VLM を用いた科学ベクタ図 XML の自動生成

増田 大河 [1] 田中 翔平 [2] 陳 辰昊 [1] 平川 翼 [1] 山下 隆義 [1] 齋藤 邦章 [2] 藤吉 弘亘 [1] 牛久 祥孝 [2]
[1] 中部大学 [2] オムロンサイニックエックス株式会社

{masudataiga, hirakawa, chinshinkou}@mprg.cs.chubu.ac.jp
{takayoshi, fujiyoshi}@fsc.chubu.ac.jp
{shohei.tanaka, kuniaki.saito, yoshitaka.ushiku}@sinicx.com

概要

学術論文における図は，複雑な概念や構造，関係性を視覚的に示す重要な手段である．これらの図を自動生成することで，研究者の図作成プロセスを支援することが可能である．学術論文中の図の多くはベクタ形式で作成されているが，従来のベクタ画像生成を目的とした研究は，アイコンなどを対象としている．そこで本研究では，ベクタ形式の学術論文における科学図の自動生成を目的とし，専門的な記述言語に依存せず，作図ツール上で直感的に編集可能な XML 形式の図生成手法を提案する．生成結果へのフィードバックを活用した自己改善フレームワークにより，構造的整合性を保ちつつ高品質なベクタ形式の科学図を生成し，図作成から改良までの反復的作業の効率化を実現する．

1 はじめに

学術論文や技術資料で使用される図は，複雑な概念や構造，関係性を視覚的に表現するうえで不可欠な要素である．また，重要な発見を提示するために，ベクタ画像が利用されている．これらの図を自動的に生成することは，研究の効率化に有用である．

ベクタ形式の科学図を生成する先行研究として AutomaTikZ[1] がある．AutomaTikZ 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の生成を実現している．しかし，AutomaTikZ には次の問題点が存在する．(i) 1 回限りの生成に依存しているため，生成したコードに対してエラーの検出や自動修正を行うことができない．(ii) LaTeX の TikZ パッケージのコード生成では，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になり，インタラクションが考慮されていない．

そこで本研究では，専門的な記述言語に依存することなく，人間が作図ツールのユーザインタフェース上で直感的に編集・改善可能な形式で科学図を生成する手法を提案する．具体的には，XML 形式を対象とし，生成結果に対して AI が自らフィードバックを生成し自己改善を行うフレームワークを構築する．これにより，高品質かつ構造的に整合性の取れた XML 形式の科学図を自動生成が可能となる．これより，生成された科学図は既存の作図ツール上で容易に修正・拡張・再利用することができ，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ことが期待できる．

2 関連研究

Stable Diffusion[2] や DALL-E[3]，OpenAI が提供する gpt-image-1 等のモデルは，実際の写真やイラストに匹敵する画像を生成することができ，科学図に関しても高品質な画像を生成できる．しかし，生成できるアスペクト比が固定されている点や，ベクタ形式の画像ではなくラスタ形式の画像を生成している点で科学図の生成には適していない問題が存在する．

Belouadi らは，自然言語による記述からベクタ形式の科学図を自動生成することを目的として，DaTikZ データセットの構築とテキスト入力に基づく TikZ コードの自動生成する AutomaTikZ を提案している [1]．DaTikZ データセットは，ベクタ形式の図を描画する TikZ コードと，その内容を記述した自然言語によるキャプションのペアデータから構成され，約 12 万件ものサンプルからなる大規模データセットである．データは，インターネット上で公開されている実用的な TikZ コードを広く収集することで構築されており，TEX Stack Exchange の投稿，

arXiv 論文の TeX ソースファイル，教育目的の TikZ 図共有サイトから収集されている．AutomaTikZ では，DaTikZ データセットを用いて，大規模言語モデルによる TikZ コードの自動生成手法を複数検討している．はじめに，事前学習済みの LLaMA[4] をファインチューニングし，入力された自然言語キャプションから対応する TikZ コードを生成する手法が検討されている．さらに，視覚情報の活用を目的として，CLIP[5] による画像特徴を LLaMA に統合したモデルを提案している．このモデルでは，キャプションに加えて真値となる図の画像そのものを入力として追加することで，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．

3 提案手法

ベクタ形式の科学図を生成するための記述形式には，TikZ をはじめとして SVG など多様な選択肢が存在する．しかし，それぞれの形式は構造的特徴や文法の複雑さ，編集容易性，LLM との親和性といった観点で大きく性質が異なる．SVG は <path> 要素を中心として記述されるため，同じ線分や矩形であっても複雑なパス列に変換される．したがって，複雑な科学図の構造の意味理解や自動整形が困難である．TikZ は高度な数理表現が可能であるが，文法が複雑であり，LaTeX コンパイル環境への依存が強く，人間による図の追加の修正等は TikZ の専門知識が必要となる．一方で，draw.io の XML ではノード・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る．draw.io は作図のための Web アプリケーションでありインターフェース上で即座に編集できるため，人間による追加の修正・拡張・再利用が容易である．

そこで本研究では，draw.io のインターフェース上で利用可能な XML 形式を対象とする．また，最新の画像生成モデルは科学図もある程度の性能でラスタ画像を生成することができる．そのため，最新の画像生成モデル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するフレームワークを構築する．

XML 生成フレームワークの概要図を図 1 に示す．フレームワークは主に以下の 4 つのモジュールを組み合わせて構築する．

- **Query Expansion**
  与えられたクエリを基に，図を構成する要素や構造を明示化するプロンプト文に変換する．これにより後続のモジュールがより曖昧さの少ない構造的な情報を入力として利用でき，図の意味的整合性および生成品質の向上が期待される．
- **Image Generator**
  画像生成モデルを用いて，Query Expansion により拡張されたプロンプト文を入力としてラスタ形式の画像を生成する．生成した画像を後続の処理で入力に追加することで，デザイン的な補助情報として活用する．
- **IR Generator**
  IR (Intermediate Representation) Generator は Query Expansion により拡張されたプロンプト文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．IR Generator で生成したグラフ表現から XML Converter で XML のドラフトをルールベースで作成する．
- **Visual Evaluator**
  生成された XML をレンダリングして得られた図が，ユーザ指示や意図した構造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**
  現状の XML と Visual Evaluator から出力されたフィードバック文を入力として受け取り，改善された XML を生成する．生成された XML は外部ツールによって構文チェックおよび再

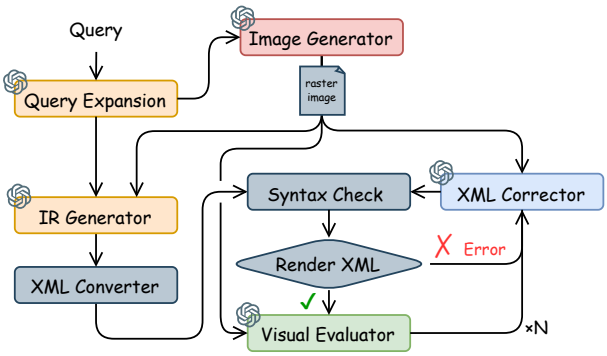

図 1 XML 生成フレームワークの概要図

Algorithm 1 フィードバックループ

Require: クエリ q, 最大反復回数 L, 目標スコア θ

$q' \leftarrow$ Query Expansion(q)
$I_{ref} \leftarrow$ Image Generator(q')
$IR \leftarrow$ IR Generator(q', I_{ref})
$XML \leftarrow$ Compile(IR)
$best \leftarrow (0, \emptyset)$ ▷ best = (score, XML)
for $t = 1$ to L **do**
  $PNG \leftarrow$ Render(XML)
  $(s, fb) \leftarrow$ Visual Evaluator(q', PNG, I_{ref})
  if $s > best.score$ **then**
    $best \leftarrow (s, XML)$
  end if
  if $s \geq \theta$ **then**
    break
  end if
  $XML \leftarrow$ XML Corrector(XML, fb, I_{ref})
end for
return $best.XML$

レンダリングが行われ，その結果は再び Visual Evaluator に入力される．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．

これらのモジュールを組み合わせて，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を構成する．ここで，ラスタ形式の画像を生成する Image Generator 以外の Query Expansion，IR Generator，Visual Evaluator，XML Corrector では Vision-Language Model (VLM) を利用する．

提案手法におけるフィードバックループをアルゴリズム 1 に示す．フィードバックループでは Visual Evaluator により評価された結果のスコアを基に，ループの終了を判断する．これにより，Visual Evaluator の評価に基づいて図表表現を逐次的に修正し，目標スコアに到達するまで改善を繰り返す適応的な生成プロセスを実現する．

4 評価実験

提案したフレームワークの有効性を検証するために，図の生成性能比較を行う．比較対象として Zero-shot prompting 手法で XML を直接生成した場合と，中間表現としてグラフ表現を直接生成しルールベースで XML に変換した場合，フレームワークにおける Image Generator の有無とで比較を行う．既存の科学図データセットとして DaTikZ データセットを利用する．また，生成した XML を検証するために XML データをインターネット上から収集し，DiagramXML データセットを構築した．DiagramXML は，収集した XML をレンダリングした画像を GPT-4o を用いて図の説明文から構成される．図の説明文の正確さと図の完成度を人間により検証し，高い評価であった 70 件を採用する．実験では，VLM に Qwen2.5-VL-72B-Instruct[6] を利用する．また，Image Generator では，gpt-image-1 モデルを利用する．

4.1 定量的評価

フレームワークが生成した XML をレンダリングした図の生成性能を定量的評価により比較する．また，フレームワーク中における Image Generator (IG) の有無による比較も行う．評価指標として CLIPScore[7]，C-BLEU[8]，DiagramEval[9]，生成成功率 (SR) を用いる．CLIPScore は，CLIP モデルを用いて，生成画像と対応する真値画像との整合性を評価する指標である．また，C-BLEU は，テキスト類似度を計算する BLEU[10] を基本にプログラミング言語の構文の冗長性に対応した指標である．DiagramEval は，図を画像としてではなく，テキスト属性付きのグラフとして扱い評価を行う．つまり，図中の各テキスト要素をノード，それらを結ぶ矢印や線分を有向エッジと見なすことで，図全体を有向グラフとして構造化し，このグラフ同士を比較することで評価を行う．これにより，学術図にとって本質的な情報のみを分離して評価することができ，人間の評価との相関が高い評価を可能にしている．

Qwen2.5-VL-72B-Instruct を利用したときの各手法の評価結果を表 1 に示す．これより，DaTikZ において，Zero-Shot graph は高い Node / Path 精度および SR 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極めて有効であることが確認できる．また，提案手法である framework w/ IG は，Zero-Shot graph に匹敵する高い性能を安定して達成しており，特に DiagramXML においては C-BLEU を除くすべての評価指標で最良の結果を示している．これは，自己改善ループによる反復的な修正が，ノード構造や関係性の整合性を高める上で有効に機能していることを示唆している．

表 1　Qwen2.5-VL-72B-Instruct を利用したときの各手法の評価結果

対象データ	生成手法	CLIPScore	C-BLEU	DiagramEval						SR
				Node			Path			
				prec.	recall	F1	prec.	recall	F1	
DaTikZ	Zero-Shot XML	62.80	-	0.320	0.215	0.241	0.094	0.062	0.067	0.49
	Zero-Shot graph	**79.78**	-	**0.757**	**0.597**	**0.637**	**0.301**	**0.184**	**0.192**	**1.00**
	framework w/o IG	77.02	-	0.594	0.511	0.520	0.279	0.138	0.152	0.96
	framework w/ IG	77.93	-	0.650	0.535	0.565	0.285	0.166	0.179	0.97
DiagramXML	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	**6.332**	0.859	0.708	0.752	0.396	0.290	0.300	**0.99**
	framework w/o IG	84.79	6.060	0.857	0.725	0.762	0.440	0.332	0.336	0.97
	framework w/ IG	**87.32**	6.123	**0.872**	**0.783**	**0.802**	**0.516**	**0.443**	**0.426**	**0.99**

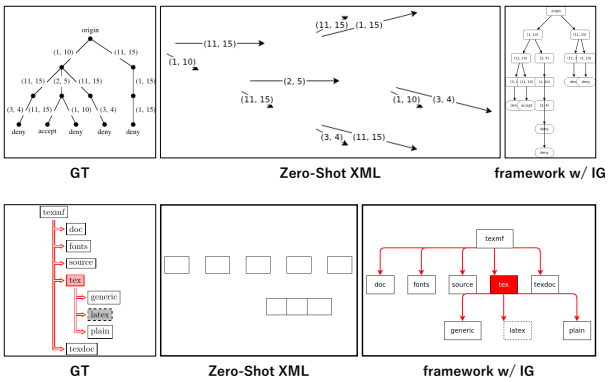

図 2　各手法により生成された科学図

4.2　生成画像の定性的評価

framework 手法により生成した科学図と直接 XML を生成した場合の科学図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，グラフ表現と gpt-image-1 で生成した画像を中間表現として用いつつ XML を生成する framework w/ IG は直接 XML を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．

さらに，Visual Evaluator による生成図の変化を図 3 に示す．これより，Visual Evaluator で生成されたフィードバック文を用いて修正された図は回数を重ねるごとに正解図に近づいていることが確認できる．しかし，フィードバック文で改善案が明記されているにも関わらず図の改善は僅かであることがわかる．これより，XML の改善を行う XML Corrector は現在の XML とフィードバック文を入力として受け取り改善された XML 生成するが，現在の XML の影響が大きく，大きな改善を行うことが難しいと考えられる．

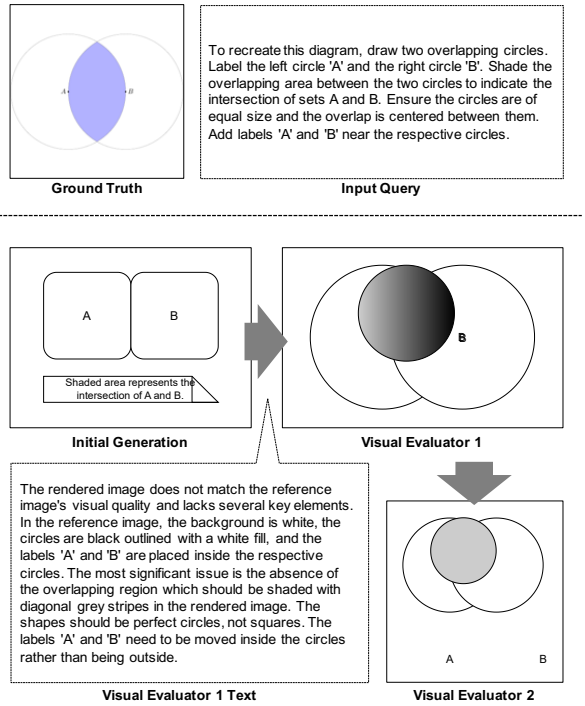

図 3　Visual Evaluator による生成図の変化：正解図（上階左）と入力クエリ（上階右）および初期生成図（下階左上）と Visual Evaluator による生成図の変化，Visual Evaluator によって生成されたフィードバック文（下階左下）

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フレームワークによる XML 形式の科学図生成手法を提案した．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．特に，グラフ表現を中間表現として用いることの有効性を確認した．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を目指す．

謝辞

本研究は，JST【ムーンショット型研究開発事業】【JPMJMS2236】の支援を受けたものです．

参考文献

[1] Jonas Belouadi, Anne Lauscher, and Steffen Eger. Automatikz: Text-guided synthesis of scientific vector graphics with tikz. In **The Twelfth International Conference on Learning Representations**, 2024.

[2] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In **Proceedings of the IEEE/CVF conference on computer vision and pattern recognition**, pp. 10684–10695, 2022.

[3] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. **arXiv preprint arXiv:2204.06125**, Vol. 1, No. 2, p. 3, 2022.

[4] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. Llama: Open and efficient foundation language models, 2023.

[5] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In **International conference on machine learning**, pp. 8748–8763. PmLR, 2021.

[6] Binyuan Hui, Jian Yang, Zeyu Cui, Jiaxi Yang, Dayiheng Liu, Lei Zhang, Tianyu Liu, Jiajun Zhang, Bowen Yu, Keming Lu, et al. Qwen2. 5-coder technical report. **arXiv preprint arXiv:2409.12186**, 2024.

[7] Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. Clipscore: A reference-free evaluation metric for image captioning. In **Proceedings of the 2021 conference on empirical methods in natural language processing**, pp. 7514–7528, 2021.

[8] Aryaz Eghbali and Michael Pradel. Crystalbleu: precisely and efficiently measuring the similarity of code. In **Proceedings of the 37th IEEE/ACM International Conference on Automated Software Engineering**, pp. 1–12, 2022.

[9] Chumeng Liang and Jiaxuan You. Evaluating LLM-generated diagrams as graphs. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng, editors, **Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing**, pp. 12689–12701, Suzhou, China, November 2025. Association for Computational Linguistics.

[10] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In **Proceedings of the 40th annual meeting of the Association for Computational Linguistics**, pp. 311–318, 2002.

A　参考情報

A.1　DiagramXML データセット

本研究では，既存の科学図データセットに加えて，XML 形式における科学図を評価するため，新たに DiagramXML データセットを構築する．具体的には以下の 4 段階の処理によって構築する．

1. 科学図データの収集

はじめに，インターネット上に公開されている科学図データを収集する．draw. io などで展開可能な XML 形式をそのまま公開している場合はほとんどなく，レンダリング後の PNG 形式や JPEG 形式，SVG 形式，PDF 形式などの形式で公開されている事が多い．そのため，XML への逆変換の可能性高い SVG 形式のデータを収集した．

2.XML への逆変換

収集した SVG 形式のデータに対して，draw.io で展開し，XML 形式へ逆変換を行う．この時逆変換に失敗したサンプルはこの段階で除外した．

3. 科学図の説明文の生成

次に，各科学図に対して，VLM を用いて図の説明文を生成する．説明文は，図の内容を自然言語で要約したものであり，図中の主要な構造や関係性を記述することを目的とした．この時，XML 形式のデータを PNG 形式にレンダリングして VLM に入力する．また，VLM は gpt-4o を用いた．

4. 人手による品質評価

VLM を用いて生成させた説明文は，科学図を正確に説明してない可能性がある．また，インターネット上から収集した科学図データには，チュートリアルのような図が含まれており，評価用のデータとして科学図の品質が担保されていない．そのため，生成された科学図の説明文および対応する科学図画像について，生成した科学図の説明文が正確であるか，科学図の複雑度が適切であるかを人手による評価を行う．

これらの観点から総合的に評価し，高い評価を得た 70 件のデータのみを最終的な DiagramXML データセットとして採用した．DiagramXML データセットの例を図 4 に示す．

A.2　他のモデルでの定量的評価

gpt-4o を利用したときの各手法の評価結果を図 5 に示す．これより，gpt-4o を利用したときも

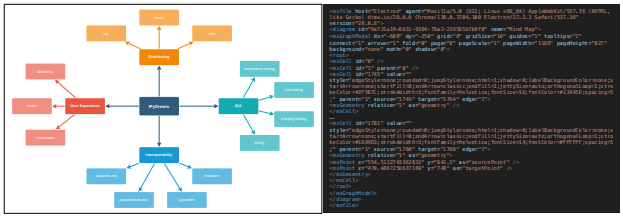

図 4　DiagramXML データセットの例：収録されている科学図（左上）および対応する XML（右上）と画像の説明文（下）

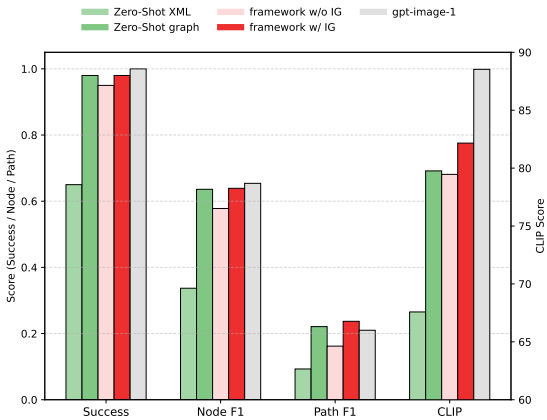

図 5　gpt-4o を利用したときの各手法の評価結果

Zero-Shot graph，framework 手法は直接 XML を生成する手法に比べてより高品質な科学図を生成できていることが確認できる．さらに，framework w/ IG は，CLIPScore で gpt-image-1 を超える精度を示しており，gpt-image-1 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ことが確認できる．Zero-shot graph が高い性能を示した点から，グラフ表現の有効性が高いといえる．一方で，提案手法である framework w/ IG は，単発生成に依存する手法とは異なり，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一貫性の観点で優れた性能を示している．なお，本研究で用いた DiagramEval は，生成画像からグラフ構造を再構成して評価を行う指標であるため，グラフ表現を用いる手法が相対的に有利に評価される可能性がある．しかし，この評価特性は，科学図を構造的情報表現として捉える本研究の立場と整合している．