

# 大規模言語モデルを用いた科学研究のための対話的作図システムの開発

一関高専 未来創造工学科情報・ソフトウェア系

村上 力

## 1. はじめに

学術論文や理工系科目の教材作成、サイエンスコミュニケーションの分野においては、時に文章よりもグラフやイラストを用いた説明の方が難解な概念の説明に適している場合がある。しかしながら TikZ や p5.js のようなグラフ作成プログラミング言語は習得が難解であり、また習得したとしてもコーディング作業の労力は膨大である。このような問題を解決するために、本研究では大規模言語モデルを用いて「グラフや概念図の作成を対話的に指示することで逐次修正が可能な作図システム」の開発に挑戦した。また、成果物としてこれを実現するためのデータセットを整備した。

## 2. 大規模言語モデル

俗に深層学習と呼ばれる膨大な学習可能パラメータが深い層状に重ねられたニューラルネットワークが様々な分野に適用され、それぞれの分野で高い性能を発揮している。この深層学習の一つとして、Transformer アーキテクチャ[1]を用いた言語モデルは自然言語処理の分野でこれまでのニューラル言語モデルの性能を凌駕する性能と、複数の異なるタスクへの高い応用性を示している。これを大規模言語モデル[2]（以降 LLM）と呼ぶ。膨大な学習可能パラメータと、これに見合った膨大な訓練データ（コーパス）を元に訓練されたモデルは、基盤モデル・事前学習済みモデルとも呼ばれており、多くの研究においてはこれを用いて、それぞれのタスクへとチューニングして利用している。

## 3. 大規模言語モデルのチューニング

LLM はプロンプトと呼ばれる指示文を受けて、これに対する出力結果を返す形が基本である。よってこのプロンプトを調整することで、作業者の意図した出力結果を生成させる技術が流行している。これをプロンプトエンジニアリングと呼び、またプロンプトを適応的に修正することはプロンプトチューニング[3]と呼ばれている。これらだけでも様々なタスクへ LLM を適応させることが可能であるが、一方で、インストラクションチューニングされていたとしても、事前学習済みモデルをプロンプトの工夫だけで期待通りの結果が得られるようにすることは困難な場合も多い。この場合に用いられる解決策がファインチューニングである。伝統的なファインチューニングはベースモデルの全てのパラメータを追加データによって微調整することが多いが、LLM の分野ではこのファインチューニングも膨大な学習可能パラメータのせいで実行が難しい。これを解決するために、事前学習済みモデルのパラメータを固定し、新たに比較的少数のパラメータを追加して効率的に LLM の挙動を調整する LoRA[4]–[6] と呼ばれる手法が登場した。LoRA は追加した学習可能パラメータ以外の訓練を

行わずに、LLM に新たな能力を獲得させることができる画期的な技術であった。事前学習に用いられた膨大なコーパスの中にはこの研究で着目した作図プログラムやその説明文が存在すると期待できるので、LoRA を用いることで、この知識を回答生成により集中的に活用させることができる。

#### 4. 本研究の目的と方針

本研究では、LLM を用いて学術論文・サイエンスコミュニケーションのための作図を支援するシステムを、ノートパソコンのような GPU 環境の乏しい計算機において動作する形で製作することを目的とした。そのために、利用できる LLM は約 30 億（以下 3B）パラメータまでとする。また、matplotlib, plotly などの Python のグラフ作成ライブラリの自動生成は、調査中に発表された軽量 LLM でも十分にコード生成が可能であることが確認できたため、これについては対象外とした。残る自由度の高いグラフ生成ライブラリ・ソフトウェアとして、有名なものは、TikZ と p5.js である。本研究では Web 上での公開に適した図を生成するために、p5.js のコードを生成するシステムを構築する。3B パラメータの LLM においては p5.js コードを過不足なく生成することは困難であるため、まずは p5.js コード生成能力を獲得させるためのデータセットを作成し、これを用いて 3B 以下の LLM を訓練する。

#### 5. Tikz データセットを元にした p5.js のデータセットの設計

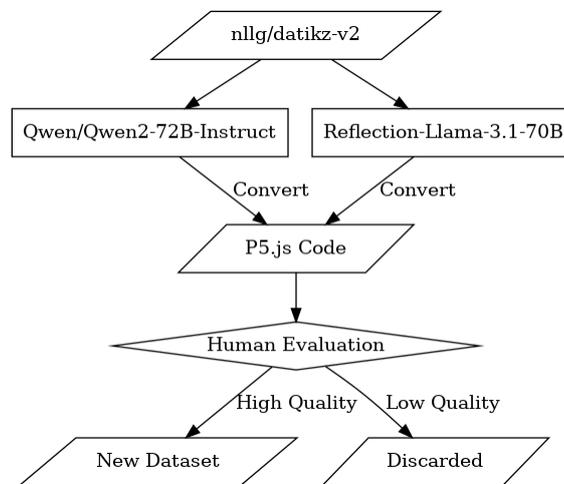


図 1. データセット作成の流れ

プログラミング能力が十分に実用に耐え得る LLM が公開されている現状で、本研究の目標に適した更なる調整を行うために、同様の目的の先行研究を TikZ において行った研究[7], [8] を元に、ここで用いられたデータセットを TikZ から p5.js へ翻訳するデータセットを作成する。nllg/datikz-v2[9]データセットには Web 上から集められた TikZ 製ベクターイメージと、これの caption, TikZ コードが収録されている。データセットには、再配布可能なデータとして 94532 個の訓練データ、442 個のテストデータが含まれている。本研究ではこのデータセットをベースにして、画像、キャプションはそのままに、TikZ コードを p5.js へ翻訳したコードを新しい項目として追加したデータセットを作成する。生成には LLM を用いた半自動化アプロ

ーチを採用した. Qwen/Qwen2-72B-Instruct[10], mattshumer/Reflection-Llama-3.1-70B[11]を用いて TikZ コードを p5.js へ翻訳する処理を行い, 適切に動作する方をデータセットへと収録する. 明らかに生成に失敗しているプログラムなどは人の手によって削除した. データセット作成の流れを図 1 に示す. その結果, p5.js コードの生成に失敗したデータや元々キャプションがないデータを除き, 作成に成功したデータは訓練データ 52390 個, テストデータ 442 個となった.

## 6. 訓練と生成結果

Qwen 2.5 coder 3B[12]に対して, このデータを用いたインストラクションチューニングを行った. フルファインチューニングは行わず, LoRA を用いた. 学習には RTX3060 12GB を用い, 48 時間かかった. 設定は表 1 に示す.

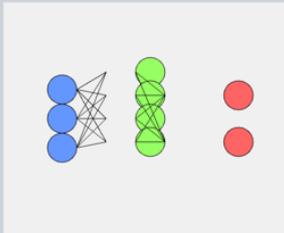
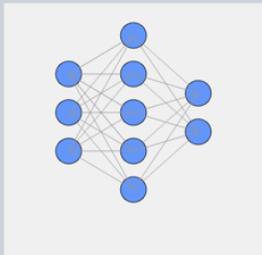
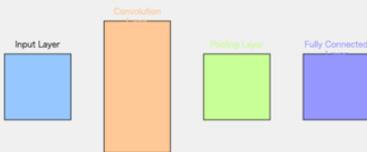
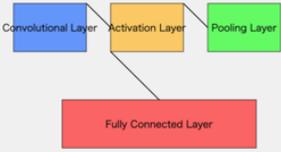
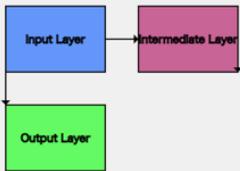
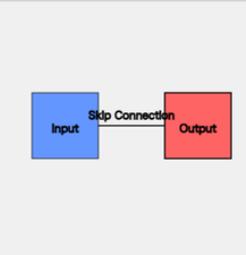
表 1. ファインチューニングの設定

base_model	Qwen/Qwen2.5-Coder-3B	ベースとなる事前学習済みモデル
max_seq_length	2048	最大シーケンス長
load_in_4bit	True	4bit 量子化の使用有無
lora_r	64	LoRA のランク (低ランク行列の次元数)
lora_alpha	128	LoRA のスケール係数
lora_dropout	0.05	LoRA のドロップアウト率
batch_size	2	バッチサイズ
gradient_accumulation_steps	4	勾配蓄積ステップ数
num_epochs	3	学習エポック数
learning_rate	2e-4	学習率
weight_decay	0.01	重み減衰率
warmup_steps	100	学習率ウォームアップのステップ数
optimizer	adamw	使用するオプティマイザ
lr_scheduler	cosine	学習率スケジューラの種類

ファインチューニング前のモデル (図中左列, 元のモデル) と, ファインチューニングされたモデル (図中中央列, 微調整後のモデル) を用いて, いくつかの指示文 (図中右列, プロンプト) をもとに図を生成した. それぞれ 10 回生成し, 最も良かった結果を表 2 に示す. プロンプト「三層のニューラルネットワーク」では, 全結合層が適切にノード間に図示されるようになったことが分かるが, それ以外のプロンプトでは大きな改善は見られなかった. ただしプロンプト「Skip-connection を図示せよ」では, 図中にそれと分かる言葉が登場する傾向が強くなったことが確認できた. CNN の図はデータセット中にも登場していたが, これ

を生成するための p5.js コードは比較的文字数が多かったため、今回の最大トークン数では十分にそれを再現してコード生成することができなかつたのだらうと推測できる。総じて、GPU 資源が乏しい環境において推論時間の短いモデルを用いた作図支援システムの構築においては、この最大トークン数がボトルネックになるのではないらうと考えられる。また、パラメータ数の少ない LLM では、複雑な概念を図示することが難しいのらうとも推測でき、これについては RAG[13], [14]のように外部のコーパスから情報収集を行わせるなどの対応が必要になるらう。

表 2. プロンプトと生成された svg 画像

プロンプト	元のモデル	微調整後のモデル
三層のニューラルネット		
シンプルな Convolutional Neural Networkを図示せよ		
Skip-connectionを図示せよ		

## 7. 作成したシステム

今回の研究のための作成した Web インターフェイス (図 2) を紹介する。図中左側「テキストプロンプト」のタブにおいて、作成したい図の説明を行う。「図を生成」ボタンをクリックするとサーバーサイドで LLM が動作し、p5.js のコードが生成される。この p5.js コードが生成終了すると、「図のプレビュー」タブで図を表示する。生成された p5.js コードも確認できるため、手作業で図を修正することも可能である。一方で、生成された画像に対して、さ

らに手書きで修正してより良い図を作成するシステムを構想していたが、web UI の作成に利用した streamlit の仕様上これの実装が困難であり、現時点では作成できていない。これについては今後の課題とし、作成が終わり次第 GitHub などで公開する。

図 2. 作成した「P5.js LLM Drawing Assistant」の Web インターフェイス

## P5.js LLM Drawing Assistant

テキストプロンプトと手書き入力を使って、P5.jsで図を作成・修正するためのアシスタント。

このアプリケーションでは、テキスト指示またはペン入力による手書きで図形を作成・修正できます。1. 「テキストプロンプト」タブで初期の図を生成します。2. 「手書き修正」タブで図に直接描き込んで修正指示を行います。3. 「履歴」タブで過去の操作を確認・復元できます。

テキストプロンプト 図のプレビュー 履歴

テキストプロンプト

3層のニューラルネットワークを図示せよ。  
入力層は4ユニット、中間層は5ユニット、出力層は1ユニットとせよ。

図を生成

テスト図形を表示

図の生成が完了しました！「図のプレビュー」タブで確認できます

## P5.js LLM Drawing Assistant

テキストプロンプトと手書き入力を使って、P5.jsで図を作成・修正するためのアシスタント。

このアプリケーションでは、テキスト指示またはペン入力による手書きで図形を作成・修正できます。1. 「テキストプロンプト」タブで初期の図を生成します。2. 「手書き修正」タブで図に直接描き込んで修正指示を行います。3. 「履歴」タブで過去の操作を確認・復元できます。

テキストプロンプト 図のプレビュー 履歴

### 生成された図

表示を更新

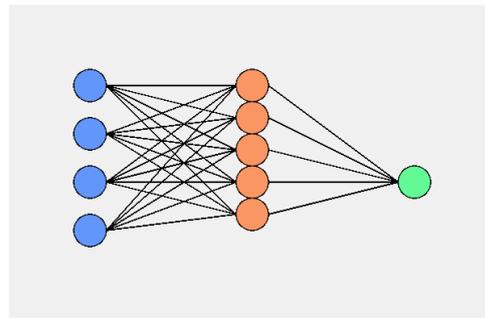
更新カウント:0

P5.jsコードを表示

デバッグモード

### プレビュー

⚠ Streamlitの制限により、図形が表示されない場合があります。その場合は下のダウンロードボタンを使用してブラウザで直接開いてください。



## 9. まとめと大規模言語モデルの現在

本研究の着想は LoRA を用いることで、本研究の申請当時では難しかった LLM による作図支援機構を構築することであった。一方で、この分野の研究は研究者のみならず、様々な企業・組織・個人が精力的に取り組んでいるため、この分野の調査中にも様々な新規モデルが登場した。これらの中には、本研究で求めていた機能の一部を備えた LLM も含まれており、本研究は一度方向性を修正する必要に迫られた。本研究で本来対象としていた matplotlib, plotly, p5, TikZ について高い生成能力を持った、LLM や LLM に基づく作図システムとしては、表 3 のものが有力である。

一方で、ここで列挙した LLM/LLM ベースシステムは大規模なパラメータ数を持ったものであり、10万円以下で買える安価なグラフィックボードや、ノートパソコンに搭載されたグラフィックボードを用いた推論は困難、または推論自体はできるものの非常に時間がかかるものであることに注意したい。本研究では 3B 以下のパラメータの LLM を用いた作図支援システムの開発に挑戦し、結果として実用に耐え得るコード生成能力の獲得には至っていない。

い。しかしながら、この比較的軽量な LLM の分野においても近年では高性能化・高機能化が進んでいる。今後発表される軽量 LLM はほぼ間違いなく今回用いた LLM よりも性能が向上していただろうと推測できるため、今回整備したデータセットを用いたファインチューニングや、今回は見送ったプロンプトチューニングなどの技術、または Mixture of Experts[15]のようなテクニックを用いることで、数ヶ月程度の近い未来に当初計画していたシステムが実装できるだろうと期待している。

表 3. TikZ/p5.js 等の生成が可能な LLM/LLM ベースシステム

名称	作者	概要
AutomaTikZ[7]	Jonas Belouadi, Anne Lauscher, Steffen Eger	TikZ コードを LLM で生成する。
DeTikZify[8]	Jonas Belouadi, Simone Paolo Ponzetto, Steffen Eger	AutomaTikZ の発展版。TikZ コードを手書きで修正可能にした。
DeepSeek-R1[16]	DeepSeek-AI	ゼロショットでの高いコード生成能力を有する。
Claude 3.7 sonnet[17]	Anthropic	ゼロショットでの高いコード生成能力を有する。

## 謝辞

本研究を遂行するにあたり、公益財団法人天野工業技術研究所から多大なご支援を頂きました。ここに記して謝意を示します。本当にありがとうございました。

## 参考文献

- [1] A. Vaswani *et al.*, “Attention is All you Need,” in *Advances in Neural Information Processing Systems*, 2017, vol. 30.
- [2] W. X. Zhao *et al.*, “A survey of large language models,” *arXiv [cs.CL]*, 31-Mar-2023.
- [3] B. Lester, R. Al-Rfou, and N. Constant, “The power of scale for parameter-efficient prompt tuning,” *arXiv [cs.CL]*, 17-Apr-2021.
- [4] E. J. Hu *et al.*, “LoRA: Low-Rank Adaptation of Large Language Models,” *arXiv [cs.CL]*, 17-Jun-2021.
- [5] Q. Zhang *et al.*, “AdaLoRA: Adaptive budget allocation for parameter-efficient fine-tuning,” *arXiv [cs.CL]*, 18-Mar-2023.
- [6] T. Detrmers, A. Pagnoni, A. Holtzman, and L. Zettlemoyer, “QLoRA: Efficient Finetuning of Quantized LLMs,” *arXiv [cs.LG]*, 23-May-2023.
- [7] J. Belouadi, A. Lauscher, and S. Eger, “AutomaTikZ: Text-guided synthesis of scientific vector graphics with TikZ,” *arXiv [cs.CL]*, 30-Sep-2023.
- [8] J. Belouadi, S. P. Ponzetto, and S. Eger, “DeTikZify: Synthesizing graphics programs for scientific figures and sketches with TikZ,” *arXiv [cs.CL]*, 24-May-2024.
- [9] “nllg/datikz-v2 · Datasets at Hugging Face.” [Online]. Available: <https://huggingface.co/datasets/nllg/datikz-v2>. [Accessed: 10-Mar-2025].
- [10] A. Yang *et al.*, “Qwen2 Technical Report,” *arXiv [cs.CL]*, 15-Jul-2024.
- [11] “mattshumer/Reflection-Llama-3.1-70B · Hugging Face.” [Online]. Available: <https://huggingface.co/mattshumer/Reflection-Llama-3.1-70B>. [Accessed: 10-Mar-2025].

- [12] “Qwen/Qwen2.5-Coder-3B · Hugging Face.” [Online]. Available: <https://huggingface.co/Qwen/Qwen2.5-Coder-3B>. [Accessed: 10-Mar-2025].
- [13] A. Asai, Z. Wu, Y. Wang, A. Sil, and H. Hajishirzi, “Self-RAG: Learning to retrieve, generate, and critique through self-reflection,” *arXiv [cs.CL]*, 17-Oct-2023.
- [14] S. Jeong, J. Baek, S. Cho, S. J. Hwang, and J. C. Park, “Adaptive-RAG: Learning to adapt retrieval-Augmented Large Language Models through Question complexity,” *arXiv [cs.CL]*, 21-Mar-2024.
- [15] W. Cai, J. Jiang, F. Wang, J. Tang, S. Kim, and J. Huang, “A Survey on Mixture of Experts,” *arXiv [cs.LG]*, 26-Jun-2024.
- [16] DeepSeek-AI *et al.*, “DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning,” *arXiv [cs.CL]*, 22-Jan-2025.
- [17] Anthropic, “Claude 3.7 Sonnet System Card.” .