色の帯は平均の 95%信頼区間を表す．右図を見ると，near 条件の方が，far 条件と比較して Edited-ACC のステップごとの低下が極端に大きいことがわかる．一方で，左図を見ると，near 条件の方が Edited-ACC はやや低いものの，条件間の差は Edited-ACC の差と比べて小さ

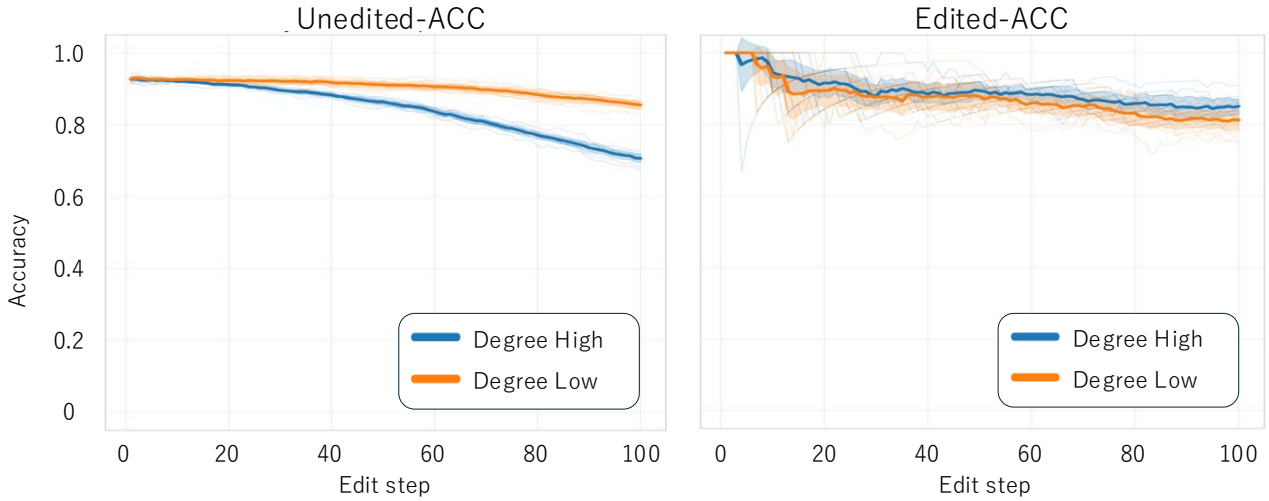


**図 2** degree 条件における継続知識編集の性能推移．横軸は edit step，左は固定した未編集集合に対する正解率（Unedited-ACC），右は時点 $k$ までに編集した知識に対する正解率（Edited-ACC）を示す．high/low は編集 subject の次数に基づくサンプリング条件である．

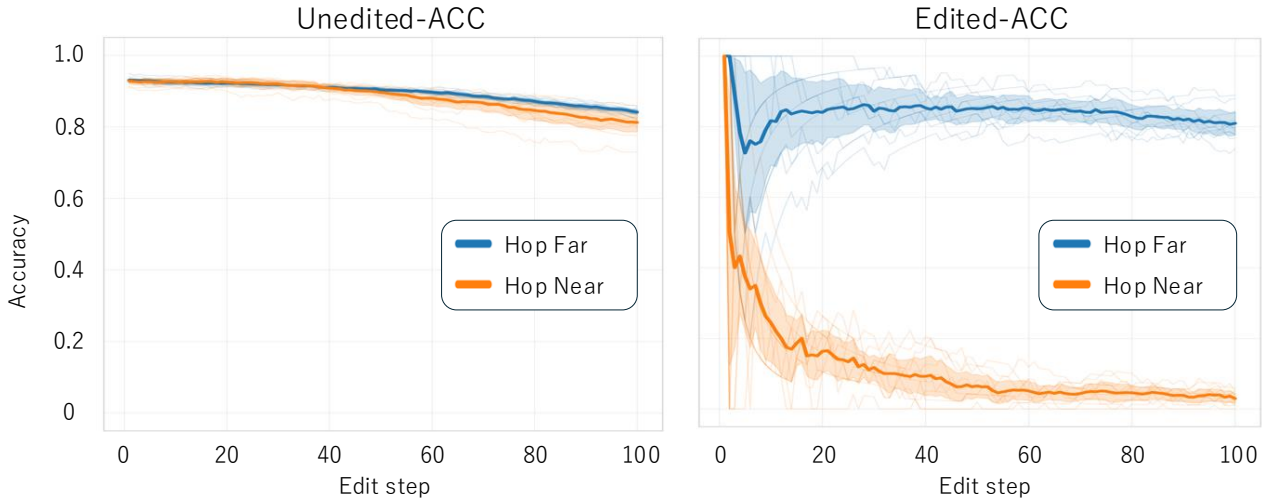


**図 3** hop 条件における継続知識編集の性能推移．横軸は edit step，左は固定した未編集集合に対する正解率（Unedited-ACC），右は時点 $k$ までに編集した知識に対する正解率（Edited-ACC）を示す．far/near は編集 subject 同士の hop 距離に基づくサンプリング条件である．

いことがわかる．また，分散については，いずれの条件においても，Edited-ACC の方が，Unedited-ACC より大きいことがわかる．

# 5 考察

## 5.1 degree 条件

degree 条件の結果（図 2）で得られた傾向は，high 条件で編集対象となる subject が KG 内でハブに近い存在であり，その subject に紐づく事実（$(s, r) \to o$）が多いことと整合的である．すなわち，ハブ subject に対する局所修正は，モデル内部では subject 表現やその周辺の結合に対して相対的に広い影響を持ちやすく，ターゲット編集自体を成立させることはできても，未編集知識に対して副作用が蓄積しやすい，という解釈ができる．

一方，Edited-ACC における high/low 差が Unedited-ACC ほど大きくない点は，ハブは編集が難しいという単純な仮説だけでは説明しにくい．むしろ，ROME の更新がターゲットに対しては強制的に整合させられる一方で，副作用は編集の回数に比例して蓄積しやすいことが考えられる．Edited-ACC の分散が大きい点に関しては，同じ degree 帯からサンプリングしても，対象 subject が持つ relation の種類や局所近傍の構成により，編集が干渉しやすい経路が試行ごとに変わりうるためだと考えられる．

### 5.2 hop 条件

図 3 の Edited-ACC において，near 条件で Edited-ACC がステップの進行に伴って急激に低下していることは，編集が局所的な近傍に集中し，強く関連した subject 周辺に対して繰り返し介入が起きていると考えられ，この場合，モデル内部では類似した表現に対して編集が重畳しやすく，結果として知識の保持が崩れやすいと考えられる．一方で far 条件では，編集対象が既編集集合から相対的に離れた領域へ分散するため，編集間干渉が弱まり，Edited-ACC の劣化が緩やかになったと解釈できる．

興味深いのは，near/far の差が主に Edited-ACC で顕在化し，Unedited-ACC では差が比較的小さい点である．KG の素朴な直観では，分散した編集は広域に波及し，Unedited-ACC に差が現れやすいと予想される．それにもかかわらず差が小さいことは，LLM が学習元の KG と同じ表現をしているとは限らないと言う可能性を示唆する．もっとも，本実験では編集手法を ROME に固定しているため，観測された差が ROME 特有の局所更新様式に依存している可能性は残る．したがって，hop 条件で見られた傾向が編集手法一般の性質なのか，あるいは特定手法の挙動なのかの切り分けは今後の課題とする．

## 6 関連研究

知識編集は，単発の編集成功率だけでなく，同一モデルに複数回の編集を継続的に適用したときに，過去編集の保持と未編集知識の維持が両立しにくい点が繰り返し指摘されている．とくに，編集回数の増加に伴う性能低下が，緩やかな劣化として進行する gradual forgetting と，ある時点で急激に破綻する catastrophic forgetting の二相として現れることが報告されており [3]，逐次編集下で更新知識の保持が段階的に弱まる現象を knowledge attenuation として整理する研究もある [5]．この不安定性は編集対象の選び方や順序にも依存し，同一ドメイン内での反復編集が周辺知識や推論を不安定化させ，順序依存性が顕著になりうることが示されている [6]．また，ターゲット更新に成功しても未編集知識や一般能力に副作用が及ぶ可能性があり，編集強度や回数の増加により推論・安全性など幅広い能力が一貫して悪化しうること [7] や，編集成功率と局所性・汎化の間にトレードオフがあることが分析されている [8]．さらに，編集の影響が近傍知識へ波及する Ripple Effect は，知識グラフ上の近傍エンティティに対する挙動変化として系統的に測定され [9]，連鎖的に更新されるべき知識をチェーンとして扱い整合性の観点から Ripple を捉える枠組みも提案されている [10]．干渉は多言語・多領域ではより顕在化しうるため，多言語 sequential editing での negative interference とその緩和手法も報告されている [11]．一方で，逐次編集を生涯学習として捉え，編集資源を分散配置して干渉と忘却を軽減する設計（例：Mixture-of-LoRA）[12] や，ROME/MEMIT 系を知識メモリ管理として再解釈し継続知識編集の問題を整理する研究 [13] も進んでいる．加えて，事前学習 LM の知識は外部から制御しづらいことから，人工的に設計した知識グラフを学習させ，トポロジーを制御しつつ編集の副作用を解析する制御実験が有効であり，次数分布の偏りが副作用の出方に影響しうることも示されている [4]．以上を踏まえると，本研究は，継続編集に固有の不安定性や Ripple の蓄積を，編集対象同士の距離や次数といった KG 構造に基づく構造要因として制御し，忘却がどの条件で生じやすいかを分析する点で，既存の逐次編集・副作用研究を補完する位置づけにある．

## 7 おわりに

本研究では，人工知識グラフを学習した小型 LLM を「(subject, relation)→object」と返す推論非依存の知識ベースとして位置づけ，継続知識編集に伴う知識保持を KG 構造（subject の次数と編集 subject 間の hop 距離）に基づいて分解・観測する評価枠組みを提案した．その結果，次数や距離といった KG 上の構造変数では説明しきれない挙動が一部条件で観測され，構造に基づく予測と評価の対応が必ずしも一貫しなかった．これは，LLM が学習元 KG をそのまま内部表現しているとは限らず，条件により異なる表現を形成する可能性を示唆する．

今後は，ROME 以外の編集手法や更新強度の違いを含めて再現性を検証し，観測された差が手法固有か一般的性質かを切り分ける．また，BA 以外のグラフ生成モデルや規模条件へ拡張し，忘却の傾向をより体系的に整理することで，継続知識編集の不安定性を説明・制御するための設計指針へとつなげたい．

# 参考文献


[1] Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in GPT. In **Advances in Neural Information Processing Systems 35**, Vol. 35, pp. 17359–17372. Curran Associates, Inc., 2022.

[2] Kevin Meng, David Bau, Alex Andonian, Yonatan Belinkov, and James Zou. Mass-editing memory in a transformer. In **International Conference on Learning Representations (ICLR)**, 2023.

[3] Akshat Gupta, Anurag Rao, and Gopala Anumanchipalli. Model editing at scale leads to gradual and catastrophic forgetting. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, **Findings of the Association for Computational Linguistics: ACL 2024**, pp. 15202–15232, Bangkok, Thailand, August 2024. Association for Computational Linguistics.

[4] Ryosuke Takahashi, Go Kamoda, Benjamin Heinzerling, Keisuke Sakaguchi, and Kentaro Inui. The curse of popularity: Popular entities have catastrophic side effects when deleting knowledge from language models, 2024.

[5] Qi Li and Xiaowen Chu. Can we continually edit language models? on the knowledge attenuation in sequential model editing. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, **Findings of the Association for Computational Linguistics: ACL 2024**, pp. 5438–5455, Bangkok, Thailand, August 2024. Association for Computational Linguistics.

[6] Zenghao Duan, Wenbin Duan, Zhiyi Yin, Yinghan Shen, Shaoling Jing, Jie Zhang, Huawei Shen, and Xueqi Cheng. Related knowledge perturbation matters: Rethinking multiple pieces of knowledge editing in same-subject. In Luis Chiruzzo, Alan Ritter, and Lu Wang, editors, **Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers)**, pp. 363–373, Albuquerque, New Mexico, April 2025. Association for Computational Linguistics.

[7] Jia-Chen Gu, Hao-Xiang Xu, Jun-Yu Ma, Pan Lu, Zhen-Hua Ling, Kai-Wei Chang, and Nanyun Peng. Model editing harms general abilities of large language models: Regularization to the rescue. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen, editors, **Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing**, pp. 16801–16819, Miami, Florida, USA, November 2024. Association for Computational Linguistics.

[8] Zhoubo Li, Ningyu Zhang, Yunzhi Yao, Mengru Wang, Xi Chen, and Huajun Chen. Unveiling the pitfalls of knowledge editing for large language models. In B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun, editors, **International Conference on Representation Learning**, Vol. 2024, pp. 19962–19978, 2024.

[9] Jun-Yu Ma, Zhen-Hua Ling, Ningyu Zhang, and Jia-Chen Gu. Neighboring perturbations of knowledge editing on large language models. In **Proceedings of the 41st International Conference on Machine Learning**, ICML'24. JMLR.org, 2024.

[10] Zilu Dong, Xiangqing Shen, Zinong Yang, and Rui Xia. ChainEdit: Propagating ripple effects in LLM knowledge editing through logical rule-guided chains. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar, editors, **Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)**, pp. 13558–13571, Vienna, Austria, July 2025. Association for Computational Linguistics.

[11] Wei Sun, Tingyu Qu, Mingxiao Li, Jesse Davis, and Marie-Francine Moens. Mitigating negative interference in multilingual knowledge editing through null-space constraints. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar, editors, **Findings of the Association for Computational Linguistics: ACL 2025**, pp. 8796–8810, Vienna, Austria, July 2025. Association for Computational Linguistics.

[12] Jiaang Li, Quan Wang, Zhongnan Wang, Yongdong Zhang, and Zhendong Mao. Elder: Enhancing lifelong model editing with mixture-of-lora. **Proceedings of the AAAI Conference on Artificial Intelligence**, Vol. 39, No. 23, pp. 24440–24448, Apr. 2025.

[13] Peng Wang, Zexi Li, Ningyu Zhang, Ziwen Xu, Yunzhi Yao, Yong Jiang, Pengjun Xie, Fei Huang, and Huajun Chen. Wise: rethinking the knowledge memory for lifelong model editing of large language models. In **Proceedings of the 38th International Conference on Neural Information Processing Systems**, NIPS '24, Red Hook, NY, USA, 2024. Curran Associates Inc.

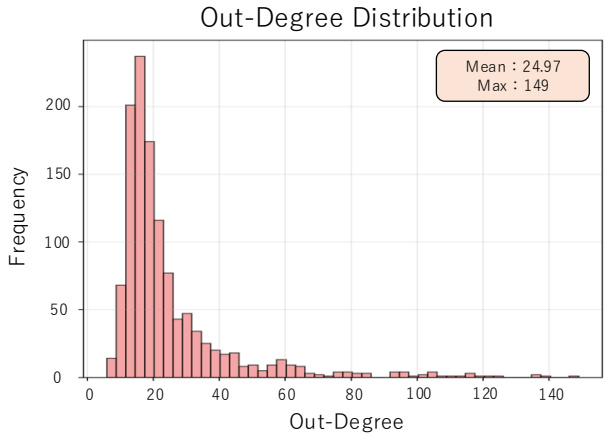


**図 4**　学習に使用した BA グラフの次数分布（Out-degree）.

# A　学習した LLM の設定

本研究で用いたモデルは，人工知識グラフ（BA）から生成したトリプルを用いて学習した decoder-only のアーキテクチャである GPT である．本モデルは，入力として $(s, r)$ を与えたときに対応する $o$ を出力する知識ベース（KB）的ふるまいを学習しており，継続知識編集実験における初期モデルとして用いる．モデルアーキテクチャは，GPT であり，層数 $n_{\text{layers}} = 12$，ヘッド数 $n_{\text{heads}} = 8$，隠れ次元 $d_{\text{model}} = 512$，MLP 次元 $d_{\text{mlp}} = 2048 (= 4 \times d_{\text{model}})$ とする．

学習データは Barabási–Albert（BA）知識グラフに基づく．各知識は 1 つのトリプル（Subject–Relation–Object）であり，“E_XXXX R_XXX E_YYYY” の形式で表現する（例: “E_0001 R_022 E_0000”）．エンティティは E_XXXX（4 桁），リレーションは R_XXX（3 桁）の統一命名を用いた．学習結果として，最高訓練精度は 93.02%となった．

# B　各条件のサンプリング方法

継続知識編集における編集対象知識の選び方として，subject 次数に基づく degree 条件と，編集対象 subject 間のホップ距離に基づく hop 条件の 2 種類を比較した．いずれも，各条件について 10 試行を行い，試行ごとのランダム性を平均化して傾向を評価する．また，各試行において未編集集合は 1,000 トリプルを固定して用い，編集ステップ数は 100 とする．

## B.1　Degree 条件

degree 条件では，知識グラフ（KG）における各 subject の出次数（out-degree）に基づいて，高次数（degree_high）と低次数（degree_low）を対比する．全 subject を degree でソートし，上位・下位の極端な部分のみを候補プールとして用いる．degree の分布は，図 4 のようになっており，平均的な degree が含まれないようそれぞれ 10 %の候補から 100 件を用いた．

## B.2　Hop 条件

編集対象 subject が KG 上で互いに分散する（hop_far）／密集する（hop_near）ように，subject 間の hop 距離を制御して編集対象を選んだ．subject 間距離は，subject を起点として有向辺 $(s, r, o)$ を辿る最短距離として定義した．まず初期トリプルを 1 つ選び，その subject を選択集合 $S$ に加える．次に，候補 subject $s$ に対し，既選択集合 $S$ への最小距離

$$d_{\min}(s; S) = \min_{s' \in S} \text{hop}(s, s') \tag{1}$$

を考える．それを繰り返し，$|S|$ が所定数（今回は 100）に達するまで反復する．hop_far では $d_{\min}(s; S)$ が大きい候補を優先し，既選択 subject 群から遠い subject を追加していくことで，編集対象を広く分散させる．hop_near では $d_{\min}(s; S)$ が小さい候補を優先し，既選択 subject 群の近傍に subject を集めることで，編集対象を局所に密集させる．