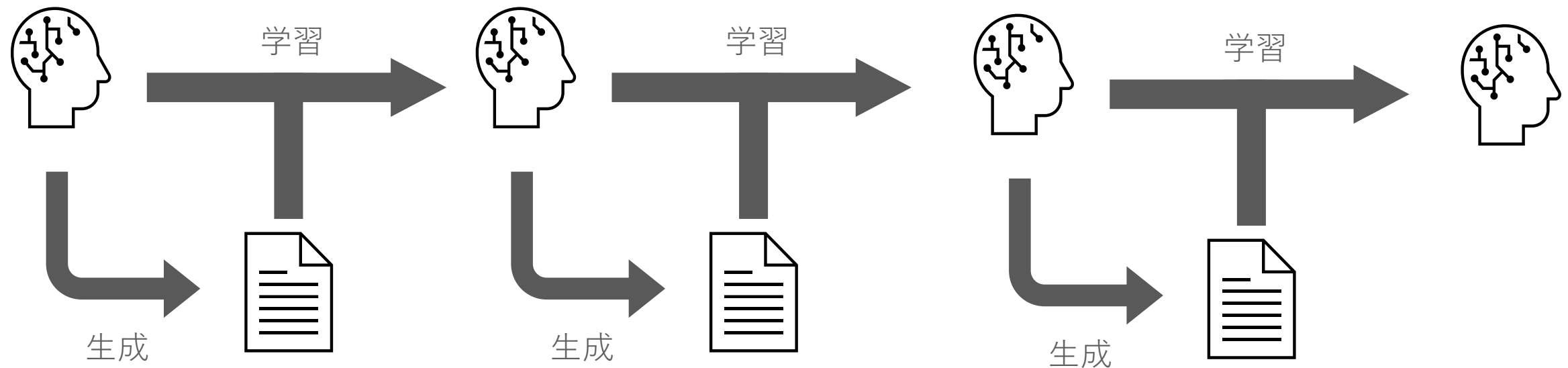


コード生成タスクにおける モデル崩壊現象の分析

S2310233 藤原颯太

背景：モデル崩壊とは

再帰的な反復学習によるモデルの退行プロセス^[1]

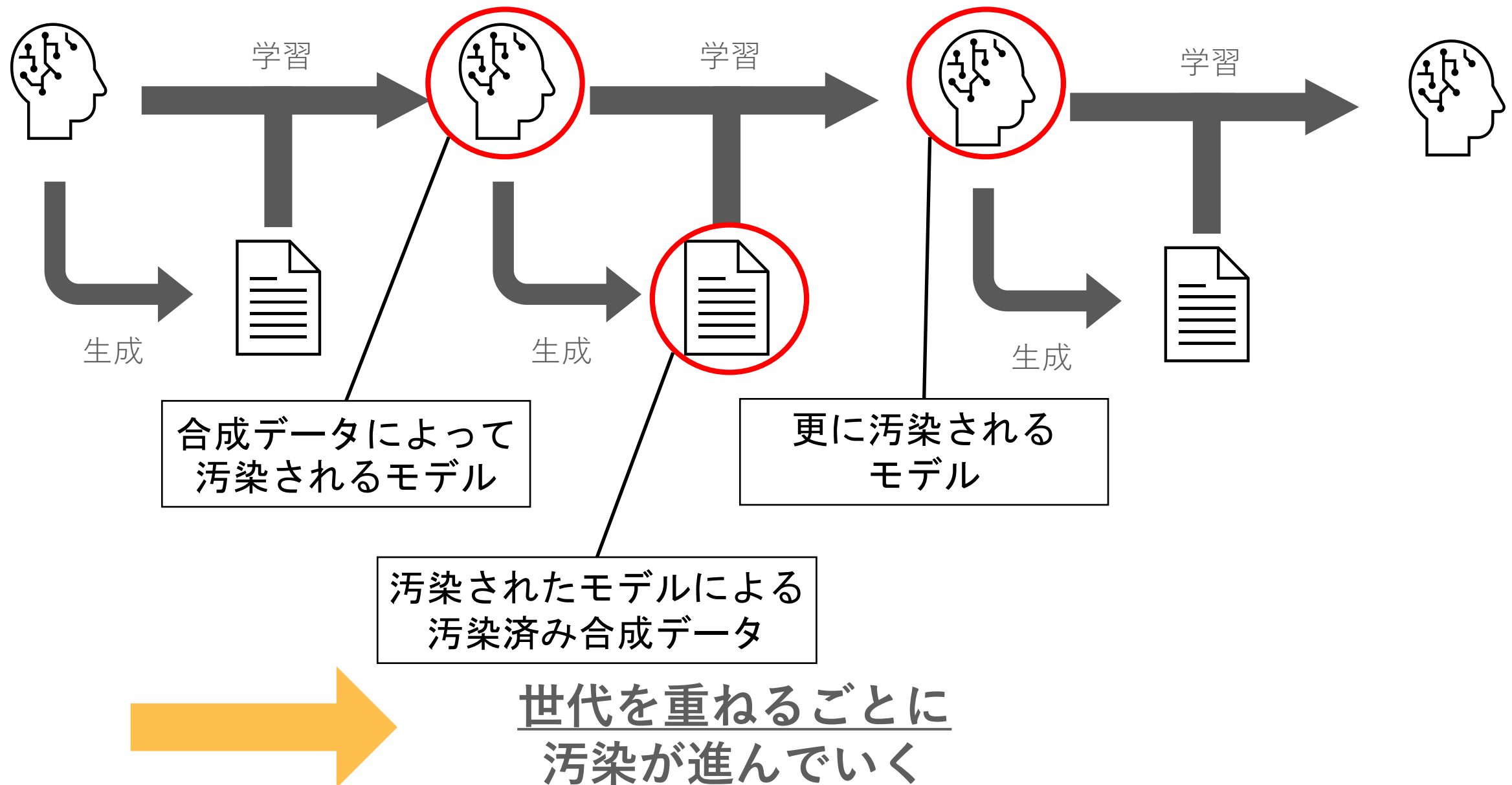


- LLMが生成した合成データを再学習する
- 世代を重ねるごとに元データの分布から乖離する
- 出現頻度の低い表現が消失していく

[1] I. Shumailov, Z. Shumaylov, Y. Zhao, N. Papernot, R. Anderson, and Y. Gal, "AI models collapse when trained on recursively generated data," Nature, vol. 631, pp. 755-759 (2024).

背景：モデル崩壊とは

再帰的な反復学習によるモデルの退行プロセス



背景：合成データ

生成AIの普及 [2]

- 大量の合成データがウェブ上に公開されるようになった
- 次世代モデルの学習時に合成データが混入

実データの枯渇 [3]

- 新規に収集可能な実データは先細りになっていく
- 合成データへの依存度は今後さらに高まると予想される



モデル崩壊の危険性は高まっている

[2] A. Al Mujahid, P. Chatterjee, and M. M. Imran, "Empirical study on the characteristics and evolution of AI-usage in GitHub repositories," arXiv:2606.06843 (2026).

[3] P. Villalobos, A. Ho, J. Sevilla, T. Besiroglu, L. Heim, and M. Hobbhahn, "Will we run out of data? Limits of LLM scaling based on human-generated data," in ICML '24 (2024).

先行研究^[1]

言語モデルの反復学習で崩壊を実証

- 各世代のモデルが生成した文章だけで次世代を学習
- 世代を重ねるごとに出力の多様性が失われる

微調整の繰り返しでも崩壊は起きる

- 事前学習済みモデルを出発点にした設定でも退行を確認
- 学習をやり直さなくても世代を重ねれば劣化が進む

課題

崩壊したモデルが生成した誤ったコードを採用してしまう危険性がある

- 開発中の手戻りが増加してしまう
- 詳細なレビューが必要になる

コード生成分野における再帰的な反復学習でのモデル崩壊は十分に検証されていない

- コード生成にLLMを利用するケースが増加している
- GitHub上のソースコードにも多くの合成コードが含まれている

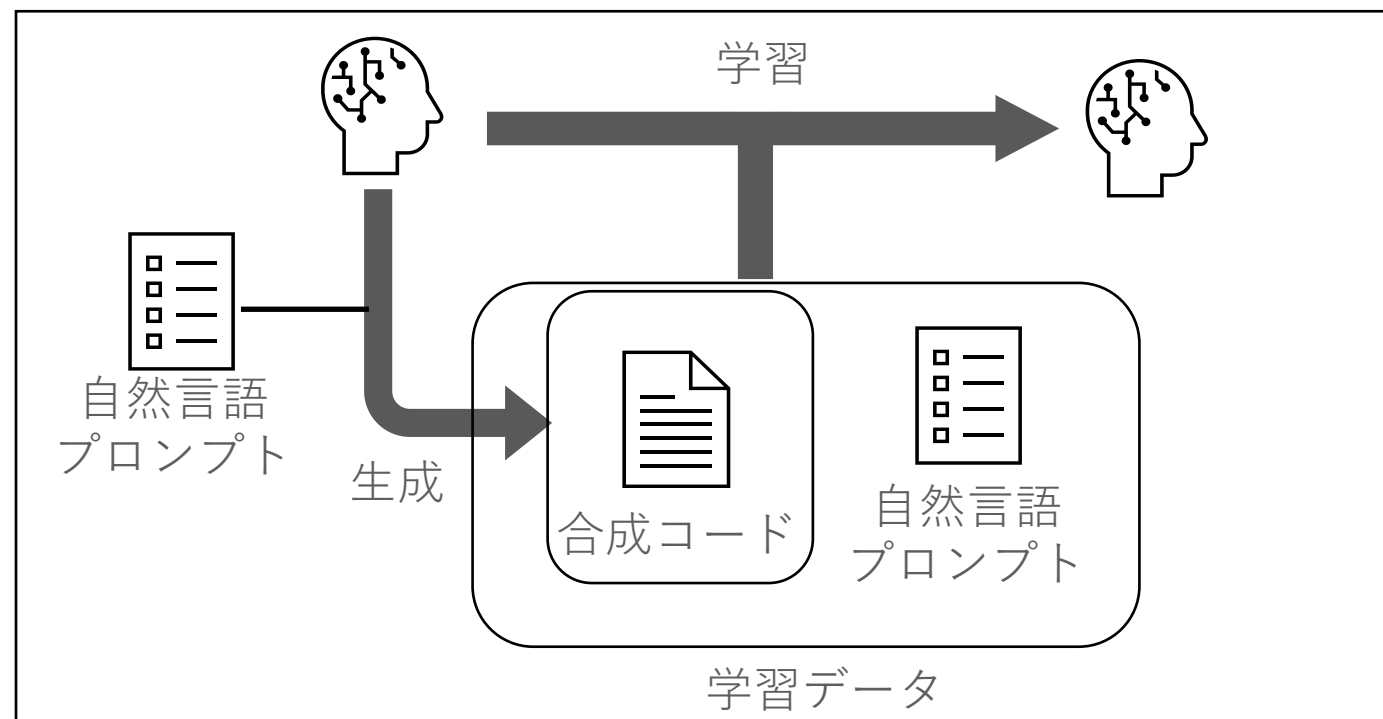
目的・アプローチ

本研究の目的

- コード生成タスクでのモデル崩壊現象の分析

アプローチ

- コード生成モデルを用いて再帰的な反復学習を行う
- 各世代のモデルの性能を評価する



1世代の学習

実験設定

モデル

- olmo3-32B [4]
 - 学習に使われたデータが公開されている
 - データリークが対策済み

データセット

- CodeSearchNet [5]
 - 2019年に収集されたデータ
 - 実データのみで構成されている
 - 学習データとの重複を除去して利用

[4] Team OLMO et al., "Olmo 3: Charting a path through the model flow to lead open-source language modeling," arXiv:2512.13961 (2025).

[5] H. Husain, H.-H. Wu, T. Gazit, M. Allamanis, and M. Brockschmidt, "CodeSearchNet challenge: Evaluating the state of semantic code search," arXiv:1909.09436 (2019).

実験設定

学習法

- QLoRA [6]
 - LoRAと重みパラメータの量子化を合わせた軽量な学習法 [7]
- SFT
 - 教師あり学習によるファインチューニング [8]
 - docstringを入力,コードを出力とする

合成データ生成

- temperature sampling [9]

[6] T. Dettmers, A. Pagnoni, A. Holtzman, and L. Zettlemoyer, "QLoRA: Efficient finetuning of quantized LLMs," in NeurIPS '23 (2023).

[7] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, "LoRA: Low-rank adaptation of large language models," in ICLR '22 (2022).

[8] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. L. Wainwright, et al., "Training language models to follow instructions with human feedback," in NeurIPS '22, pp. 27730-27744 (2022).

[9] A. Holtzman, J. Buys, L. Du, M. Forbes, and Y. Choi, "The curious case of neural text degeneration," in ICLR '20 (2020).

評価指標

ベンチマークによる評価

- モデルの性能を評価
 - HumanEval ^[10]
 - MBPP+ ^{[11][12]}

PPLによる評価

- トークンの予測しやすさを評価する指標
 - gen_nモデルで実データのPPLを測定
 - gen_0モデルでgen_nモデルによる合成データのPPLを測定
- モデル崩壊による分布の変化を観測

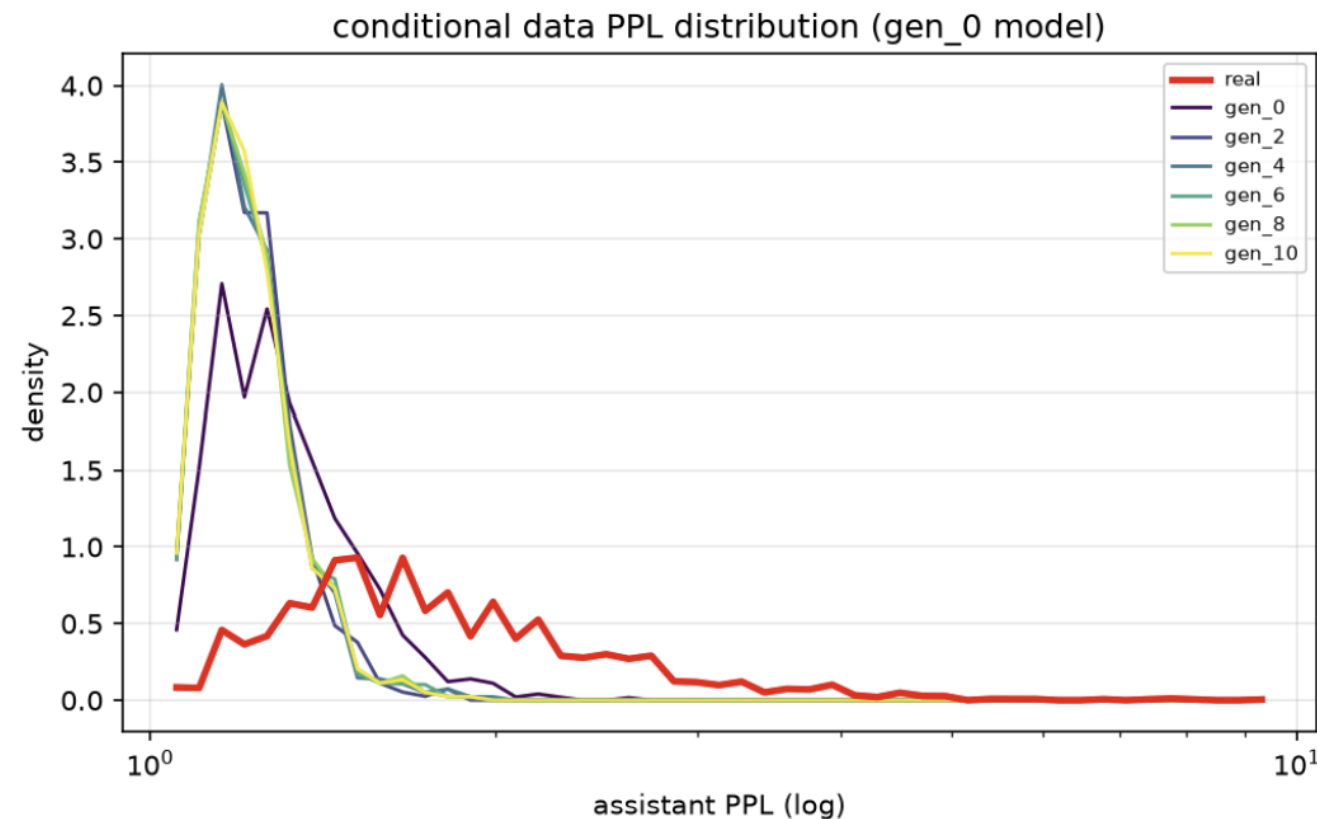
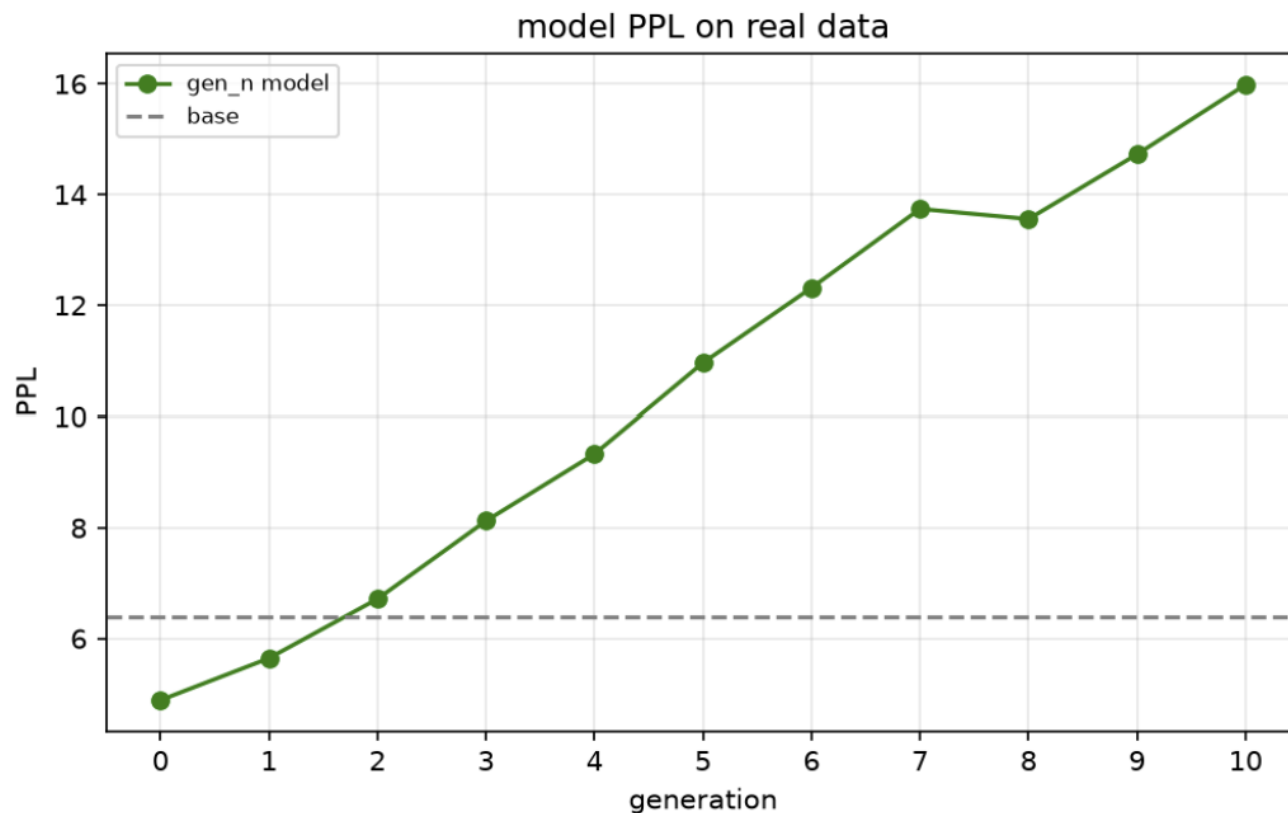
[10] M. Chen, J. Tworek, H. Jun, Q. Yuan, H. P. de O. Pinto, J. Kaplan, et al., "Evaluating large language models trained on code," arXiv:2107.03374 (2021).

[11] J. Austin, A. Odena, M. Nye, M. Bosma, H. Michalewski, D. Dohan, et al., "Program synthesis with large language models," arXiv:2108.07732 (2021).

[12] J. Liu, C. S. Xia, Y. Wang, and L. Zhang, "Is your code generated by ChatGPT really correct? Rigorous evaluation of large language models for code generation," in NeurIPS '23 (2023).

実験結果①

PPLによる評価(gen10まで)

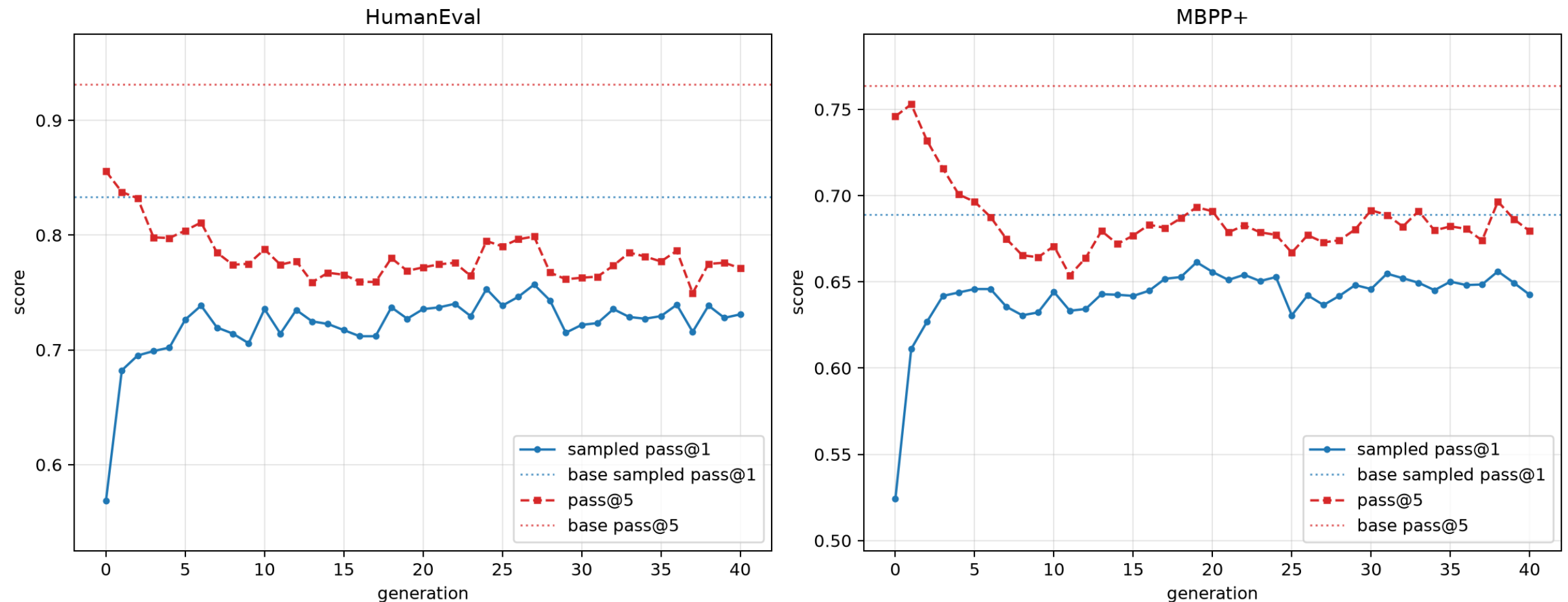


- 左図ではモデルの平均PPLの増加が確認できる
 - 実データを予測しにくくなっている->期待通りの結果
- 右図では分布の変化が確認できる
 - data_0からdata_10で左に山が表れている->頻出表現の強化で概ね期待通り

実験結果②

ベンチマークによる評価(gen40まで)

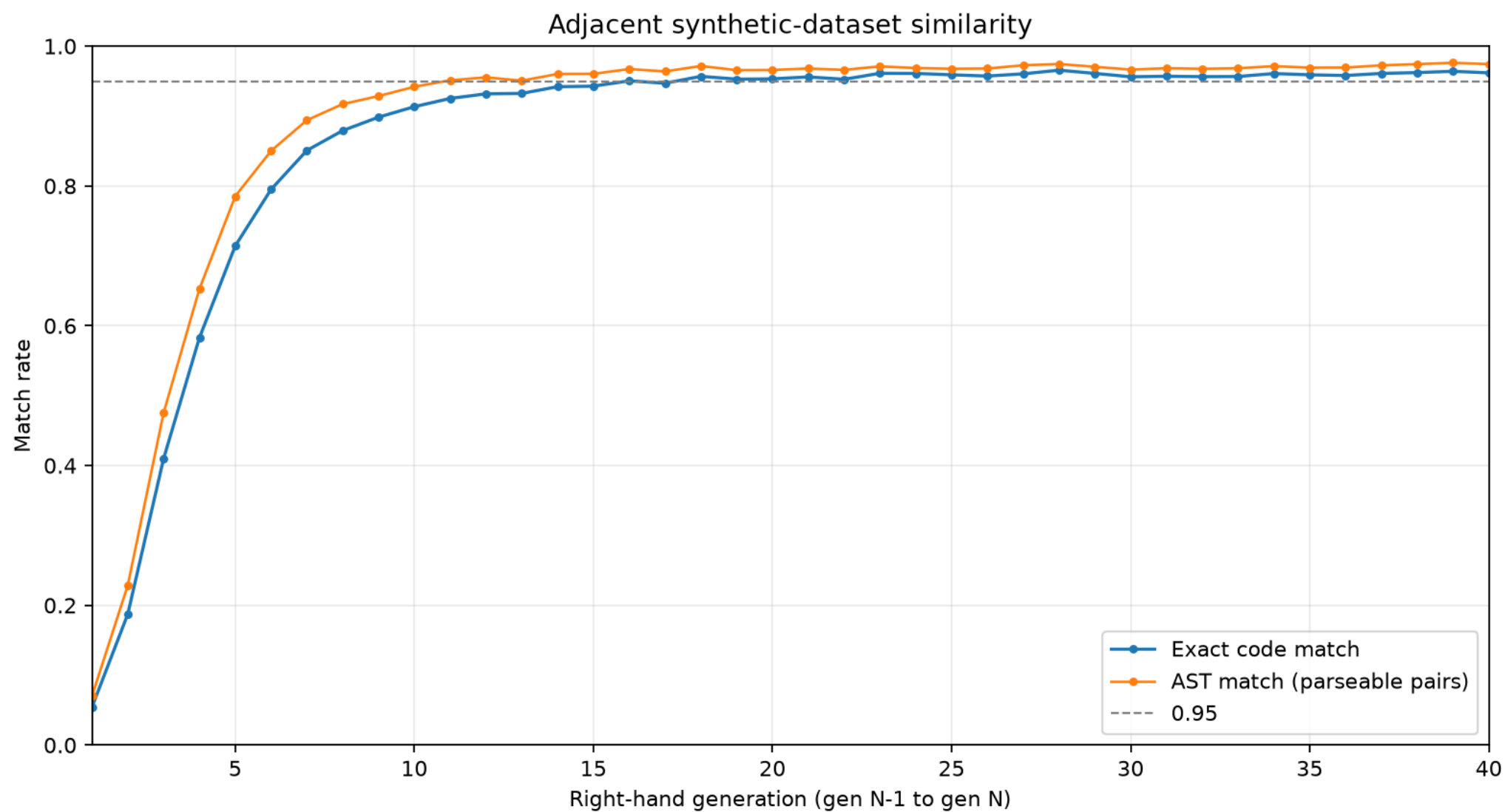
Benchmark performance across generations



- pass@5は初回以降も減少,greedyは初回以降少し増加
- Gen10あたりから変化が小さくなっていく

実験結果(補)

前世代との一致率



補足資料

参考文献

- [1] I. Shumailov, Z. Shumaylov, Y. Zhao, N. Papernot, R. Anderson, and Y. Gal, "AI models collapse when trained on recursively generated data," *Nature*, vol. 631, pp. 755-759 (2024).
- [2] A. Al Mujahid, P. Chatterjee, and M. M. Imran, "Empirical study on the characteristics and evolution of AI-usage in GitHub repositories," *arXiv:2606.06843* (2026).
- [3] P. Villalobos, A. Ho, J. Sevilla, T. Besiroglu, L. Heim, and M. Hobbhahn, "Will we run out of data? Limits of LLM scaling based on human-generated data," in *ICML '24* (2024).
- [4] Team OLMo et al., "Olmo 3: Charting a path through the model flow to lead open-source language modeling," *arXiv:2512.13961* (2025).
- [5] H. Husain, H.-H. Wu, T. Gazit, M. Allamanis, and M. Brockschmidt, "CodeSearchNet challenge: Evaluating the state of semantic code search," *arXiv:1909.09436* (2019).
- [6] T. Dettmers, A. Pagnoni, A. Holtzman, and L. Zettlemoyer, "QLoRA: Efficient finetuning of quantized LLMs," in *NeurIPS '23* (2023).

参考文献

- [7] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, "LoRA: Low-rank adaptation of large language models," in ICLR '22 (2022).
- [8] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. L. Wainwright, et al., "Training language models to follow instructions with human feedback," in NeurIPS '22, pp. 27730-27744 (2022).
- [9] A. Holtzman, J. Buys, L. Du, M. Forbes, and Y. Choi, "The curious case of neural text degeneration," in ICLR '20 (2020).
- [10] M. Chen, J. Tworek, H. Jun, Q. Yuan, H. P. de O. Pinto, J. Kaplan, et al., "Evaluating large language models trained on code," arXiv:2107.03374 (2021).
- [11] J. Austin, A. Odena, M. Nye, M. Bosma, H. Michalewski, D. Dohan, et al., "Program synthesis with large language models," arXiv:2108.07732 (2021).
- [12] J. Liu, C. S. Xia, Y. Wang, and L. Zhang, "Is your code generated by ChatGPT really correct? Rigorous evaluation of large language models for code generation," in NeurIPS '23 (2023).