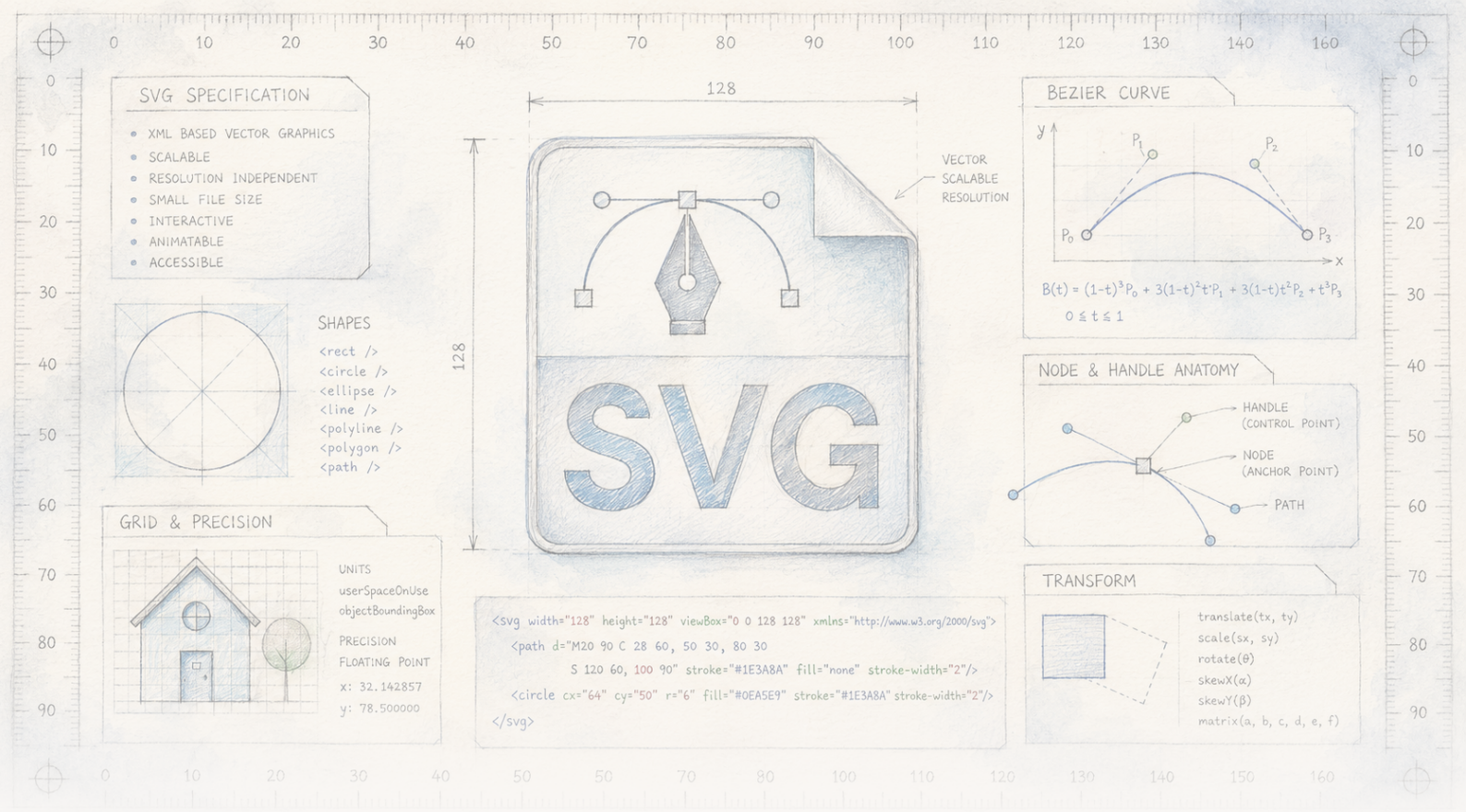


The Clean Paper

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

그림 그리는 AI 에게 페이지를 보는 법 가르치기

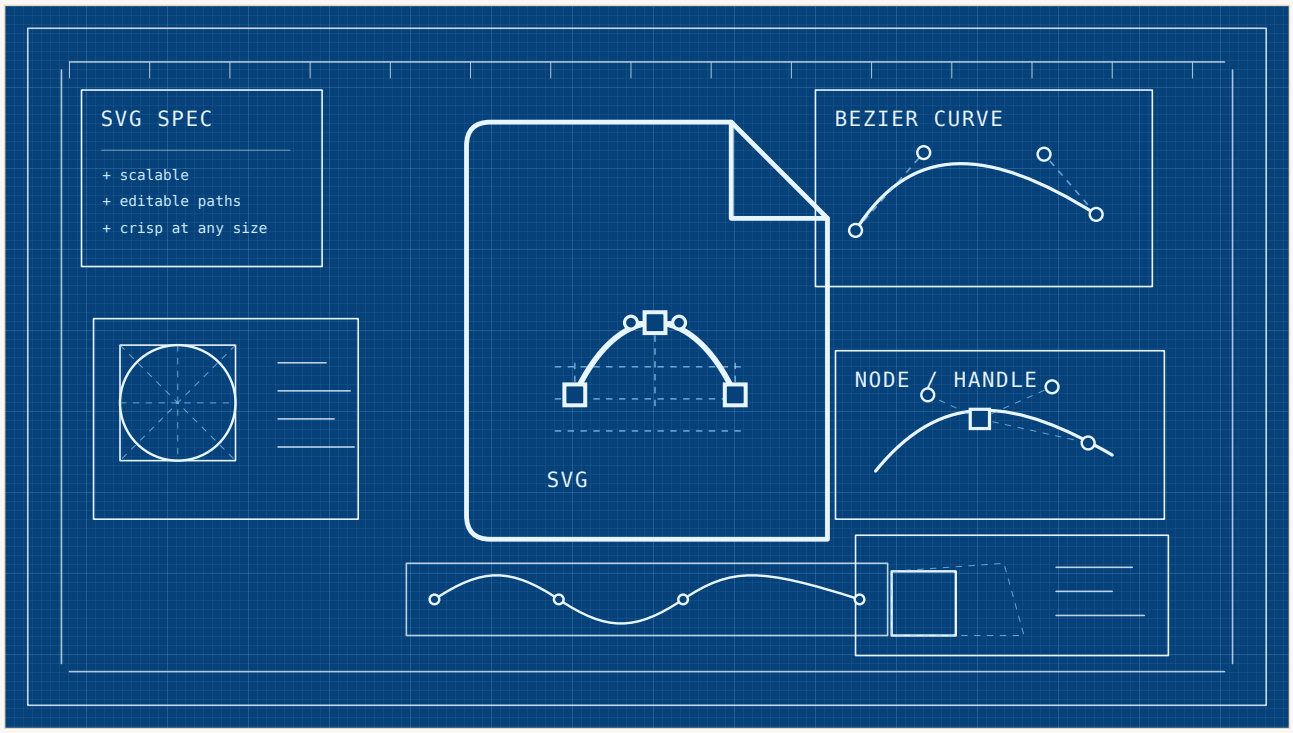
벡터 그래픽을 생성하는 언어 모델은 결과를 보지 못한 채 그림 명령을 작성해 왔다. 새 방법은 획을 하나씩 추가할 때마다 자기 캔버스를 보게 한다. 그런데 솔직하고 중요한 반전은 단순히 눈을 달아주면 오히려 성능이 나빠진다는 것이다. 그 눈을 사용하는 법을 다시 학습해야 한다. 그렇게 한 모델은 최대 *20* 배 많은 데이터로 학습한 경쟁 모델과 비슷하거나 조금 앞섰다. 하나의 벤치마크에서 나온 신중하고 실제적인 결과이지 혁명은 아니다.

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)

눈을 감고 그림 그리기

오늘날의 AI 모델에게 그림을 그려 달라고 하면 내부에서는 서로 아주 다른 두 가지 일 중 하나가 벌어진다. 익숙한 이미지 생성기, 즉 문장 하나로 사진 같은 이미지를 만들어내는 모델은 픽셀을 직접 그리고 작업 중인 캔버스를 볼 수 있다. 하지만 더 조용한 두 번째 종류의 그림 생성이 있다. 모델이 직접 칠하지 않고 여기에 원을 그리고, 저기에 선을 긋고, 이 모양을 파란색으로 채워라 같은 지시문을 작성하는 방식이다. 그 지시문은 코드다. 웹의 수많은 아이콘과 로고 뒤에 있는 SVG(Scalable Vector Graphics) 와 같은 코드다. 장점은 분명하다. 결과가 고정된 픽셀 격자가 아니라 모양들의 집합이므로 흐려지지 않은 채 크기를 바꾸고, 색을 바꾸고, 계속 편집할 수 있다.



원본 SVG 청사진 아트워크. 이 이미지는 고정 픽셀이 아니라 경로, 격자선, 노드와 핸들로 만들어진 편집 가능한 벡터 파일 그 자체다.

Laura Nesso / The Clean Paper Permission on file

문제는 최근까지 이런 그림 코드를 쓰는 모델이 눈을 감고 작업했다는 것이다. 원, 선, 채우기 같은 전체 명령열을 한 번에 내보내고 실제로 렌더링해 자기가 무엇을 만들었는지 보지 않았다. 눈을 감고 얼굴을 스케치한다고 생각해 보자. 눈 하나는 대충 알맞은 곳에 놓을 수 있겠지만 두 번째 눈이 볼에 붙었는지, 머리카락 모양이 코를 덮어버렸는지 알아차릴 방법이 없다. 이런 모델들도 대체로 그랬다. 그래서 코드는 완벽히 유효하면서도 시각적으로는 엉망인 결과가 자주 나왔다.

Guotao Liang 이 이끄는 연구팀은 이제 거의 우스울 정도로 당연해 보이는 일을 했다. 모델이 눈을 뜨게 한 것이다. 이들의 *Render-in-the-Loop* 방법은 각 단계가 끝날 때마다 미완성 그림을 렌더링해, 다음 획을 그리기 전에 그 이미지를 모델에 다시 보여준다. 정말 흥미로운 부분은 이것이 도움이 된다는 사실 자체가 아니다. **도움이 되게 만들기 위해 무엇을 해야 했는가**다.

그림의 점수는 어떻게 매길까?

논문이 자기 모델이 다른 모델을' 이겼다' 고 말하면, 무엇을 어떤 방식으로 측정해 이겼는지 묻는 것이 타당하다. 생성 그림의 품질을 평가하는 일은 실제로 어렵다. ' 노트북을 그려라' 에는 하나의 정답 그림이 없기 때문이다. 그래서 이 분야는 사람이 직접 결과를 보는 일을 불완전하게 대신하는 몇 가지 자동 지표에 의존한다.

이 논문에도 몇 가지가 나온다. **FID** 는 생성 이미지 묶음 전체의 통계적 특성을 실제 이미지와 비교하며 낮을수록 좋다. 다만 개별 그림이 아니라 묶음의 특성을 말한다. **CLIP 점수**는 별도의 AI 에게 이미지가 프롬프트의 단어와 얼마나 잘 맞는지를 묻는다. 유용하지만 그 심사 모델이 보는 만큼만 정확하다. **DINO**, **SSIM**, **LPIPS** 는 원시 픽셀부터 학습된 특징까지 서로 다른 수준에서 재구성 이미지와 목표 이미지를 비교한다.

이 지표 중 어느 것도' 진실' 은 아니다. 모두 대리 지표이며 대개 작은 폭으로 움직인다. 한 모델이 127.6 이고 다른 모델이 128.8 이라면 의도된 방향의 실제 차이일 수 있지만 압도적인 차이가 아니라 작은 움직임이다. 게다가 그 숫자가 사람 눈의 판단을 느슨하게만 추적한다는 점도 기억해야 한다. 특히 제목이'20 배 많은 데이터로 학습한 모델을 이겼다' 일 때 더욱 그렇다.

연구진이 한 일

저자들이 고치려 한 핵심 문제는 바로 '눈을 감고 그리는 것' 이었다. 기존 모델은 SVG 작성을 순수한 텍스트 작업으로 취급한다. 지금까지 나온 코드에서 다음 코드 조각을 예측하고, 렌더링은 하지 않는다. 그러면 현대 멀티모달 모델이 이미 갖고 있는 강력한 '눈', 즉 비전 인코더가 놀고 있게 된다. Render-in-the-Loop 는 작업을 단계별 시각 과정으로 다시 구성한다. 그림 코드 조각 하나를 만든 뒤 부분 SVG 를 이미지로 렌더링해 모델에 다시 넣고, 모델은 지금까지의 캔버스를 실제로 보면서 다음 조각을 고른다.

첫 번째 결과는 경고에 가깝고, 저자들은 이를 솔직하게 보고한다. 기존 상용·범용 모델에 이 루프를 그냥 붙이는 것만으로는 잘되지 않는다. 특별한 학습 없이 강한 범용 모델에 중간 렌더링을 보여주자 품질이 좋아지지 않았고, 오히려 전반적으로 나빠졌다. 이런 방식으로 눈을 쓰도록 배운 적이 없는 모델이 갑자기 잘 보게 되는 것은 아니다.

그래서 작업의 대부분은' 보는 법을 가르치는 것' 에 있다. 연구진은 각 그림을 작고 시각적으로 의미 있는 여러 단계로 나누도록 학습 데이터를 다시 만들었다. 복잡한 도형을 단순한 조각으로 나눠 각 단계마다 새로 볼 것이 생기게 한 뒤, Qwen3-VL 기반의 80 억 파라미터 오픈 모델을 이 단계별 시퀀스로 미세조정했다. 이를 Visual Self-Feedback 이라고 부른다. 생성 시점에는 두 번째 장치인 Render-and-Verify 도 추가했다. 새 획을 받아들이기 전에 렌더링해 그림이 실제로 바뀌었는지, 아니면 직전 획을 반복했을 뿐인지 확인한다. 아무것도 추가하지 않는 획은 버리고, 더 이상 도움이 되는 변화가 없으면 그만 그리도록 유도한다. 주목할 점은 이 모든 것이 비교적 작은 데이터셋, 약 85 만 개 예시에서 이뤄졌다는 것이다. 한 경쟁 모델의 절반보다 적고, 다른 경쟁 모델과 비교하면 작은 일부에 불과하다.

무엇을 발견했나

- 이렇게 학습한 모델은 눈을 감고 그리는 버전보다 더 잘 그렸다. 차이는 특히 실패 사례에서 눈에 띈다. 블라인드 모델이 눈을 볼에 그리거나 요청받은 막대그래프를 빼먹고 평범한 모니터를 내놓는 식의 오류가 줄었다.
- 표준 벤치마크인 MMSVGBench 에서 이 방법은 강한 경쟁 모델들과 비슷하거나 여러 지표에서 약간 앞섰다. 데이터가 두 배 넘게 사용된 OmniSVG와 약 20 배 사용된 InternSVG도 포함된다.
- 격차는 작다. 예를 들어 아이콘 세트에서 주요 이미지 품질 점수는 127.6 으로 최고 경쟁 모델의 128.8 과 비슷했고, 프롬프트 일치 점수는 0.293 대 0.291 이었다. 방향은 일관되지만 차이는 가늘다.
- 추가된 두 요소 모두 실제 기여를 한다. 특수 학습을 끄거나 생성 시 검증을 제거하면 수치가 측정 가능하게 떨어진다. 특히 검증 단계는 모델이 같은 것을 반복해서 다시 그리는 데 빠지는 것을 막는다.
- 저자들이 가장 강조하는 결과는 효율성이다. 선두 모델보다 훨씬 적은 학습 데이터로 이 수준에 도달했다는 점이다.

이 연구가 증명하지 않는 것

- 모델에게’ 보게 하는 것’ 이 공짜 이득이라는 것을 보여주지 **않는다**. 오히려 반대다. 논문 자체 실험에서 재학습 없는 시각 피드백은 성능을 악화시켰다. 이득은 눈 자체가 아니라 재학습에서 온다.
- 크거나 결정적인 우위를 확립하지 **않는다**. 대부분 점수에서 경쟁 모델들과 매우 비슷하다. ’20 배 데이터로 학습한 모델을 이겼다’ 는 문장은 사실이지만, 하나의 벤치마크에서 대리 지표가 조금 움직인 정도다.
- 일반적인 예술 능력을 보여주지 **않는다**. 이는 224×224 픽셀이라는 작은 고정 해상도에서 아이콘과 단순 일러스트를 그리는 80 억 파라미터 연구 모델이지 범용 디자이너가 아니다.
- 대형 범용 모델 GPT-5 와의 비교는 **동일 조건 비교가 아니다**. GPT-5 는 이런 좁은 코드 기반 그림 과업에 맞춰 제작·조정된 모델이 아니므로 여기서 이긴다고 두 모델 전반에 대해 많은 것을 말해주지는 않는다.
- 비용이 없는 방법도 **아니다**. 매 단계 캔버스를 렌더링하고 다시 읽어야 하므로 코드를 한 번에 내보내는 블라인드 생성보다 느리다. 저자들도 이를 인정한다.

근거는 얼마나 탄탄한가?

- 자체 조건에서 통제 비교를 한 부분은 **탄탄하다**. 학습이나 검증을 제거하는 절제 (ablation) 실험에서 점수가 떨어지므로 두 구성요소가 저자들이 말한 역할을 실제로 하고 있음을 보여준다.
- 예상 밖 결과를 **솔직하게 보고한다**. 단순한 시각 피드백이 오히려 해롭다는 결과를 숨기지 않았고, 이것이 논문의 가장 흥미로운 부분이기도 하다. 입력이 많으면 항상 더 좋다는 직관을 유용하게 교정한다.
- 순위표에서 앞선다는 주장의 근거는 **제한적이다**. 더 많은 데이터를 쓴 경쟁 모델보다 높은 수치는 작고, 그 경쟁 모델 저자들이 만든 벤치마크 하나에서 나온 결과다. 하나의 시험 세트에서 대리 지표가 조금 앞선 것은 시사적이지만 결론을 내리기에는 부족하다.
- 규모와 실제 환경에서는 **아직 시험되지 않았다**. 여기서는 저해상도 아이콘과 단순 일러스트만 다룬다.

복잡한 고해상도 또는 실무 디자인에서도 같은 아이디어가 통하는지는 향후 과제이며 저자들도 그렇게 쓴다.

왜 중요한가

논문의 중심 아이디어는 민망할 만큼 단순해 보이지만 그림을 넘어선다. 프로그램이 코드를 작성해 무언가를 생성한다면, 웹페이지든 차트든 도식이든 3D 장면이든, 전체를 눈 감고 작성한 뒤 잘 되기를 바랄 수도 있고, 중간중간 렌더링해 보면서 방향을 수정할 수도 있다. 사람에게는 이 루프를 닫는 것이 당연하다. 우리는 계속 페이지를 흘끗 본다. 이런 모델에게는 의외로 최근에야 본격적으로 시도된 방식이다.

이 논문을 읽을 가치가 있게 만드는 것은 그 아이디어에 붙은 별표다. 모델에게 볼 수 있는 통로를 주는 것과 **보는 법을 가르치는 것**은 다르다. 눈은 훈련돼야 하고, 그 뒤에야 루프가 이득을 준다. 그 이득도 진짜이지만 절제돼 있다. 전혀 다른 종류의 예술가가 된 것이 아니라 더 안정적인 제도사가 된 것이다. 설명을 편집 가능한 그래픽으로 바꾸는 도구, 즉 앱의 아이콘이나 일러스트 뒤에서 일하는 시스템을 만드는 사람에게는 이 조용한 실용적 교훈이 중요하다. 여기서의 승리는 더 많은 데이터가 아니라 더 저렴하고 영리한 학습에서 나왔다. 순위표 한 칸을 올리는 것보다 배울 가치가 큰 결과다.

핵심 요약

웹 아이콘과 로고 뒤의 편집·확대 가능한 코드인 벡터 그래픽을 생성하는 언어 모델은 전통적으로 '눈을 감고' 작업했다. 그림 명령 전체를 작성하면서 결과를 한 번도 렌더링해 보지 않았다. Guotao Liang 연구팀은 Render-in-the-Loop 를 제안한다. 각 단계 후 미완성 그림을 렌더링해 모델에 다시 보여주고, 모델은 캔버스를 보면서 다음 획을 그린다. 핵심적이고 솔직한 결과는 기존 모델에 이 기능만 붙이면 성능이 오히려 나빠진다는 것이다. 시각 피드백을 사용하도록 모델을 재학습하고, 아무 변화도 만들지 않는 획을 버리는 생성 시점 검증을 더한 뒤에야 이득이 나타난다. 재학습한 80 억 파라미터 모델은 표준 벤치마크에서 최대 20 배 많은 데이터로 학습한 경쟁 모델과 비슷하거나 조금 앞섰다. 진짜 결과이지만 저해상도 아이콘·단순 일러스트를 대상으로 한 대리 점수의 작은 격차다. 저자들이 강조하는 핵심은 효율성이다. 작업 결과를 제대로 볼 줄 아는 것이 단순한 데이터 규모의 일부를 대신할 수 있다.

과장 없이 보면

논문이 보여주는 것: 벡터 그래픽 모델을 재학습해 미완성 그림을 단계마다 렌더링하고 스스로 보게 하면 눈을 감고 그리는 모델보다 더 완전하고 나은 결과를 내며, 더 적은 학습 데이터로 하나의 표준 벤치마크에서 대규모 데이터 경쟁 모델과 비슷하거나 약간 앞설 수 있다.

그럴듯하지만 입증되지 않은 것: ' 자기 작업을 보는 것' 이 일반적으로 데이터 규모를 늘리는 것보다 좋은 전략이라는 것, 복잡한 고해상도 또는 실제 그래픽에서도 같은 아이디어가 도움 된다는 것, 작은 벤치마크 차이가 사람이 실제로 알아볼 품질 차이를 뜻한다는 것.

보여주지 않는 것: 시각 피드백 자체가 도움이 된다는 것 (재학습 없이는 해롭다), 경쟁 모델보다 크고 결정적인 우위가 있다는 것, 범용 그림 능력, 이 과업용으로 만들어지지 않은 GPT-5 같은 범용 모델과의 공정한 일대일 비교.

주요 한계: 하나의 벤치마크, 그 일부는 경쟁 모델 저자들이 만든 것이라는 점, 대리 지표의 작은 격차, 80억 파라미터 모델과 224×224 해상도의 아이콘·단순 일러스트, 매 단계 렌더링 때문에 더 느린 생성.

일반 독자는 얼마나 신뢰해도 될까? 그리면서 렌더링하고 다시 보는 페루프가 실제로 도움이 되며, 단순히 기능을 켜는 것이 아니라 그 방식으로 학습해야 한다는 점에는 높은 신뢰를 둘 수 있다. 이 특정 모델이 경쟁 모델을 결정적으로 이겼다는 데에는 낮음 ~ 중간 정도가 적절하다. '20 배 데이터 모델을 이겼다' 는 표현은 사실이지만 하나의 벤치마크에서 나온 작은 차이로 읽어야 한다. 기억할 핵심에는 높은 신뢰를 둘 수 있다. **그림을 만드는 코드를 쓸 때는 눈을 감고 끝까지 쓰는 것보다 중간중간 보면서 쓰는 편이 낫다.**

출처

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/