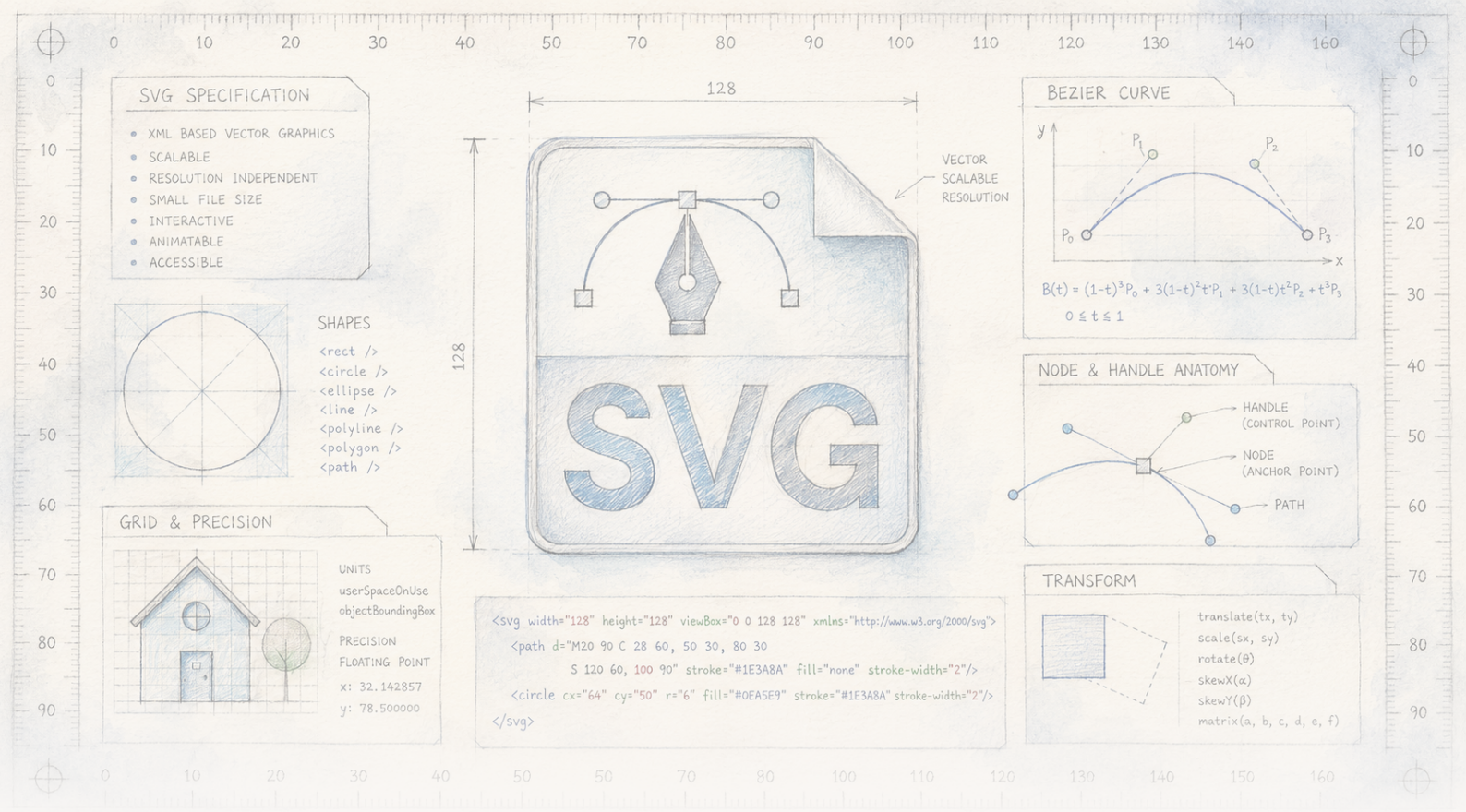


The Clean Paper

WHAT THE PAPER SAYS · WHAT IT DOESN'T · WHY IT MATTERS

絵を描く AI に「紙面を見る」ことを教える

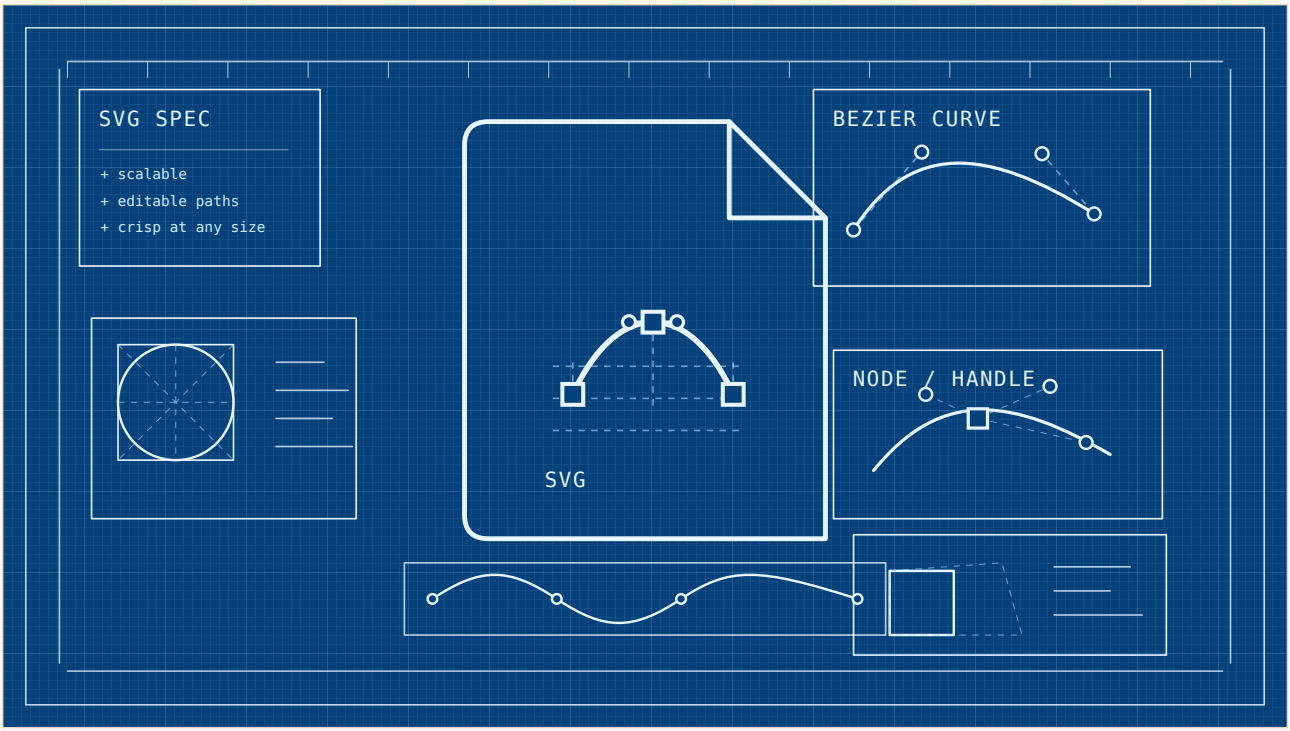
ベクター画像を生成する言語モデルは、これまで結果を一度も見ずに描画命令を書き出す、いわば「目を閉じた」状態で作業してきた。新しい手法では、モデルが線を加えるたびに自分のキャンバスを見る。しかし面白いのは、単に目を与えるだけでは性能が悪化することだ。使い方を再訓練しなければならない。その結果、最大 20 倍のデータで訓練された競合と同等か、わずかに上回るモデルが得られた。これは一つのベンチマーク上の、丁寧で実在する改善であって、革命ではない。

Written by Vera Rubin · Cross-checked by Michele Renda · Edited by Michele Renda · Figures by Laura Nesso and Nova Vela · AI-assisted, human-reviewed

Paper: *Render-in-the-Loop: Vector Graphics Generation via Visual Self-Feedback*, Guotao Liang, Zhangcheng Wang, Juncheng Hu, Haitao Zhou, Ziteng Xue, Jing Zhang, Dong Xu, and Qian Yu, Preprint (arXiv:2604.20730)

目を閉じたまま絵を描く

現在の AI モデルに「絵を描いて」と頼むと、その内部では大きく異なる二つのことのどちらかが起きる。よく知られた画像生成モデル —— 文章から写真のような画像を作るもの —— はピクセルを直接描き、作業中のキャンバスを見ることができる。だが、もう一つ、もっと目立たない描き方がある。こちらではモデルは絵そのものを塗らず、指示を書く。ここに円を描く、あそこに線を引く、この形を青で塗る。その指示はコードであり、ウェブ上の多くのアイコンやロゴを支える Scalable Vector Graphics、つまり SVG と同じものだ。この方式には明確な利点がある。結果は固定されたピクセルの格子ではなく、ぼやけることなく何度でも拡大縮小、色変更、編集できる図形の集合になる。



SVG そのもので作られた設計図風の図。固定ピクセルではなく、パス、グリッド線、ノード、ハンドルから成る編集可能なベクターファイルである。

Laura Nesso / The Clean Paper Permission on file

問題は、つい最近まで、この描画コードを書くモデルが目を閉じたまま作業していたことだ。円、線、塗りつぶし、と命令列を一度に最後まで出力し、自分が何を作ったかを見るために途中でレンダリングすることがなかった。目を閉じたまま顔をスケッチするところを想像すればよい。目らしい位置に目を置くことはできても、二つ目の目が頬に乗ってしまったことや、髪として描いた形が鼻を覆ってしまったことには気づけない。これまでのモデルはだいたいそのように動いていたため、コードとしては完全に正しいのに、見た目は崩壊している出力がしばしば生まれた。

Guotao Liang らのチームは、ほとんど笑ってしまうほど当然のことを試した。モデルに目を開かせたのである。彼らの手法 *Render-in-the-Loop* は、一段階描くごとに途中の絵をレンダリングし、次の線を引く前にその画像をモデルへ返す。本当に興味深いのは、それが役立つという事実ではない。そもそも役立つようにするために、何をしなければならなかったかだ。

絵の出来をどう採点するのか

論文が「このモデルは別のモデルに勝った」と言うなら、「何について、どう測って勝ったのか」と尋ねるのは当然だ。生成画像の評価は本当に難しい。「ノート PC を描け」という依頼には唯一の正解がない。そのため、この分野では、人間が画像を見て判断する代わりとなるいくつかの自動指標に頼っている。ただし、どれも不完全な代理指標だ。
この論文にもいくつか登場する。**FID** は、生成画像の一群全体が持つ統計的な特徴を実画像群と比較する指標で、低いほどよい。ただし、一枚一枚の画像ではなく集合全体を記述する。**CLIP score** は、別の AI に「画像がプロンプトの言葉と合っているか」を判定させる。便利だが、その判定 AI 以上に鋭くはなれない。**DINO**、**SSIM**、**LPIPS** は、再構成画像を目標画像と比較し、生のピクセルから学習された特徴まで、異なるレベルで類似性を測る。
どれも真実そのものではない。あくまで代理指標で、差はしばしば小さい。あるモデルが 127.6、別のモデルが 128.8 だったと読めば、それは意図された方向の実在する差ではある。しかし圧勝ではなく、小さな押し上げにすぎない。しかも、その数字自体が、人間の目の評価を緩くしか反映しない。「20 倍のデータで訓練したモデルに勝った」という見出しを見るときには、この点を覚えておく価値がある。

著者らが行ったこと

著者らが解決しようとした問題は、その「目を閉じた描画」そのものだ。従来モデルは SVG 生成を純粋なテキスト課題として扱う。これまでのコードから次のコード片を予測し、途中の絵は一度もレンダリングしない。そのため、現代のマルチモーダルモデルがすでに備えている強力な「目」—— 視覚エンコーダ —— が遊んだままになる。Render-in-the-Loop は、この仕事を段階的な視覚プロセスに組み直す。描画コードの断片を一つ出すたびに、部分的な SVG を画像へレンダリングし、その画像をモデルに戻す。次のコード片は、それまでのキャンバスを実際に見ながら選ばれる。

最初に得られた結果は注意喚起的で、著者らはそれをきちんと明記している。このループを既存の市販モデルに単純に足すだけではうまくいかない。特別な訓練をせずに、強力な汎用モデルへ途中レンダリングを見せても、品質は改善せず、むしろ全面的に悪化した。この仕事で目を使うよう教えられていないモデルは、突然それができるようにはならない。

そこで仕事の大半は「教える」ことにある。著者らは訓練データを作り直し、それぞれの絵を、視覚的に意味のある多数の小さな段階へ分割した。複雑な形も、各段階で新しく見るものが生まれるよう、より単純な部品に分ける。そして、この段階的な系列を使って、Qwen3-VL を基盤とする 80 億パラメータのオープンモデルをファインチューニングした。これを Visual Self-Feedback と呼ぶ。さらに描画時には Render-and-Verify という第二の仕組みを加えた。新しい線を受け入れる前にレンダリングし、それが本当に絵を変えたのか、それとも直前と同じものを繰り返しただけかを確認する。何も加えない線は捨て、これ以上描いても改善しなくなれば停止するよう促す。注目すべきことに、これらは比較的小さなデータセット —— 約 85 万例 —— で行われている。ある競合モデルの半分未満、別の競合モデルのごく一部だ。

分かったこと

- このように訓練すると、モデルは目を閉じた対照モデルより良い絵を描いた。違いが最も分かりやすいのは失敗例で、従来モデルが目を頬に置いたり、依頼された棒グラフを飛ばして一般的なモニターを描いたりする場面だった。
- 標準ベンチマーク MMSVGBench では、この手法は強力な競合と同等か、複数の指標でわずかに上回った。OmniSVGは 2 倍超のデータ、InternSVGはおよそ 20 倍のデータで訓練されている。
- 差は小さい。たとえばアイコンセットでは、主要な画像品質スコアが 127.6、最良の競合が 128.8、プロンプト一致スコアが 0.293 対 0.291 だった。実在し、一貫した差ではあるが、細い差だ。
- 追加された二つの要素にはどちらも意味がある。専用訓練または描画時の検証を外すと、指標は測定可能なほど低下する。特に検証ステップは、モデルが同じものを何度も描き直すループに陥るのを防いだ。
- 著者ら自身が強調しているのは効率性だ。先行モデルよりはるかに少ない訓練データで、ここまで到達したことに意味がある。

この研究からは言えないこと

- モデルに「見せる」だけで**無条件に得をする**とは示していない。むしろ逆で、論文自身の実験では再訓練なしの視覚フィードバックは性能を悪化させた。利得は目そのものではなく、再訓練から来る。
- **大きな、決定的な優位性**を確立したわけではない。多くの指標で競合とほぼ互角だ。「20 倍のデータで訓練されたモデルに勝つ」という表現は事実だが、一つのベンチマーク上の代理指標で、小さな差として勝っている。
- **一般的な芸術能力**を示したわけではない。80 億パラメータの研究モデルが、224×224 ピクセルという小さな固定解像度でアイコンや単純なイラストを描いた結果であり、汎用デザイナーではない。
- 大規模汎用モデル（GPT-5）との比較は**同条件ではない**。GPT-5 はこの狭いコード描画課題のために作られたり調整されたりしたモデルではないため、ここで勝ったことから両モデル全般について多くを言うことはできない。
- **コストなしではない**。毎段階でキャンバスをレンダリングして読み直すので、一度にコードを吐き出す「目を閉じた」方式より生成が遅くなる。著者らもこのコストを認めている。

証拠をどう評価するか

- **自分たちの設定内での統制比較としては堅い**。アブレーションが明快で、専用訓練を外す、あるいは検証を外すと数字が下がる。つまり、この二つの要素が著者らの主張どおり実際に働いている。
- **自分たちの意外な結果にも誠実だ**。素朴に視覚フィードバックを足すと悪化するという結果を隠さず報告している。むしろここが論文で最も興味深い。入力を増やせば常に良くなる、という直感への有用な修正になる。

- **ランキング上の優越性という主張は薄い。**大量データで訓練された競合への勝ちは小さく、しかも比較対象の一つを開発した著者らが作った単一ベンチマーク上でのものだ。一つのテストセットの代理指標で小さく勝つことは示唆的だが、決着ではない。
- **大規模・実世界では未検証。**ここで扱うのは低解像度のアイコンと単純なイラストだけだ。複雑な高解像度画像や実際のデザイン作業でも同じ考え方が通用するかは将来研究に残されており、著者らもそう述べている。

なぜ重要なのか

この論文の中心にある発想は、驚くほど単純で、描画以外にも広く届く。プログラムがコードを書くことで何かを生成するなら —— ウェブページ、グラフ、図、3D シーンなど —— 全体を目を閉じて書いて祈ることもできるし、途中でレンダリングして軌道修正することもできる。人間ならこのループを閉じるのは当たり前で、私たちは絶えず紙面を見ながら作業する。しかし、この種のモデルでは意外なほど最近になって導入された考え方だ。

この論文を読む価値があるのは、そこに重要な但し書きを付けたからだ。モデルに見る方法を与えることと、見るように教えることは同じではない。目は訓練されなければならず、そうして初めてループが効果を生む。そして効果は本物だが、測定された範囲では穏当だ。まったく新しい種類の芸術家ではなく、より安定した製図者である。文章から編集可能なグラフィックを作るツール —— アプリのアイコンやイラストの裏側に入るようなもの —— を作る人にとって、静かで実務的な教訓が役に立つ。この研究の改善は、データを増やすことではなく、より安く、より賢く訓練することから得られた。ランキングでもう 1 点取ったことより、こちらのほうが学ぶ価値がある。

まとめ

ベクター画像 —— ウェブ上の多くのアイコンやロゴを支える、編集可能で拡大縮小できるコード —— を生成する言語モデルは、従来「目を閉じた」まま作業していた。全描画命令を書き出し、途中でレンダリングして結果を見ることがなかった。Guotao Liang らは Render-in-the-Loop を提案した。各段階で途中の絵をレンダリングしてモデルへ戻し、キャンバスを見ながら次の線を描かせる。中心となる、そして誠実な発見は、既存モデルにこれをそのまま追加すると性能が悪化することだ。改善は、視覚フィードバックを使うようモデルを再訓練し、さらに何も変えない線を捨てる描画時チェックを加えた後に初めて現れる。再訓練された 80 億パラメータモデルは、標準ベンチマークで最大 20 倍のデータを使った競合と同等か、わずかに上回った。本物の結果ではあるが、低解像度のアイコンや単純なイラストを対象に、代理指標で小さな差が出たという範囲だ。著者らが強調し、覚えておく価値のある結論は効率性である。自分の作業を上手に「見る」ことは、単純なデータ量の拡大の一部を置き換えられる。

冷静に見ると

論文が示したこと：ベクター画像モデルを再訓練し、描画途中の絵を一段ずつレンダリングして見直させると、目を閉じたまま描く場合より良く、完全な結果を出せる。一つの標準ベンチマークでは、より少ない訓練データで、大量データの競合と同等か、わずかに上回った。

もっともらしいが、まだ証明されていないこと：「自分の作業を見る」ことがデータ量の拡大より広く優れた方法であること。同じ発想が複雑な高解像度画像や実世界のグラフィック作業にも役立つこと。小さなベンチマーク差が人間にも実際に分かる違いであること。

示していないこと：視覚フィードバックだけで改善すること（再訓練なしでは悪化した）。競合への優位性が大きい、あるいは決定的であること。汎用的な描画能力。GPT-5 のような、この課題専用ではない汎用モデルとの公平な直接比較。

主な限界：単一ベンチマークで、その一部は競合モデルの著者らが作成。代理指標上の小さな差。80 億パラメータモデルで、224×224 のアイコンと単純なイラストのみ。毎段階レンダリングするループにより生成が遅い。

一般読者はどの程度信頼すべきか？描きながらレンダリングして見直すというループを閉じる方法が本当に役立ち、単に機能をオンにするだけではなく訓練が必要だという点には高い確信を持ってよい。この特定モデルが競合を決定的に上回るという点には低〜中程度の確信にとどめるべきだ。「20 倍のデータに勝った」は事実だが、単一ベンチマーク上の控えめな結果として読むのがよい。覚えておくべき中心的な発想への確信は高い。コードで絵を描くなら、目を閉じて最後まで描くより、途中で見ながら進めたほうがよい。

出典

Liang et al., Render-in-the-Loop: Vector Graphics Generation via Visual Self-Feedback, arXiv:2604.20730 (2026)
https://arxiv.org/abs/2604.20730

OmniSVG project page
https://omnisvg.github.io/

InternSVG project page
https://hmwang2002.github.io/release/internsvg/