

## 1. はじめに

学術論文における科学図は、複雑な概念や構造、関係性を直感的に理解させるための重要な役割を担っている。科学図を自動生成することで、研究者の図作成プロセスの支援が可能である。学術論文内の科学図の多くがベクタ形式で表現されるため、ベクタ形式に基づく科学図の自動的生成が求められている。AutomaTikZ[1] は、大規模言語モデル (LLM) を活用して LaTeX の TikZ パッケージのコード生成を行うことでベクタ形式の科学図生成を実現している。しかし、AutomaTikZ には以下の 2 つの問題がある。  
 (i) 構文エラーなどを含むコードが生成されることがある。  
 (ii) 生成結果を人間が修正するには TikZ パッケージの専門知識が必要となる。

本研究では、専門的な記述言語に対する知識を必要とせず、作図ツール上で人間が直感的に編集可能な形式である XML を対象とする。そして、MLLM を用いた科学図の自動生成手法を提案する。具体的には、エラーの自動修正と生成結果の自己改善を行う機能を導入する。これにより、構造的な整合性を保ちつつ、高品質なベクタ形式の科学図を生成可能となる。また、生成された XML 形式の科学図は既存の作図ツール上で容易に修正・拡張・再利用ができる。そのため、図の作成から改良に至る反復的な作業プロセスの効率化に貢献する。

## 2. AutomaTikZ

Belouadi らは、科学図を対象として TikZ コードの自動生成手法である AutomaTikZ を提案している [1]。AutomaTikZ は、CLIP による画像特徴を事前学習済みの LLaMA に統合したモデルをファインチューニングし、自然言語キャプションと真値となる図の画像から TikZ コードを生成する。これにより、テキストと図の整合性を考慮した高品質な TikZ コードの生成を可能にしている。

生成性能の評価には DaTikZ データセットが用いられている。DaTikZ は、インターネット上から収集された約 12 万件もの TikZ コードと自然言語キャプションのペアから構成される大規模データセットである。データは、TeX Stack Exchange の投稿、arXiv 論文の TeX ソース、および教育目的の TikZ 図共有サイトなど、実用的に利用されている公開リソースから収集されている。AutomaTikZ は、DaTikZ データセットを用いて学習することで TikZ コードの自動生成を実現している。しかし、生成対象が TikZ に限定されている点や、構文エラーを含むコードを生成する場合があるといった課題がある。

## 3. 提案手法: XML-Diagram Agent

ベクタ形式の科学図の記述方法には、TikZ や SVG など多様な形式が存在し、それぞれ異なる目的に基づいて設計されている。AutomaTikZ で対象とされている TikZ は TeX 用の描画パッケージであり高度な数理表現が可能である。しかし、文法が複雑であり、人間による図の追加修正等は TikZ の専門知識が必要となる。一方で、XML はノード・エッジ・レイアウト情報が明確に分離された構造として記述でき、draw.io のような作図ツール上で追加修正可能である。

本研究では、クエリから XML の科学図を生成するフレームワークである XML-Diagram Agent (XDA) を提案する。XDA は、ラスタ画像生成モデルと MLLM を用いてエラーの修正や図の品質の改善を自律的に行うことで、高品質なベクタ形式の科学図を生成する。XDA のフレームワークを図 1 に示す。本フレームワークは以下のモジュールを組み合わせて構築する。

### Query Expansion

Query Expansion は与えられたクエリを基に、図を構成する要素や構造をプロンプト文に変換する。これにより、曖昧さの少ない構造的な情報を後続のモジュールに入力で

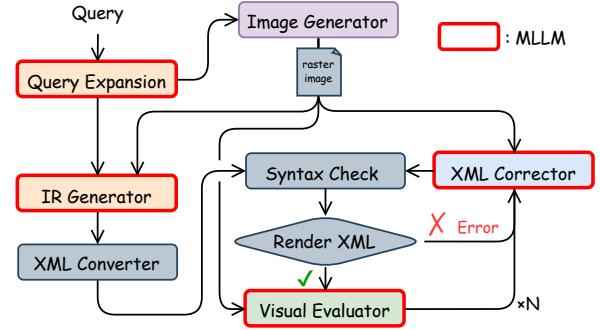


図 1: XDA のフレームワーク

き、図の意味的整合性および生成品質の向上が期待できる。

### Image Generator

Image Generator (IG) では、プロンプト文を画像生成モデルに入力してラスタ形式の画像を生成する。ここで生成した画像は、後続のモジュールに入力してデザイン的な補助情報として活用する。

### IR Generator

IR (Intermediate Representation) Generator では、プロンプト文とラスタ画像から図の構成要素やそれらの関係をグラフ表現として生成する。これにより、ノードやエッジといった要素の種類、接続関係、階層構造などを明示的に表現することができる。また、IR Generator で生成したグラフ表現は、後続の XML Converter で XML のドロフトをルールベースの作成に利用する。

### Visual Evaluator

Visual Evaluator では生成した XML をレンダリングした図が、クエリとラスタ画像に従っているかを視覚的に評価し、不整合や改善点を抽出する。評価結果は自然言語によるフィードバックとして出力され、後続の XML Corrector に提供される。

### XML Corrector

XML Corrector では XML とラスタ画像と Visual Evaluator が output したフィードバック文を入力として受け取り、改善した XML を生成する。生成した XML は外部ツールによって構文チェックおよび再レンダリングを行い、その結果を再び Visual Evaluator に入力する。この処理を反復することで、図の構造的および視覚的な品質を段階的に向上させる。

ここで、ラスタ形式の画像を生成する Image Generator 以外のモジュールに MLLM を利用する。最終的にフレームワークは、以上 4 つのモジュールに加えてルールベースで動作する XML Corrector と外部ツールによる Syntax Check, Render XML を組み合わせて、段階的かつ自律的にエラーを改善し、図の品質を向上させるフィードバックループによって構成される。

## 4. 評価実験

提案手法の有効性を検証するために、科学図の生成性能の比較を行う。

### 4.1. 実験概要

評価実験を行うために、科学図の XML データをインターネット上から収集して DiagramXML データセットを構築した。DiagramXML は、インターネット上から収集した科学図の XML と、説明文から構成される。説明文は、XML をレンダリングして得られた画像に対して GPT-4o を用いて生成した。説明文の正確さと図の完成度を人間による 0 から 100 のスコアリングにより検証し、70 以上のスコアであった 70 件を利用する。

表 1: 各手法の評価結果

生成手法	w/ IG	CLIPScore	C-BLEU	DiagramEval						SR
				Node prec.	recall	F1	Path prec.	recall	F1	
Zero-Shot XML	-	71.38	5.760	0.601	0.480	0.518	0.252	0.175	0.174	0.69
Zero-Shot graph	-	85.61	<b>6.332</b>	0.859	0.708	0.752	0.396	0.290	0.300	<b>0.99</b>
XDA	✓	84.79	6.060	0.857	0.725	0.762	0.440	0.332	0.336	0.97
		<b>87.32</b>	6.123	<b>0.872</b>	<b>0.783</b>	<b>0.802</b>	<b>0.516</b>	<b>0.443</b>	<b>0.426</b>	<b>0.99</b>

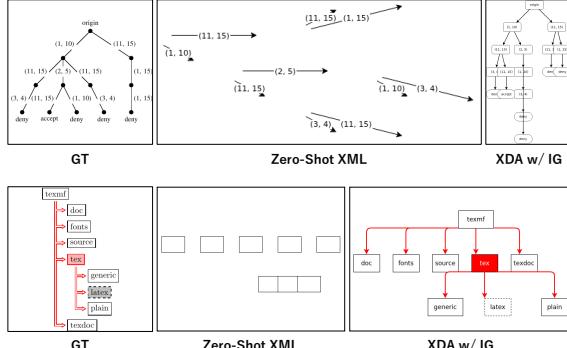


図 2: 各手法により生成された科学図

## 4.2. 定量的評価

定量的評価では、提案手法である XDA と、2 種類の Zero-Shot 手法との比較を行う。具体的には、Zero-Shot prompting により XML を直接生成する手法 (Zero-Shot XML) と、中間表現としてグラフ表現を生成した後に XML に変換する手法 (Zero-Shot graph) を用いる。これらの手法においては、いずれも MLLM として Qwen2.5-VL-72B-Instruct モデルを使用する。また、XDA における IG で利用する画像生成モデルは、gpt-image-1 モデルとする。さらに、XDA における IG の有効性を確認するために IG の有無による比較も行う。評価指標として CLIPScore, C-BLEU, DiagramEval[2], 生成成功率 (SR) を用いる。

各手法の評価結果を表 1 に示す。これより、CLIPScore と DiagramEval の評価値に注目すると、提案手法である IG あり XDA が最高精度を達成していることが確認できる。これより、ラスタ画像生成モデルの性能も活かしつつ自己改善を行うことで高品質なベクタ形式の科学図を生成できていると言える。また、Zero-Shot XML は全ての評価指標で Zero-Shot graph を超える精度を達成しており、科学図生成においてグラフ表現を中間表現として用いることが有効であることがわかった。

さらに、XDA におけるラスタ画像生成モデルの有無による精度に着目すると IG あり XDA は IG なし XDA と比較して全ての評価指標で高い精度を示している。これより、ラスタ画像生成モデルで生成した画像を基にフィードバックと修正を繰り返して生成した科学図は、より高品質な科学図を生成できているといえる。

以上より、提案手法である IG あり XDA は、Zero-Shot prompting 手法とは異なり、自己改善ループによる安定した構造生成を可能とし、成功率および構造的な一致率の観点で優れた性能を示しており、高品質かつ構造的に整合性の取れた XML 形式の科学図の自動生成が可能であることがわかった。

## 4.3. 定性的評価

提案手法である IG あり XDA により生成した場合と Zero-Shot prompting 手法で XML を直接生成した場合を定性的に比較する。各手法により生成された科学図を図 2 に示す。これより、IG あり XDA は Zero-Shot prompting 手法で XML を直接生成した場合と比べて、より正解画像に近い図を生成できていることがわかる。特に矢印の関係性が大きく向上しており、IG あり XDA 手法の有効性を確認した。

さらに、IG あり XDA 手法における Visual Evaluator による生成図の変化を図 3 に示す。これより、Visual Evaluator

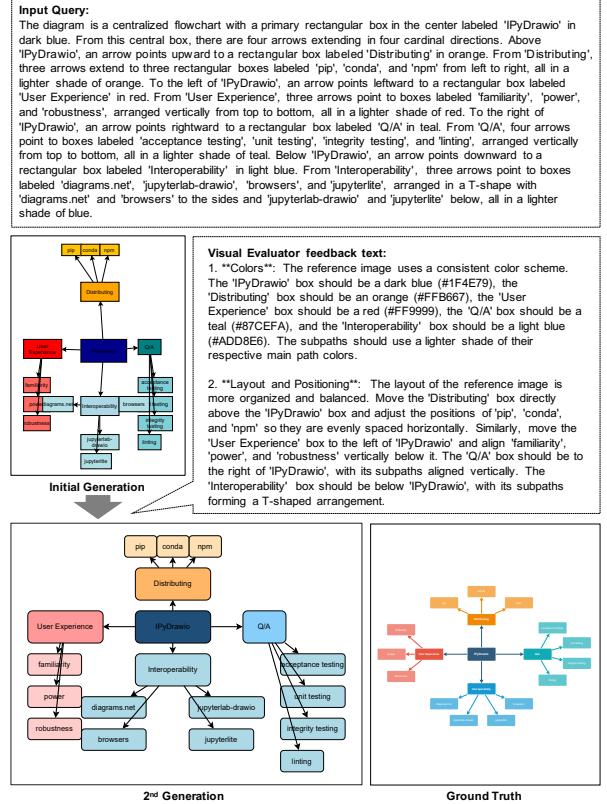


図 3: Visual Evaluator による生成図の変化

で生成されたフィードバック文を用いて修正された科学図は回数を重ねるごとに真値に近づいていることがわかる。特に、図中のオブジェクトの配置や色味などが改善されており、生成結果によるフィードバック文の指摘を踏まえて改善できたといえる。

## 5. おわりに

本研究では、グラフ構造を中間表現として利用し、生成結果に対するフィードバックを活用した自己改善を行うフレームワークである XDA を提案した。実験結果より、提案手法は直接 XML を生成する場合と比較して、生成の安定性および構造的正確性の観点で優れた性能を示すことを確認した。また、特にグラフ表現を中間表現として用いることが極めて有効であることを確認した。今後の課題としては、より多様な科学分野における図表への適用や、レイアウトや視認性といった視覚的品質のさらなる向上が挙げられる。

## 参考文献

- [1] Jonas Belouadi, et al., “AutomaTikZ: Text-Guided Synthesis of Scientific Vector Graphics with TikZ” CILR. 2024.
- [2] Chumeng Liang, et al., “Evaluating LLM-Generated Diagrams as Graphs” EMNLP. 2025.

## 研究業績

- [1] 増田 大河 等, “TikZAgent: LLMs による科学ベクタ図の自動生成”, 画像の認識・理解シンポジウム (MIRU), 2025. (他 3 件)