

自律型GIS: AIを活用した次世代型GIS

Zhenlong Li*, Huan Ning

[Geoinformation and Big Data Research Laboratory](#)

[\(GIBD\)](#) サウスカロライナ大学地理学科 709 Bull

Street, Columbia, SC, USA 29208

zhenlong@sc.edu

概要 : ChatGPTに代表される大規模言語モデル (LLM) は、人間の自然言語に対する強い理解を示し、推論、創作、コード生成、翻訳、情報検索など、様々な分野で研究・応用されてきた。LLMを推論コアとして採用することで、LLMの自然言語理解、推論、コーディングの一般的な能力を活用し、空間データの自動収集、分析、可視化で空間問題に取り組むAI搭載の地理情報システム (GIS) として、**Autonomous GISを紹介**しています。我々は、自律型GISが5つの自律的な目標を達成する必要があると想定している：自己生成、自組織化、自己検証、自己実行、自己成長。我々は、GPT-4 APIを用いたLLM-GeoというプロトタイプシステムをPython環境で開発し、自律型GISがどのようなもので、**人間の介入なしに期待通りの結果をもたらす**かを3つのケーススタディで実証しました。すべてのケーススタディにおいて、LLM-Geoは、集計された数値、グラフ、地図などの正確な結果を返すことができ、**人手による操作時間を大幅に短縮**することができました。

LLM-Geoは、ロギングやコードテストなどの重要なモジュールが不足しているため、まだ発展途上であるが、**次世代のAI搭載GISへの道筋を示す**ものである。我々は、GIS科学コミュニティが自律型GISの研究開発にさらに力を注ぎ、空間分析をより簡単に、より速く、より多くの人々が利用できるようにすることを提唱する。

キーワード:自律エージェント、GIS、人工知能、空間解析、大規模言語モデル、ChatGPT

本論文に対する著者の貢献は同等であることを宣言する。

投稿日：2023年05月09日

最終更新日：2023年05月28日

1. はじめに

人工知能（AI）の急速な発展により、人間の介入を最小限または全く行わずにタスクを実行し、意思決定を行うことができるコンピュータシステムである自律型エージェント、別名インテリジェントエージェント（Brustoloni 1991）が誕生しました（Wooldridge and Jennings 1995）。自律型エージェントは、ロボット工学、ヘルスケア、輸送、金融など様々な領域で大きな可能性を示している（Russell and Norvig, 2022）。効率性の向上、エラーの減少、人間が行うには時間がかかるか不可能な複雑なタスクを処理する能力など、数多くの利点がある（Stone et al., 2022年）。与えられたコマンド、入力、環境設定に基づいて期待されるタスクを実行できる自動システム（例：自動ドアや自動販売機）とは異なり、自律エージェント（例：掃除機ロボットや自律走行車）は、一般的にもっと自律型エージェントは、複雑で、環境への適応性が高く、収集したデータに基づいて情報に基づいた意思決定を行うことができます。自律型エージェントの実装において最も困難な作業は、意思決定コアを構築することです。意思決定コアは、知覚に従って適切に反応する必要がありますが、知覚と関連する行動は事前にプログラムされていない場合があります。既存のルールやアルゴリズムに基づく意思決定コアの多くは、特定の環境や閉じた環境でしか動作できず、その推論能力も限定的である。

近年の生成モデル、特にGPT-3 (Brown et al. 2020) やGPT-4 (OpenAI 2023) などの大規模言語モデル（LLM）の進歩は、自律型エージェントの研究開発を加速させています。これらのモデルは、人間の自然言語を深く理解していることを示し、推論、創作、コード生成、翻訳、情報検索など、さまざまな分野のタスクを実行することを可能にしました。その結果、LLMはデジタル自律型エージェントの意思決定の中核としてますます利用されるようになり、AIを搭載したシステムの開発が進んでいます。例えば、Vempralaら（2023）は、相互作用、基本的な論理的、幾何学的、数学的推論などの様々なタスクをテストして、ChatGPTをロボット工学に使用しました。研究者はまた、LLMを使用して、事前に訓練された基礎モデル（Liangら 2023; Shenら 2023）に、画像操作（C. Wuら 2023）やキャプション付け（Orlo 2023）など、自然言語を介して人間のタスクを達成するように命令しました。

一方、AutoGPT (Richards, 2023年)、BabyAGI (Nakajima, 2023年)、AgentGPT (Reworkd, 2023年) など、LLMに支えられたデジタル自律型エージェントが開発されています。

NexusGPT（以下、NexusGPT、2023年）は、AIフリーランサープラットフォームとして、様々なデジタル自律型エージェントをリリースし、ビジネスコンサルティングやソフトウェア開発といったサービスを顧客に提供しています。

GIScienceコミュニティでは近年、地理空間研究およびアプリケーションにAIを取り入れる動きがあり、AIとGIScienceの交差に焦点を当てたサブフィールドであるGeoAIへとつながっています（VoPham et al. 2018）。GeoAIは、空間データ処理やマイニングなどの様々な地理空間アプリケーションにAIを活用することで大きな成果を上げているが、空間解析やGISにおける人工一般知能（AGI）（例えば、AGIの初期段階としてのLLM）の探求と採用はまだ初期段階である。Maiら(2023)による一つの注目すべきは、GeoAIのためのLLM的な基礎モデルを開発する機会と課題を探求していることである。最近の人文的側面からのGISの構想では、Zhao（2022）がGISを4つのカテゴリーに分類している：人間、GIS、場所との関係（距離）を考慮し、具象GIS、解釈GIS、自律GIS、背景GISの4つに分類した。Zhaoは自律型GISを「独立したエージェントか場所である」とみなし、ドローンやロボット掃除機、自律走行車、あるいは画像中の土地オブジェクト（建物など）を認識するように訓練されたモデルなどを挙げた。その他、LLMを利用してQGISの単純なデータ操作（読み込みなど）を自動化したり（Kyriakou 2023; Mahmood 2023）、事前に訓練したモデルを採用して画像のセグメント化を行ったりする試み（Q. Wu 2023）などもある。

本論文では、LLMの自然言語理解、推論、コーディングといった一般的な能力を活用し、空間データの自動収集、分析、可視化といった空間問題に取り組むAI搭載GISという意味合いに絞り込み、AIとGISの統合を探るという異なるアプローチをとっています。自律型GISは自律型エージェントと同様のコンセプトを持つが、先に述べた既存のエージェントの多くは、ビジネスレポートや旅行計画など、テキストベースの情報検索、分析、要約に焦点を当て、必ずしもユニークで正しい答えを持っていないものである。これらの設計原理やアーキテクチャは、GISや空間解析にはうまく適応できない。GISは、データ量が多く、一般的に正解が1つしかない（例えば、病院から10km以内に住む人口など）。我々は、自律型GISをプログラミングとデータ中心のフレームワークとして設計し、自動コーディングを使用して決定論的な性質を持つGIS科学の問題に対処することを提案する。

自律型GISの実現可能性を示すために、我々は自律的に空間解析を行うことができるLLM-Geoという概念実証のプロトタイプを開発した。LLM-Geoは、ユーザからタスク（空間問題・質問）を受け取り、そのタスクを有向非循環グラフとして連続した連結データ操作に分解してソリューショングラフ（ジオプロセッシングワークフロー）を生成します。各操作は、LLMが実装すべき機能であり、本研究ではGPT-4である。次に、LLM-Geoはすべての操作のコードを統合して結合プログラムを生成し、タスクの最終結果を生成するために結合プログラムを実行する。LLM-Geoの能力をテストするために、3つのケーススタディが使用されている。その結果、**LLMのGISへの統合は、複雑な空間解析タスクを自動化し、GIS技術をGISのバックグラウンドを持たない個人でも利用できるようにすることで、この分野に革命を起こす可能性がある**ことが示された。プロトタイプ of 自律型GISであるLLM-Geoは、AIを搭載した次世代の自律型GISに向けた潜在的な道しるべとなるものである。

本論文の残りの部分は、以下のように構成されている：セクション2では、AIを搭載した自律型システムとしての自律型GISのコンセプトと設計上の考慮点を詳しく説明する。セクション3では、自律型GISのプロトタイプであるLLM-Geoの実装を紹介し、セクション4では、3つのケーススタディを紹介する。セクション5と6では、LLM-Geoから学んだこと、その限界、そして将来的な研究の方向性について述べている。第7節では、本論文の結論を述べる。

2. AIを搭載した自律型システムとしての自律型GIS

自律型システムは、人間の介入なしに意思決定を行い、タスクを実行するように設計されています。自律システムは、変化する状況に適応し、環境から学習し、収集したデータに基づいて情報に基づいた意思決定を行うことができます。LLM（またはAGI）を、空間問題を解決するための戦略や手順を生み出す意思決定の中核として組み込むことで、自律型GISは、必要な空間データを既存のオンライン地理空間データカタログから検索して取得するか、センサーから新しいデータを収集し、既存の空間アルゴリズム、モデル、ツールを使用して（または新しいものを開発して）、収集データを処理して最終結果（地図、チャート、またはレポートなど）を作成できるようになる。LLMは自律型GISの「頭脳」、あるいは環境センサーを搭載した場合の「頭部」であり、実行プログラム（Pythonなど）はそのデジタル「手」と考えることができます。他の自律エージェント（Sifakis 2019）と同様に、自律型GISには、意思決定（LLMを中核とする）、データ収集、データ操作、操作ログ、履歴検索の5つの重要なモジュールが必要であることを提案します。これらのモジュールにより、GISは、自己生成、自己組織化、自己検証という5つの自律的な目標を達成することができる、自作自演、自作自演（表1）。

詩を書くような、これらの答えを評価するための厳格な基準がない非決定論的なタスクのための多くの自律的エージェントとは異なり、GIScienceアプリケーションでは、答えを出すために空間データの定量的な計算と分析が必要です。GISのような決定論的システムとは、システムの初期条件とその挙動を支配する根本的なルールに関する十分な情報があれば、システムの将来の状態を正確に予測することができるものである。決定論的なシステムでは、初期条件と結果の間に直接的な関係があり、ランダム性や不確実性が入り込む余地はない。この意味で、自律型GISは、厳密に制御可能で説明可能な答えのアプローチを必要とし、そのような答えは、与えられたデータ、例えば、病院から10キロメートル以内に住む人口に基づいて、ユニークで定量的に正しいものでなければなりません。したがって、自律型GISはプログラミング中心のフレームワークとして設計され、自動コーディングを使用して決定論的な性質を持つGISサイエンスの問題に対処することが望ましいと考えられる。

表1 自律的な目標と自律的なGISのモジュール

自律的な目標	モジュールの巻き込み	機能性
自己発電	意思決定、データ収集	ソリューションとデータ運用プログラムを生成する
セルフオーガナイザー	意思決定、オペレーションロギング	業務が正しい順序で実行され、データが適切な方法で保存されていることを確認する。
セルフベリフィケーション	意思決定、データ運用	生成されたワークフロー、コード、プログラムのテストと検証
セルフエグゼキューション	データ操作、操作ロギング	生成されたワークフロー、コード、またはプログラムを実行する。
セルフグロウイング	操作ロギング、履歴検索	検証された操作の再利用

3. LLM-Geo: 自律型GISのプロトタイプ

我々は、LLM-Geoに自律型GISの重要なモジュールである意思決定とデータ操作の2つを実装し、3つの自律的な目標（自己生成、自己組織化、自己実行）を達成しました。現在、追加モジュールを開発中である。意思決定モジュールは、LLM（本研究ではGPT-4）をコア、つまり「頭脳」として採用し、様々な空間的問題に対処するためのステップバイステップの解決ワークフローを生成し、各ステップの関連コードを開発します。データ操作モジュールは、空間データのロード、処理、可視化、保存など、生成されたコードを実行するためのPython環境である。

図1は、LLM-Geoがどのように質問に答えるかの全体的なワークフローを示しています。このプロセスは、ユーザーが空間的な質問と、オンラインデータのURL、REST (REpresentationalStateTransferサービス、API(ApplicationProgramInterfaceドキュメント)などの関連データの場所を入力するところから始まります。次に、LLMはジオプロセッシングのワークフローに似たソリューショングラフを生成します。ソリューショングラフに基づき、LLM-Geoは各操作ノードの要件をLLMに送り、コードの実装を要求する。次に、LLM-Geoは、すべての操作コードの実装を集め、ワークフローに基づいて操作を接続するアセンブリプログラムを生成するようLLMに依頼する。最後に、LLM-Geoがアセンブリプログラムを実行し、最終的な回答を作成します。

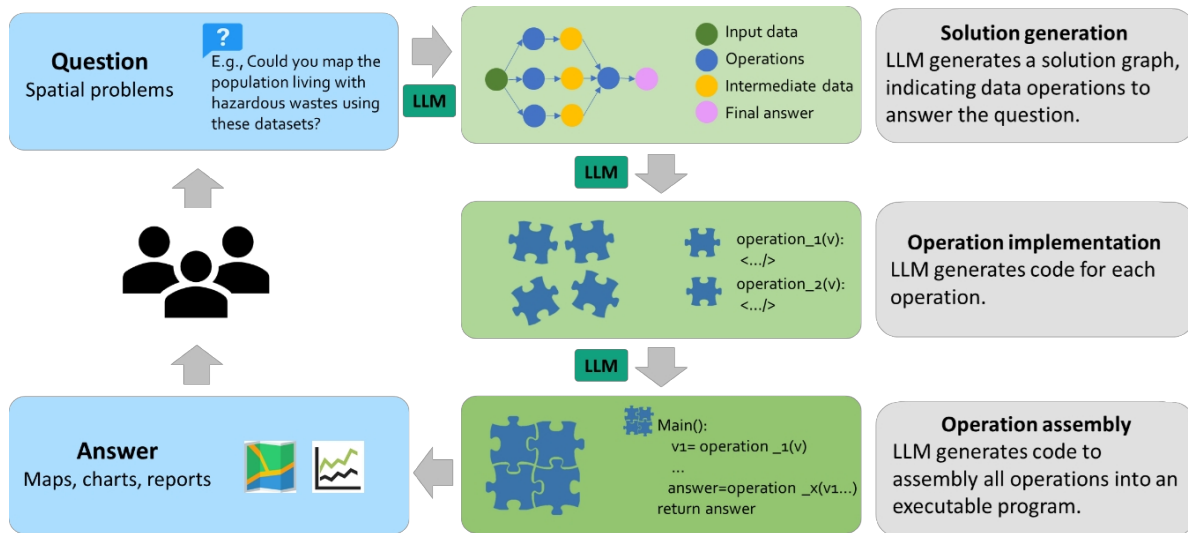


図1.LLM-Geoの全体的なワークフロー

3.1 ソリューションジェネレーション

LLM-Geoにおけるソリューションとは、一連の接続された操作とデータからなるデータ処理のワークフローを指します。これらの操作を実行してデータを連続して処理することで、地図、チャート、テーブル、新しい空間データセットなど、問題に対する最終結果が生成されます。ワークフローを表現するために有向グラフを使用し、すべてのノードをデータと操作の2つのカテゴリに分類します。データノードとは、操作の入力データまたは出力データを指し、入力データノード、中間データノード、出力データノードの3種類からなる。入力データノードは、ローカルデータパス、URL、またはプログラムによるデータアクセスのためのREST APIを通じて、入力データをロードする場所をシステムに指示します。出力ノードは、タスクや質問の最終結果で、数値データ、インタラクティブまたは静的なマップ、テーブル/チャート、新しいデータセット、その他ユーザーが要求するタイプになります。他のすべてのデータノードは、中間データノードとみなされます。操作ノードは、入力と出力でデータを操作するためのプロセスである。その入力、入力データ・ノードまたは祖先の操作ノードからの中間データ・ノードとすることができます。出力は、子孫の操作ノードの中間データノードか、出力ノード（最終結果）である。有向エッジはデータの流れを示す。最終結果を得るためには、すべてのデータフローを出力ノードで収束させる必要があるため、解グラフでは切断されたノードは許されない。

これらの解グラフの構造制約は、ユーザーからの質問とともにAPI経由でLLM（GPT-4）に供給される。空間解析は、基本的に一連の空間データ処理タスクからなるジオプロセッシングワークフローであるため、ソリューショングラフは、データが処理ステップをシームレスに流れ、最終的に最終結果に収束することを保証します（図2）。付録1は、ソリューショングラフのプロンプトとLLMが返すコードのサンプルである。

我々の実験によると、操作ノードの粒度は、質問の複雑さとサポートされる最大トークン長に基づいてLLMによってオンザフライで決定されることがわかった。トークン長の制限により、LLMは複雑なタスクを詳細なグラフに分解することが困難な場合があり、その結果、粒度が低く、ノード数が少ない解答グラフになる。現在のLLMはすべてトークン長に制限があるため（GPT-4などの最先端モデルも含む）、この実用的な制約により、複雑なタスクに対して再帰的なアプローチが必要になります。このアプローチでは、正確なコード生成と再利用のために粒度が適切なレベルに達するまで、操作ノードを、より粒度の細かい操作を含むサブソリューショングラフにさらに分解することができます。

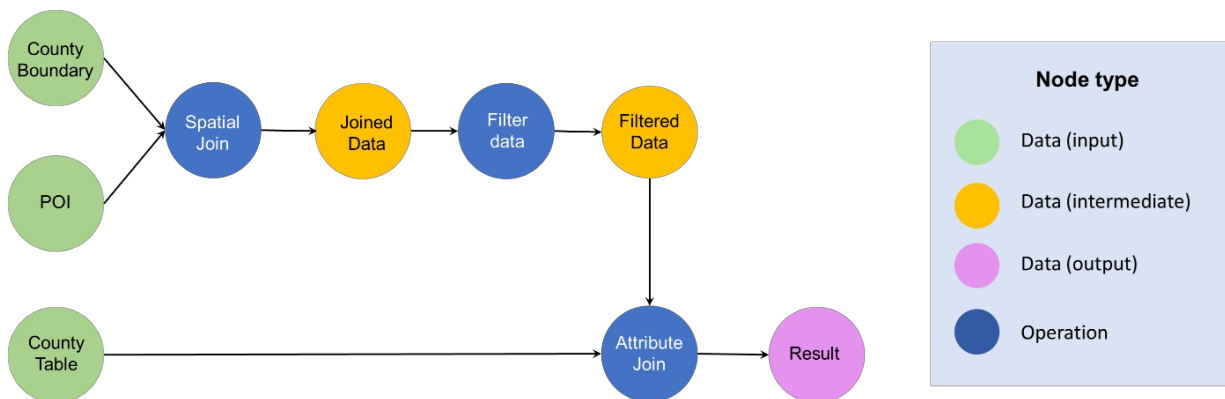


図2.異なるノードタイプを持つソリューショングラフの図解

3.2 オペレーション実施

生成されたソリューショングラフは、指定された空間問題に対する「データ処理計画」として機能し、入力データ、出力データ、データ操作があらかじめ定義されている。操作の実装段階では、ソリューショングラフの操作定義に従って、各操作が実行可能なコードスニペットとして実装されることになる。LLM-Geo では、データ操作モジュールとして Python ランタイムを使用しているため、操作の実装は LLM による Python コードの生成となります。LLM-Geo では、操作ノードとデータノードが隣接するノードに接続するように、ノード間のインターフェイスを作成するアルゴリズムを使用しています。例えば、LLM-Geo の現在のバージョンは、各操作に対して Python 関数を生成する。関数の定義と戻りデータ（関数の名前と関数の入出力変数）は、ソリューショングラフにあらかじめ定義されている。この戦略は、コード生成の不確実性を低減するために使用されます。

我々の実験によると、GPT-4は信頼性の高いコード生成のために十分な情報と特別なガイダンスを必要とする。この文脈では、現在の操作ノードに先行する解グラフパスに関連するすべての情報が必要となる場合がある。例えば、祖先ノードが新しい列やファイルを作成することがあり、LLMは現在の操作ノードの関数コードを生成する際に、その列やファイルの名前を知って使用しなければならない。さらに、コード生成プロセスをガイドするために、子孫ノードに関する情報も必要である。そこで、LLM-Geo は、祖先ノードで生成されたコードと子孫ノードの情報を LLM に送り、各操作ノードのコード実装を要求する。ノード情報以外にも、特別なガイダンスが必要である。例えば、GPT-4 は、テーブル結合のための同じ列データ型や、オーバーレイ解析のための同じ地図投影法など、いくつかの空間データ操作の前提条件について記憶が曖昧であることがわかった。また、GPT-4は空間データ処理用のPythonライブラリであるGeoPandasの使い方を習得しているが、このライブラリがオンザフライ再投影に対応していないことを知らないようである。このような場合、LLMが操作ノードに対して正しいコード実装を行うための特別なガイダンスが必要である。表 2 は、LLM-Geo が操作の実装段階で使用したガイダンスの一部である。付録 2 は、操作のプロンプトと LLM が返すコードのサンプルです。

表2 操作コード生成のガイダンス例

1	与えられた変数名やパスは変更しないでください。
2	返信はPythonのコードブロック（`pythonと`で囲む）に入れて、コードブロックの外に説明や会話をしないでください。
3	GeoPandasを使用してURLから圧縮されたESRIファイルを読み込む場合、ESRIファイルを解凍せずに直接ファイルを読み込んでください。 例) <code>gpd.read_file(URL)</code>
4	入力引数、出力引数の記述を生成する。
5	関数からデータを受け取る必要があります。他の関数が事前にデータをロードして返している場合は、関数でロードしないでください。
6	モジュール'pandas'には属性'StringIO'がないことに注意。
7	最新のPythonモジュールのメソッドを使用する。
8	空間解析を行う場合は、関係するレイヤーを同じ地図投影に変換してください。
9	テーブルを結合する場合、関係するカラムを先頭のゼロを除いた文字列型に変換してください。
10	空間結合を行う場合、結果の重複を削除してください。または、削除する必要があるかどうか考えてください。

3.3 オペレーションアセンブリ

LLM-Geo は、すべての操作ノードのコード（関数）を生成した後、コードを収集し、ソリューショングラフと事前に定義されたガイダンスとともに LLM に提出し、タスクの最終プログラムを作成する。ガイダンスに基づき、LLMはソリューション・グラフを使用して、最終プログラムにおける操作ノードの実行順序を決定します。中間変数は、操作ノードの出力データを格納するために生成され、後続の操作ノードに供給される。最後に、LLM-Geoはアセンブリプログラムを実行し、空間問題の最終結果を生成します。GPT-4 API に基づく LLM-Geo の全体の実装構造とワークフローを図 3 に示す。**付録 3** には、アセンブリプログラムのプロンプトと LLM が返すコードのサンプルが含まれています。

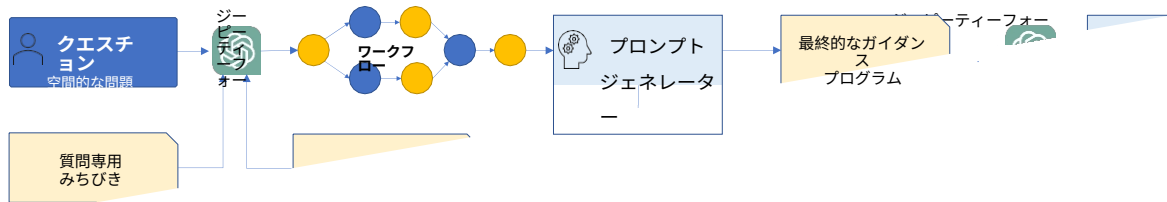


図3.GPT-4 APIを用いたLLM-Geoの実装ワークフロー

4. 事例紹介

4.1 ケース1: 有害廃棄物周辺に住む人口をカウントする。

この空間問題は、有害廃棄物に囲まれて暮らす人口を調べ、その分布を地図にすることである。調査地域は、アメリカ合衆国（US）のノースカロライナ州である。タスク（課題）をLLM-Geoに入力すると、Box 1のようになる。

課題:

- 1) 有害廃棄物施設を含む国勢調査地区内に居住する総人口を調べることができます。調査対象地域は米国ノースカロライナ州。
- 2) トラクトレベルの人口の空間分布を示す地図を作成し、有害廃棄物施設を持つトラクトの境界を強調表示する。

データの場所:

1. NC 危険廃棄物施設 ESRI シェイプファイルの場所:

https://github.com/gladcolor/LLM-Geo/raw/master/overlay_analysis/Hazardous_Waste_Sites.zip

2. NC 範囲の境界シェイプファイルの場所:

https://github.com/gladcolor/LLM-Geo/raw/master/overlay_analysis/tract_shp_37.zip。トラクトIDのカラムは「範囲」です。

3. NC 範囲人口CSVファイルの場所: https://github.com/gladcolor/LLM-Geo/raw/master/overlay_analysis/NC_tract_population.csv.

人口は 'TotalPopulation' カラムに格納されています。トラクトIDカラムは'GEOID' です。

ボックス1. ケース1のLLM-Geoに提出された空間問題。

この質問は、LLM-Geo に、危険な廃棄物とともに暮らす総人口を調べ、データの場所や使用する列名など、分析に十分な詳細を含む人口分布のマップを生成するよう求めるものです。ベクトルレイヤー（有害施設とトラクト境界）は同じ地図投影ではなく、範囲IDのデータ型は境界レイヤーではテキストですが、Pandasが人口CSVファイルを読み込む際には整数となることに注意してください。

また、境界層ではトラクト ID のデータ型はテキストですが、Pandas が人口 CSV ファイルを読み込むと整数になります。GPT-4 はこれらの不整合を前処理する記憶が曖昧なので、LLM-Geo はプロンプトに特別なガイダンスを入れて GPT-4 に思い出させる必要があります：「空間解析を行う場合、関係するレイヤーを同じ地図投影に変換する。空間解析を行う場合は、関係するレイヤーを同じ地図投影に変換する。"テーブルを結合する場合は、関係するカラムを先頭ゼロのない文字列型に変換する。"

図 4 は、このケーススタディにおける LLM-Geo の出力です。データ処理ステップを詳述したソリューショングラフ、最終的なアセンブリ Python プログラム、および質問に対する最終結果（有害廃棄物施設を含むトラクト内に住む総人口（5,688,769）、国土地理院レベルの人口の空間分布を示すマップ、有害廃棄物施設を持つトラクトの境界線を強調する）が示されています。手作業による検証で、この数字と地図の正確性を確認しました。LLM-Geo から生成された作業用コードを使えば、ユーザーは必要に応じて、地図の可視化スタイルをカスタマイズするために、簡単にコードを調整し、再実行することができます。

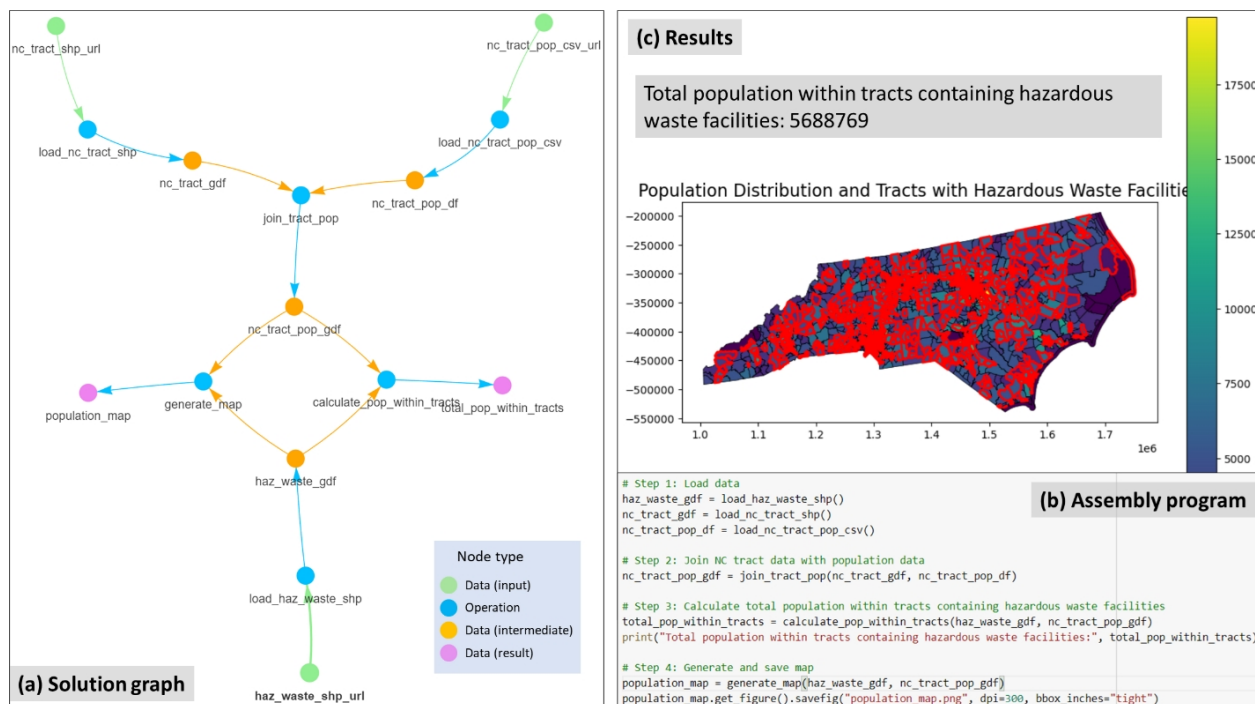


図4. LLM-Geoが危険廃棄物周辺に住む人口をカウントするために自動生成した結果。(a) ソリューショングラフ、(b) アセンブリプログラム (Pythonコード)、(c) 返された人口カウントと生成された地図。

4.2 事例2: 人の移動に関するデータ検索と傾向の可視化

この課題では、2020年にフランスで発生したCOVID-19パンデミック時のモビリティ変化を調査している。まず、LLM-GeoにREST APIを用いてODT Explorerからモビリティデータを取得し（Li et al. 2021）、2020年1月と比較した月間変化率を計算・可視化するように依頼しました。ボックス2に示すように、LLM-Geoに課題を入力した。図5は、このケーススタディに対するLLM-Geoの結果であり、データ処理ステップの詳細を示すソリューショングラフ、移動変化率の空間分布を示すマップマトリックス、移動変化率の傾向を示すラインチャート、および出力を生成した最終アセンブリプログラムを含む。すべての結果は、私たちの手作業による検証で正しいものでした。

課題:

- 1) 各行政区域の人口移動の月別変化率をフランス地図で表示します。各月はマップマトリックスのサブマップである。変化率のベースは2020年1月です。
- 2) 全行政区域の月別変化率の推移を折れ線グラフにする。X軸は月である。

データの場所:

1. フランス行政区用ESRIシェープファイル: https://github.com/gladcolor/LLM-Geo/raw/master/REST_API/France.zip.GID_1 カラムは行政区コード、「NAME_1」カラムは行政区名です。
2. モビリティデータアクセスのためのパラメータを含むREST API URL:
http://gis.cas.sc.edu/GeoAnalytics/REST?operation=get_daily_movement_for_all_places&source=twitter&scale=world_first_level_admin&begin=01/01/2020&end=12/31/2020.
レスポンスはCSV形式です。
レスポンスには、place、date (format:2020-01-07), intra_movementの3つのカラムがあります。
「place」カラムは行政地域コードで、フランスの行政地域は「FRA」で始まる。

4.3 事例3: COVID-19による米国郡レベルの死亡率解析と可視化

本件の空間問題は、米国の郡レベルにおいて、COVID-19死亡率（症例に対するCOVID-19死亡者の割合）の空間分布と、死亡率と高齢者（65歳以上）の割合の関連性を調査することである。死亡率は、New York Times（2023年）から入手できる2020年12月31日時点の累積COVID-19データから、州および地方保健機関の報告に基づいて算出したものです。人口データは、2020年ACS5年推計（US Census Bureau 2022）から抽出される。課題では、郡レベルの死亡率分布を示す地図と、死亡率と高齢者居住率の相関と傾向線を示す散布図が求められている。ボックス3は、LLM-Geoに提出された必要なデータセットの場所とともに、空間問題を示している。結果は、図6に示されています。

課題:

- 1) COVID-19の死亡率(死亡/症例)をアメリカ合衆国のカウンティ(郡)間で示す地図を作成する。
2020.12.31のCOVID-19の累積データを使って死亡率を計算する。地図のプロットにはscheme='quantiles'を使用する。地図の投影法を「コヌス・アルバース」に設定する。地図の大きさを15*10インチに設定する。
- 2) 死亡率と高齢者居住率の相関と傾向線を示す散布図を描き、 r^2 乗とp値を含める。データポイントの透明度を50%、回帰線を赤に設定する。図形の大きさを15*10インチに設定する。

データの場所:

- 1) 2020年のCOVID-19データ症例（郡レベル）：<https://github.com/nytimes/covid-19-data/raw/master/us-counties-2020.csv>。このデータは、アメリカの各郡のCOVID症例と死亡を毎日累積したものです。日付（フォーマット：2021-02-01）、郡、州、fips、症例、死亡の5カラムがあります。
 - 2) Contiguous US county boundary (ESRI shapefile):
https://github.com/gladcolor/spatial_data/raw/master/contiguous_counties.zip。郡のFIPSカラムは「GEOID」です。
 - 3) 国勢調査データ (ACS2020) :
https://raw.githubusercontent.com/gladcolor/spatial_data/master/Demography/ACS2020_5year_county.csv。必要なカラムは以下の通り：「FIPS」、「総人口」、「総人口: 65歳から74歳」、「総人口: 75~84歳」、「85歳以上」。
-

ボックス 3. ケース3のLLM-Geoに提出された空間問題

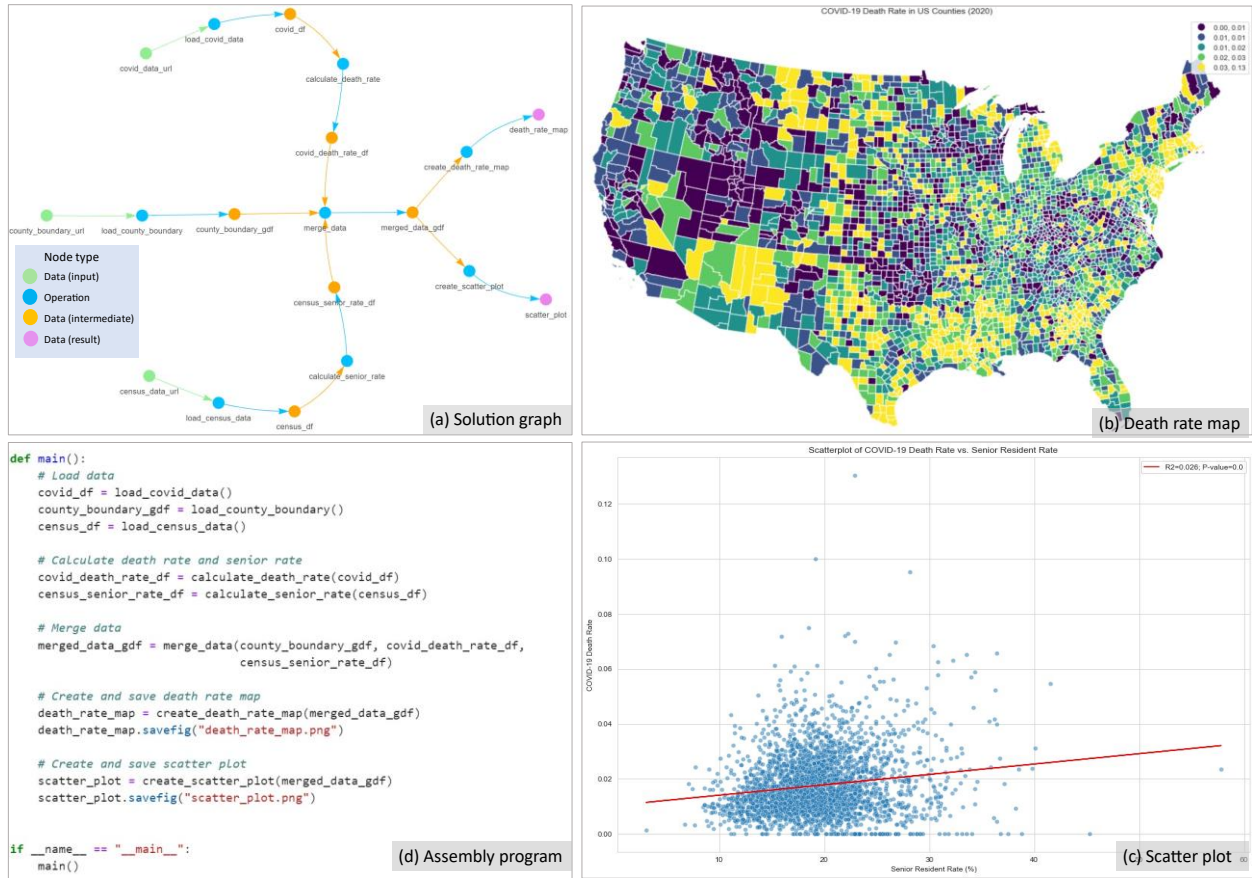


図6. 米国の郡レベルのCOVID-19死亡率の解析と可視化についてLLM-Geoが自動生成した結果
 (a) 解析グラフ、(b) 米大陸の郡レベルの死亡率マップ、(c) 郡レベルのCOVID-19死亡率と高齢者率の関連を示す散布図、(d) 組み立てプログラム。

この例では、「地図をプロットするときは、スキーム = 'quantiles' を使用する」など、地図やグラフの可視化スタイルに関する要件をより多く提示しました。地図の投影法を 'Conus Albers' に設定する。地図の大きさを 15*10 インチに設定する」。このようなガイダンスがないと、システムは同じ問題のリクエストごとに異なる設定を選択することが多い。生成された実行プログラム全体は、付録4に掲載されています。コード内の日付「2020-12-31」を変更してプログラムを再実行すれば、他の期間のCOVID-19の死亡率分布や高齢者居住率との関連性を簡単に調べることができます。

5. ディスカッションと教訓

5.1 自律型GISの形態

自律型GISは、空間解析の基礎や手順を変えるのではなく、人間の操作を模倣することを目的としています。この意味で、自律型GISは、ユーザーから質問を受け、回答を生成するGISアナリストとして機能するスタンドアロンアプリケーション（ローカルまたはクラウドベース）として機能することができます。このようなアプリケーションのグラフィックユーザーインターフェース（GUI）の1つの可能性は、図7（a）に示されており、ユーザーは空間問題の説明を入力し、送信ボタンをクリックすると、アプリケーション内で直接結果を受け取ることができます。また、モニタリング、デバッグ、カスタマイズのために、生成されたソリューショングラフ（ジオプロセッシングワークフロー）と最終結果を生成する実行コードを表示することができます。自律型GISの能力は、生成されたコードやワークフローを蓄積・再利用することで急速に自己成長することができ、特に多くのユーザーにサービスを提供するクラウド版ではその傾向が顕著です。

自律型GISは、スタンドアロン型に加えて、従来のGISソフトウェアのコパイロットとして、自然言語を使ってユーザーとコミュニケーションし、空間データ処理と分析タスクを自動化することができます。この設定は、レポート作成やスライド作成などのオフィス作業を自動化するMicrosoftのOfficeファミリーのCopilotに似ています（Microsoft 2023）。

例えば、マップビューと並んだ自律型GISパネルをArcGISやQGISに統合し、図7（b）に示すように、チャットボックス、生成されたソリューショングラフ、コードを表示することができます。結果は、内蔵のマップビューに表示されます。ユーザーは、ソリューショングラフや操作のパラメータを編集することで、ワークフローを変更することができます。生成されたコードによってオペレーションが実装されている場合、ユーザーはソリューショングラフのノードをクリックすることで、コードを検索し編集することができます。全体として、ソリューショングラフはArcGISのモジュールビルダーやQGISのモデルデザイナーに似ています。成熟したGISプラットフォーム（ArcGISなど）には、すでに豊富な操作（ArcGISツールボックスのツールなど）と確立されたドキュメントがあり、LLMがすぐに学習してソリューショングラフの生成に使用できるため、既存のGISプラットフォームに基づく自律型GISの実装は、現時点で最も実用的かつ効率的なアプローチであると考えられる。

より広い意味では、自律型GISは、COVID-19や他の疾患の公衆衛生監視ダッシュボードなど、地理空間データ操作を必要とするアプリケーションと統合された「プラグイン」として使用することができます。このような自律型GIS対応ダッシュボードは、自然言語を使用してユーザーと対話し、ユーザーの特定のニーズや好みに基づいてカスタマイズされたマップ、チャート、レポート、新しいデータセットを生成する機能を備えています。このような対話とカスタマイズのレベルにより、ユーザーは、さまざまなアプリケーションで地理空間データを扱う際に、よりパーソナライズされた効率的な体験をすることができるようになります。

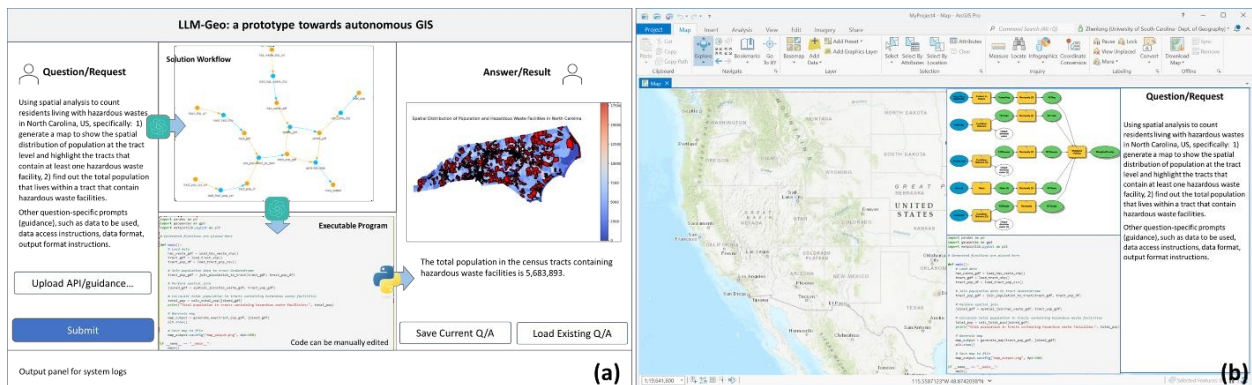


図7.(a)スタンドアロン型自律型GIS、(b)既存のGISソフトウェア（例：ArcGIS Pro）に統合したコパイロット型自律型GISのユーザーインターフェースの説明図である。注：この図の質問、ワークフロー、コード、結果は、デモンストレーションのために作られたものです。

5.2 自律型GISとAIアシスタントの違い

LLMに裏打ちされた様々なAIアシスタントが市場に出回っています。例えば、「Cursor」(カーソル2023)や「GitHub Copilot X」(ギットハブ2023)は、LLMを活用してコードやコメントを生成し、プログラマーの生産性を高める類似製品である。マイクロソフトのOfficeファミリー向けCopilotは、レポート作成やスライド作成などのオフィス作業を自動化することができます。自律型GISがこれらのAIアシスタントと異なるのは、入力、最終製品、そして結果を生み出す決定論的でデータ中心的なワークフローにある。AIアシスタントは、テキストを受け取り、段落やコードといったテキストベースのコンテンツをユーザー向けに生成することが多い。一方、GIS分析の入力は空間データであり、それをさらに処理して定量的な結果(地図やチャートなど)を導き出し、意思決定を支援する。

さらに、GISユーザーは通常、分析目的、経験、知識に基づいて、空間データとGISツールを含むジオプロセッシングワークフロー(Scheider et al.2021)を作成する。これらのワークフローは、スクリプト、モデルビルダーを使用してデータと操作をグラフィカルに接続したものの、またはGISソフトウェアでの一連の手動ステップの形をとることができます。これらのワークフローは決定論的であり、データ中心であり、他の人が複製することができる。自律型GISプロトタイプ(LLM-Geo)の設計は、まずデータ中心のジオプロセッシングワークフローを生成し、プログラミングでワークフローを実装し、ワークフローを実行して最終結果を自動生成するという、同様の問題解決アプローチに従っています。ここで重要なのは、LLMは空間データを直接操作するのではなく、データを扱うための操作(プログラム)を生成して命令することである。

5.3 分断統治

なぜGPT-4のようなLLMに解答グラフや演算を生成せずに解答を求めるという、より直接的なアプローチを選択しないのかという疑問が生じるかもしれない。実際、我々のテストでは、プロンプトに特別なガイダンスを埋め込むことで、GPT-4は比較的単純な空間解析タスクに対して正しいコードを生成できることが示されました。しかし、この直接的な方法では、スマートフォンの移動データAPIとウェブクロールを使って国レベルで自閉症介入サービスのアクセシビリティを評価したり、特定の海域で最も有望な漁場を特定するなど、より複雑なタスクに取り組むLLMの能力を制限する可能性があります。これらのタスクは長編小説を書くようなもので、LLMが大規模なプログラムを含む包括的でワンステップなソリューションを生成することは考えられません。このようなタスクは、人間だけでなくモデルにとっても挑戦的である（前田, 2023）。

人間の問題解決は、複雑なタスクに直面したとき、しばしば分割統治戦略を採用する。LLM-Geoの設計でもこのアプローチを採用することで、複雑な問題をLLMが扱えるより小さな、管理しやすいサブ問題（すなわち操作）に分割することで空間解析タスクに対処することを意図しています。そして、これらの小問題を体系的に処理し（=小問題ごとの機能開発）、最終的にこれらの機能を組み合わせて（=プログラムの組み立て）、最終的な結果を得ることができます。この分割統治手法は、AutoGPT（Richards 2023）やAgentGPT（Reworkd 2023）など、他のデジタル自律型エージェントでも採用されている。私たちは、これまでに実証されたタスクよりも複雑なタスクで、このアプローチの実現可能性をさらに追求する予定です。さらに、分割統治戦略は、検証された操作ノード（例えば、コードスニペット）を生成するのに役立つ。これらの検証されたサブソリューション（操作）は、自律型GISの貴重な資産となり、将来のタスクに再利用することができるため、自律型の目標である自己成長を強化することができます。さらに、このアプローチでは、操作ノードのエラー位置の特定が容易になり、必要なときに人間が介入しやすくなる。

5.4 GISとLLMにおける十分な情報量の必要性

先に述べたように、GISは決定論的な性質を持っており、その出発条件と運用を支配するルールについて十分な知識があれば、将来の状態を正確に予測することができる。このような決定論的なシステムでは、結果は初期条件と本質的に結びついており、ランダム性や不確実性が入り込む余地はない。決定論的システムにおいては、システムの挙動を正確に予測し、理解するために、十分な情報が不可欠である。GISの場合、空間解析は通常、データの利用可能性、管轄の境界、地図のサイズなどの制約に直面する。さらに、レイヤープロジェクション、フィールド名、データタイプなどのデータメタデータや詳細は、その後の空間プログラミングに不可欠です。人間のアナリストと同様に、自律型GISも、与えられたタスクをうまく達成するために、これらの制約と情報を認識する必要がある。例えば、カラム名の事前知識なしに属性結合操作を行ったり、レイヤープロジェクションの情報なしにオーバーレイ分析を行ったりすることはできない。したがって、十分な情報の提供は、自律型GISの正確な推論、タスク計画、行動実行に必要かつ十分な条件であると考えられる。

情報提供の方法としては、データサンプリング、ウェブ検索、自省、手書きなどが考えられる。

5.5 LLMのうろ覚えの記憶を呼び起こす

LLMは、追加情報を取り入れることで、以前のエラー出力を修正することができます。LLMは、タスクを完了するために必要な手順を理解しているが、オーバーレイ解析のための地図投影の一致や、2つの属性テーブルの共通結合列の正確なデータ型の一致など、空間解析における実用的な制約をしばしば見落とす。このようなLLMの「うろ覚え」は、おそらく限られたGISトレーニング教材に起因していると思われます。LLMがこのようなうろ覚えの記憶を呼び起こし、推論とコーディングを成功させる能力を向上させるためには、タスク記述の中に必要なリマインダーを組み込むことが重要である。LLMとトレーニングデータセットが異なれば、空間解析の記憶も異なる可能性があるが、自律型GISが正確な結果を出すために、比較的普遍的な注意事項リストを作成することは可能である。このリストは、GISユーザーのためのチェックリストに似た機能を持つ。例えば、空間解析タスクにおいて、GISユーザーは、共通の結合カラムが同じデータ型（文字列またはint）および先頭のゼロ桁を持つことを確認しなければならない。経験豊富なGISユーザーは、通常、これらの前提条件を事前に確認しますが、初心者は、エラーが発生するか、チェックリストを参照するまで、これらの手順を見落とすことがあります。このガイダンス（チェックリスト）の具体性は、ユーザーによって異なり（例えば、経験豊富なGISユーザーには短い）、また自律型GISでは、採用するLLMの洗練度によって異なる。

6. 限界と今後の課題

LLM-Geoは自律型GISの概念を検証していますが、生成されたものをデバッグできない、ラスターデータを処理できないなど、多くの制約があり、まだ発展途上の段階にあると言えます。LLMの推論機能の登場は、GISサイエンスのコミュニティに新しい時代の幕開けをもたらしますが、まだ探求すべき機会や取り組むべき課題の広大な風景が残っているのが現状です。我々は、LLM-Geoの既存の制約に取り組み、将来的に完全に動作する自律型GISのための基礎を築くために、さらなる研究と開発のためのいくつかの潜在的な道を特定した。

6.1 LLM-Geoの適応性を向上させる必要がある。

我々は、LLM-Geo をテストするために多くのケーススタディを開発し、セクション 4 で GPT-4 に基づく成功率約 80% の 3 つの典型的なケースを報告した。我々の観察によると、生成された解のグラフは典型的に正確であった。失敗の大部分は、操作の実装段階での欠陥コードに起因するものであり、分割統治アプローチであっても、1つのエラー文がプログラム全体をクラッシュさせることがあります。GPT-4は通常、空間解析に広く使われているGeoPandasライブラリ（Jordahl et al.2020）を利用しているが、一度の試行で正しいコードを生成することに苦労することが多い。さらに、Plotly(2023)などの他のライブラリを使用すると、エラーの発生率が高くなる。

現在、LLM-Geoには、生成されたコードをレビュー、テスト、デバッグ、および検証するメカニズムが欠けています。これらのモジュールは、自己検証という自律的な目標を達成するために必要である。予備的な実装として、コードとデバッグのエラーをLLMに直接プッシュし、修正されたコードを要求することができる。また、LLM-Geoでは、従来よくできたGISパッケージやツールボックス（QGISやArcGISなど）で既に利用可能な空間演算（例：最近接距離計算）を、GeoPandasや他のオープンソース空間ライブラリに基づいてゼロから複雑な関数を開発する傾向があるとのことである。このようなその場で生成される複雑な関数は、適切なガイダンスがあったとしても、エラーが発生しやすいものです。現在のコード生成アプローチと比較して、LLM-Geoは、コードのデバッグと検証モジュールと組み合わせて、確立されたテストされたGISツールボックスを利用することで、より適応的で堅牢なものになると考えています。もう一つの限界は、LLM-Geoが地図投影、データ型、属性などのデータを知ることが

できないことである。LLMとコードおよびデータとの間のこれらの相互作用を自動化された方法で管理するための洗練されたメカニズムを設計する必要があります。

6.2 メモリーシステムが必要

すべての自律システムは、将来の検索や再利用のために、文脈的で長期的な情報、データ、および結果を保存するためのメモリコンポーネントを必要としますが、この機能は現在 LLM-Geo に欠けています。LLM-Geoは、プロンプトとLLMの応答を含むコンテキスト・メモリを保存する簡単な方法を使用していますが、これは自律型システムには不十分です。自律型GISの記憶には、データソース、あらかじめ定義された解析ガイダンス、以前のタスクで生成された検証済みのコードなどがあります。これらの記憶要素は、自己成長する自律的な目標を達成するために、時間の経過とともに拡張されることが期待される。LLMベースの自律型エージェント（AutoGPTなど）の多くは、会話の保存と検索にPinecone（Pinecone 2023）のようなベクトルデータベースを使用します。自律型GISは、テキストの保存と検索以上に、複雑なタスクを再帰的に解決するために階層化されるであろうソリューショングラフ（ジオプロセシングワークフロー）を保存する必要があるという、より複雑な要件に直面しています。また、検証された解答や操作コードは、記憶すべき長期的な情報の一部となる。このように、*自己成長*という自律的な目標を達成するためには、ロギングモジュールとともに記憶システムが必要である。

6.3 GISコミュニティによって維持されるカテゴリ化されたガイダンス

GPT-4の記憶が曖昧であることを認識しながらも、我々はそれを補足し、データ操作のための正確な操作ノードを生成する方向に誘導することに成功した。GDAL/OGRのようなオープンソースパッケージのドキュメントと同様に、LLMのためのGISガイダンスはGISアナリストの経験を含み、空間分析の自動化において重要な役割を果たすLLMのためのハンドブックとして機能します。現在のガイダンスは、LLM-Geoの開発を通じて更新されています。しかし、いくつかの事例から要約されたこのようなガイダンスは、最も一般的な空間解析シナリオをカバーするには程遠いものです。例えば、ケーススタディには、GISアプリケーションで典型的なラスターデータ解析は含まれていない。我々は、GISコミュニティが、LLMの空間プログラミング（コード生成）能力を向上させるためのGISガイダンスを提供することを求める。このようなガイダンス文書は長くなる可能性があるため、分類しておく必要がある。例えば、ベクターデータ、ラスターデータ、ネットワークデータ、ジオビジュアライゼーションの解析のための別々のガイダンスドキュメントを用意することができる。自律型GISの記憶システムは、ガイダンスをすべて読むのではなく、特定の作業に対してのみ必要なガイダンスを取り出すように設計する必要がある。また、自律型GISの試行錯誤に基づく空間解析ガイダンスのまとめ方、例えば、強化学習を用いて、LLMに様々な解析タスクを生成・実行させてガイダンスを生成させる方法を探ることも研究者の仕事です。

6.4 試行錯誤、より堅牢な問題解決アプローチ

現在の LLM-Geo の実装では、必要な情報がすべて LLM に入力されれば、正しい結果が返されることを想定しています。空間解析は、データソース、アルゴリズム、仮説が複雑であることが多いため、この期待はやや理想的です。実際には、解析者は試行錯誤的な問題解決戦略を採用することが多く、さまざまな可能性のある経路を探索し、ソリューショングラフの失敗した枝を刈り取る。この戦略は、経験豊富なプログラマーでさえもコード開発中にエラーに遭遇することがよくあるように、人間の問題解決アプローチを反映している。Yaoら（2023）の「思考の木」に関する実験では、24ゲーム、創作、ミニクロスワードにおいて、LLMの推論が大幅に改善されることが示された。彼らの「思考の木」は、可能性のある枝を評価し、それを切るかどうかを決めるという試行錯誤の戦略に似ている。しかし、彼らの実験に用いられた課題が単純であるため、空間解析のようなより複雑な領域での一般化には限界がある。

6.5 オンライン地理空間データの発見とフィルタリング

今回の事例では、データファイルのパスやURL、データアクセスAPIなど、LLM-Geoのためのデータ、およびデータの説明文を提供しました。LLM-Geoは、同様のデータを入力した空間解析タスクを受け入れる準備ができています。しかし、自律型GISは、タスクを完了するために必要なデータを独自に収集する能力が必要である。LLMベースの自律型エージェントの多くは、インターネットからデータを検索して取得するための検索エンジン（APIを使用）を備えている。幸い、ここ数十年で多数の地理空間データセット、標準的な地理空間ウェブサービス（OGC WMS、WFS、WCS、WPSなど）、地理空間データアクセス用のREST APIが確立された。例えば、米国の人口データは、OpenStreetMap（OpenStreetMap 2023）と同様に、APIを介してUS Census Bureau（2023）から抽出することができる。また、Google Earth Engineのデータカタログ、NOAAとNASAのデータカタログ、EarthCube、欧州環境庁（EEA）の地理空間データカタログなど、様々な国や国際機関、政府機関によって大規模なオンライン地理空間データカタログが作成されている。課題は、時空間解像度、時空間カバー率、そして、そのような観点から、適切かつ高品質なデータレイヤーを探し、選択することにある。の精度を高める。自律型GISの開発者は、LLMが与えられた空間分析タスクに対して最も関連性が高く正確な地理空間データセットを発見し、フィルタリングし、利用するための実用的な戦略を確立する必要があります。

6.6 「なぜ」の質問に答える

GPT-4を搭載したLLM-Geoは、「どのように」空間分析を行い、多数の日常的な地理空間タスクを自動化する能力を示している。次の課題は、自律型GISが「なぜ」という問いに対応できるようにすることである。スマートフォンのモビリティデータによると、なぜ今月は顧客数が20%減少したのか」「なぜこの渡り鳥は過去10年間に進路を変えたのか」といった疑問は、基本的な空間分析にとどまらず、仮説の創出、データの選択、実験設計の領域にまで踏み込んでくる。これらの質問に答えるために、自律型GISは、提起された質問、利用可能なデータ、およびコンテキストに基づいて、情報に基づいた仮説を策定することによって研究を設計する能力を持つ必要があります。

6.7 大規模空間モデル (LSM) の構築

膨大なテキストコーパスで訓練されたLLMは、言語スキル、知識、推論能力を身につけていますが、これらのコーパス内の空間サンプルが乏しいため、空間認識能力は依然として限定的です。GIS科学では、豊富な過去のリモートセンシング画像、グローバルベクターデータ、インフラや不動産の詳細な記録、その他の膨大な地理空間ビッグデータソースなどの多くのリソースが、大規模モデルのトレーニングにまだ十分に取り入れられていない。LLMが広範なテキストコーパスで学習されるのと同じように、利用可能なすべての空間データで学習される大規模空間モデル (LSM) の可能性を考えてみましょう。このようなモデルは、地球の表面を詳細に理解し、あらゆる場所を正確に描写し、生態系や地球圏のダイナミクスを理解することができる可能性があります。これは、将来の人工知能の空間認識を豊かにするだけでなく、GISサイエンスの分野にも変革をもたらし、自律型GISが「なぜ」という問いに答える力を与える可能性があります。地球表面と人間社会をより正確に表現できる大規模空間モデルの育成に向けた、さらなる研究と努力を提唱する。

7. 結論

本研究では、GISや空間解析をより身近でユーザーフレンドリーなものにすることを目的とした、AIを活用した次世代の地理情報システムとして、Autonomous GISを提示した。自律型GISの目標は、自然言語によるタスクを受け付け、人間の介入を最小限に抑えながら空間問題を解決することである。LLMを中核とした推論機構を採用することで、自律型GISは5つの自律的な目的（自己生成、自己組織化、自己検証、自己実行、自己成長）を達成すべきと提案する。この概念を説明するために、我々はPython環境下でGPT-4 APIを用いて自律型GISのプロトタイプであるLLM-Geoを開発し、自律型GISがどのように見えるか、どのように自律的に結果を提供できるかを実証した。3つの事例において、LLM-Geoは、集計された数値、グラフ、地図といった期待通りの結果を出すことに成功し、手作業による操作時間を大幅に短縮することができました。まだ発展途上ではあるが、LLM-Geoは自律型GISの概念と実現可能性を示している。Zhuら(2021)の「次世代のGISは簡単でなければならない」というビジョンに共鳴し、次世代のAI搭載GISとしての自律型GISは、この目標の達成に向けて大きな可能性を持っていると考えています。我々は、GIS科学コミュニティが自律型GISの研究開発にさらに力を注ぎ、空間解析をより簡単に、より速く、より多くの人々が利用できるようにすることを奨励する。