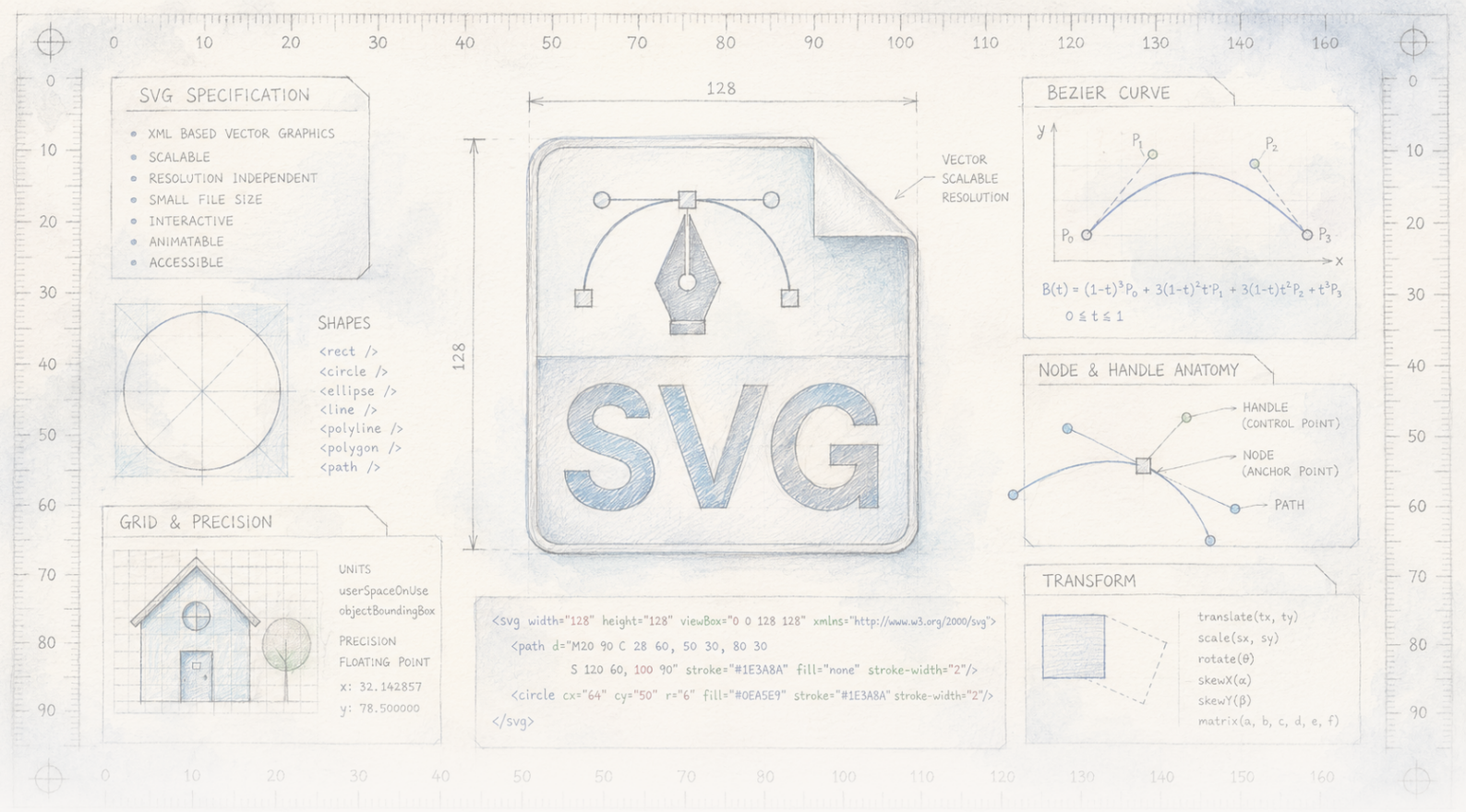


The Clean Paper

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

Să înveți o AI care desenează să se uite la pagină

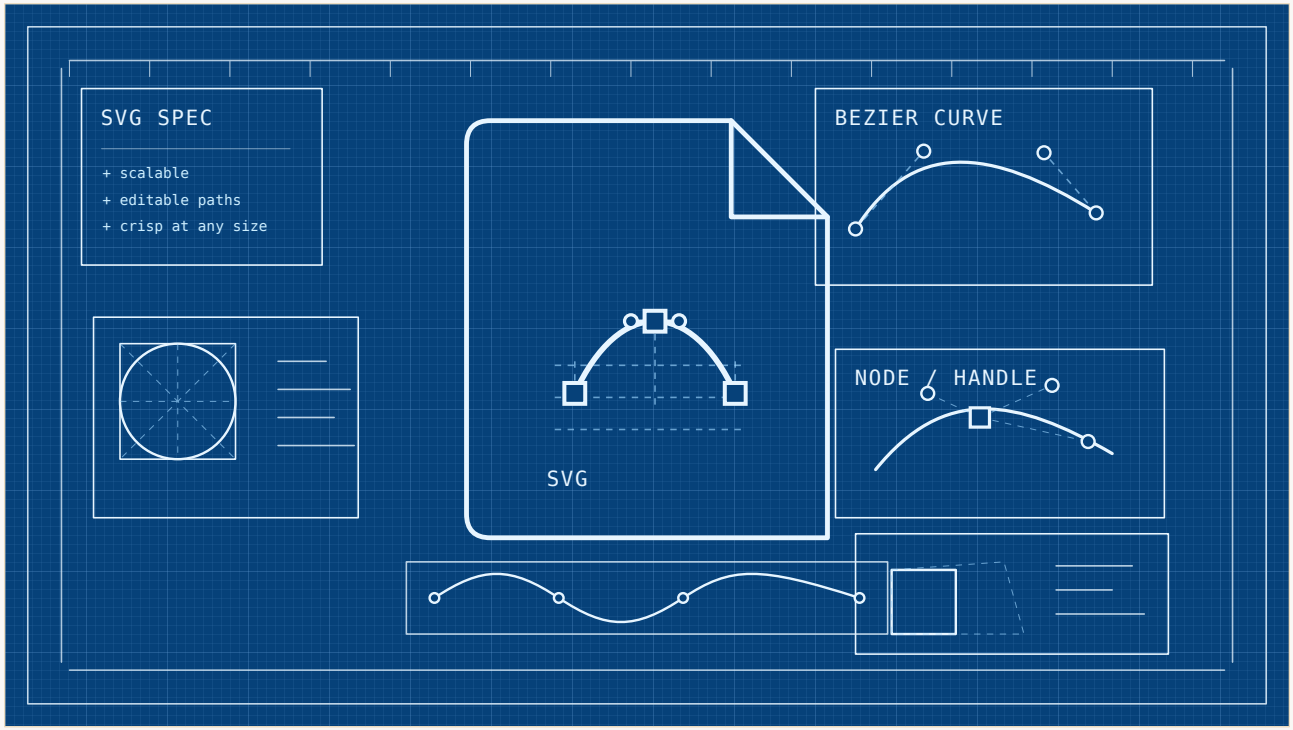
Modelele lingvistice care generează grafică vectorială o fac pe nevăzute: scriu comenzile desenului fără să vadă rezultatul. O metodă nouă lasă modelul să-și urmărească propria pânză cum se umple, trasă după trasă, iar întorsătura onestă este că simplul fapt de a-i da ochi înrăutățește lucrurile: trebuie reantrenat ca să-i folosească. Rezultatul este un model care egalează sau depășește ușor rivali antrenați cu până la de douăzeci de ori mai multe date — un rezultat real și atent pe un singur benchmark, nu o revoluție.

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)

Să desenezi cu ochii închiși

Cere-i unui model AI de astăzi să deseneze o imagine și, în interior, se poate întâmpla unul dintre două lucruri foarte diferite. Generatoarele de imagini obișnuite — cele care creează o imagine pornind de la o descriere — produc direct pixeli și pot „vedea” pânza pe măsură ce lucrează. Există însă și un alt mod de a desena, în care modelul nu pictează deloc. În schimb, scrie instrucțiuni: *desenează un cerc aici, o linie acolo, umple această formă cu albastru*. Instrucțiunile sunt cod — Scalable Vector Graphics, sau SVG, formatul aflat în spatele multor pictograme și logo-uri de pe web. Avantajul este important: rezultatul nu este o grilă fixă de pixeli, ci un ansamblu de forme care pot fi redimensionate, recolorate și editate fără să devină neclare.



Ilustrație originală în stil schiță tehnică SVG: imaginea este ea însăși un fișier vectorial editabil, construit din trasee, linii de grilă, noduri și mânere, nu din pixeli ficși.

Laura Nesso / The Clean Paper Permission on file

Problema este că, până recent, un model care scria acest cod de desen o făcea pe nevăzute. Emitea întreaga secvență de instrucțiuni dintr-o singură trecere — cerc, linie, umplere — fără să le randereze vreodată ca să vadă ce a produs. Imaginează-ți că schițezi o față cu ochii închiși: poate ai pune ochii aproximativ unde se pun ochii, dar nu ai avea cum să observi că al doilea a ajuns pe obraz, sau că forma desenată pentru păr stă acum peste nas. Cam așa lucrau aceste modele, motiv pentru care produceau atât de des cod perfect valid și totuși un dezastru vizual.

O echipă condusă de Guotao Liang a încercat o idee aproape evidentă: a lăsat modelul să-și „deschidă ochii”. Metoda lor, *Render-in-the-Loop*, randerează desenul parțial după fiecare pas și îi trimite imaginea înapoi modelului înainte ca acesta să genereze următorul element. Partea cu adevărat interesantă nu este faptul că metoda poate ajuta, ci ce a fost necesar pentru ca ea să funcționeze.

Cum notezi un desen?

Când o lucrare spune că modelul său "bate" alt model, este corect să întrebi: îl bate la ce, măsurat cum? Evaluarea unei imagini generate este cu adevărat grea — nu există un singur răspuns corect la "desenează un laptop" — așa că domeniul se sprijină pe câteva scoruri automate, fiecare un substitut imperfect pentru o persoană care privește rezultatul.
Câteva dintre aceste măsuri apar în lucrare. **FID** compară distribuția statistică generală a unui lot de imagini generate cu cea a imaginilor reale; o valoare mai mică este mai bună, dar scorul descrie lotul, nu o imagine anume. **Scorul CLIP** estimează cu ajutorul unui alt model cât de bine corespunde imaginea cuvintelor din prompt — util, dar limitat de calitatea evaluatorului. **DINO**, **SSIM** și **LPIPS** compară o imagine reconstruită cu o țintă, de la diferențe directe la nivel de pixel până la caracteristici învățate.
Niciunul dintre aceste scoruri nu reprezintă un adevăr absolut. Sunt indicatori indirecți, iar diferențele dintre modele tind să fie mici. Dacă un model obține 127,6 iar altul 128,8, diferența poate fi reală și în direcția dorită, dar rămâne modestă și apare într-o măsură care aproximează doar parțial ceea ce ar observa un om. Merită ținut minte când titlul devine „bate un model antrenat pe de douăzeci de ori mai multe date".

Ce au făcut autorii

Autorii au vrut să rezolve tocmai acest „desen pe nevăzute". Modelele existente tratează generarea SVG ca pe o sarcină pur textuală: prezic următorul fragment de cod din ceea ce au scris deja, fără să-l randereze. Astfel rămâne nefolosită componenta vizuală puternică — codificatorul vizual — pe care modelele multimodale moderne o au deja. Render-in-the-Loop transformă procesul într-unul vizual, pas cu pas. După fiecare fragment de cod, SVG-ul parțial este randat și trimis modelului ca imagine, astfel încât următorul fragment poate fi ales după ce modelul „vede" pânza obținută până atunci.

Prima constatare este și un avertisment: simpla adăugare a acestei bucle la un model general existent nu funcționează. Când autorii au oferit randări intermediare unor modele puternice, dar neantrenate special pentru această procedură, calitatea nu s-a îmbunătățit — s-a *degradat* în toate cazurile. Un model care nu a învățat să folosească informația vizuală în acest scop nu capătă spontan această abilitate.

De aceea, cea mai mare parte a muncii ține de antrenare. Autorii reconstruiesc datele astfel încât fiecare desen să fie împărțit în pași mici și semnificativi vizual — formele complexe sunt descompuse în elemente mai simple, astfel încât la fiecare etapă să apară ceva nou de observat — apoi fac fine-tuning unui model deschis cu opt miliarde de parametri (bazat pe Qwen3-VL) pe aceste secvențe. Metoda se numește Visual Self-Feedback. Ei adaugă și un al doilea mecanism în timpul generării, Render-and-Verify: înainte de a accepta fiecare element nou, modelul îl randerează și verifică dacă imaginea s-a schimbat efectiv sau dacă a repetat pasul anterior. Elementele care nu adaugă nimic sunt eliminate, iar când nu mai apare nicio îmbunătățire, modelul este determinat să se oprească. Toate acestea folosesc un set de date relativ mic — aproximativ 850.000 de exemple, mai puțin de jumătate din volumul folosit de un rival și doar o mică fracțiune din cel folosit de altul.

Ce au găsit

- Antrenat astfel, modelul desenează mai bine decât perechea sa oarbă — cel mai vizibil în cazurile de eșec, când un model orb pune un ochi pe obraz sau sare peste un grafic cu bare cerut și produce în schimb un monitor generic.
- Pe testul standard MMSVGBench, metoda este competitivă cu rivali puternici și, după mai multe măsuri, ușor înainte — inclusiv OmniSVG, antrenat pe mai mult de dublul datelor, și InternSVG, antrenat pe aproximativ de douăzeci de ori mai multe.
- Marjele sunt mici. Pe setul de pictograme, de exemplu, scorul principal de calitate a imaginii este 127,6 față de 128,8 pentru cel mai bun rival, iar scorul de potrivire cu promptul este 0,293 față de 0,291 — o diferență coerentă, dar modestă.
- Ambele piese adăugate își merită locul: oprește antrenamentul special sau verificarea din timpul desenului, iar numerele scad măsurabil. Pasul de verificare în special oprește modelul să rămână blocat redesenând același lucru iar și iar.
- Rezultatul pe care autorii înșiși îl subliniază este eficiența — să ajungi atât de departe cu mult mai puține date de antrenament decât au folosit liderii.

Ce nu demonstrează

- **Nu** arată că a lăsa un model să “vadă” este un câștig gratuit. Dimpotrivă: experimentul propriu al lucrării arată că feedbackul vizual *fără* reantrenare înrăutățește lucrurile. Câștigul vine din reantrenare, nu doar din ochi.
- **Nu** stabilește un avans mare sau decisiv. După cele mai multe scoruri, metoda este foarte apropiată de rivali; afirmația „bate un model antrenat pe 20× mai multe date” este adevărată, dar se bazează pe diferențe mici în indicatori indirecți, pe un singur test de referință.
- **Nu** demonstrează abilitate artistică generală. Acesta este un model de cercetare de opt miliarde de parametri care desenează pictograme și ilustrații simple la o rezoluție fixă mică (224×224 pixeli), nu un designer general.
- Comparația cu un model general mare (GPT-5) **nu** este una direct echivalentă: acel model nu este construit sau ajustat pentru această sarcină îngustă de desen prin cod, astfel încât faptul că este depășit aici spune puțin despre performanța generală a oricăruia dintre modele.
- **Nu** vine gratis. Randarea și recitirea pânzei la fiecare pas face generarea mai lentă decât emiterea codului într-o singură trecere oarbă — un cost pe care autorii îl recunosc.

Cât de solide sunt dovezile?

- **Solide** acolo unde comparația este controlată în condițiile definite de autori. Testele de ablație sunt clare: dacă este eliminată antrenarea specială sau etapa de verificare, scorurile scad, ceea ce arată că ambele componente contribuie la rezultat.
- **Transparentă în privința rezultatului neașteptat.** Constatarea că feedbackul vizual adăugat naiv

dăunează este raportată explicit și este unul dintre cele mai interesante rezultate ale lucrării — o corecție utilă a intuiției că mai multă informație de intrare este întotdeauna mai bună.
- **Mai slabe când este vorba despre poziția în clasament.** Avantajele față de rivalii antrenați cu mai multe date sunt mici și apar pe un singur test, construit de autorii unuia dintre acei rivali. Diferențele mici în indicatori indirecți, pe un singur set de testare, sunt sugestive, nu decisive.
- **Netestată la scară mare și în utilizare reală.** Aici totul se referă la pictograme și ilustrații simple, la rezoluție joasă. Rămâne de văzut dacă aceeași idee funcționează pentru grafică complexă, la rezoluție înaltă sau în activități reale de design; autorii spun explicit acest lucru.

De ce contează

Ideea din centrul acestei lucrări este aproape stânjenitor de simplă și ajunge mult dincolo de desen. Dacă un program va genera ceva scriind cod — o pagină web, un grafic, o diagramă, o scenă 3D — poate fie să scrie totul pe nevăzute și să spere, fie să randereze pe măsură ce merge și să corecteze cursul. Închiderea acestei bucle este firească pentru o persoană: ne uităm continuu la ceea ce desenăm. Pentru aceste modele, însă, este o idee surprinzător de recentă.

Ceea ce face lucrarea interesantă este rezerva pe care o adaugă: a-i oferi unui model posibilitatea de a vedea nu este același lucru cu a-l învăța să privească. Componenta vizuală trebuie antrenată pentru această sarcină și abia atunci bucla aduce beneficii. Chiar și atunci, câștigul este real, dar moderat: un sistem de desen mai stabil, nu un nou tip de artist. Pentru oricine construiește unelte care transformă o descriere într-o grafică editabilă — genul de lucru care ajunge în spatele pictogramelor și ilustrațiilor unei aplicații — lecția liniștită și practică este cea utilă. Câștigul de aici a venit din antrenament mai ieftin și mai inteligent, nu din mai multe date. Este un lucru mai bun de învățat decât încă un punct pe un clasament.

Pe scurt

Modelele lingvistice care generează grafică vectorială — codul editabil și scalabil din spatele celor mai multe pictograme și logo-uri de pe web — au făcut-o tradițional “pe nevăzute”, scriind toate comenzile desenului fără să le randereze vreodată ca să vadă rezultatul. O echipă condusă de Guotao Liang propune Render-in-the-Loop: randarea desenului pe jumătate terminat după fiecare pas și alimentarea lui înapoi, astfel încât modelul desenează următoarea trasă privind pânza. Constatarea lor centrală și importantă este că simpla aplicare a acestui lucru unui model existent îl face *mai rău*; câștigul apare doar după ce modelul este reantrenat să folosească feedbackul vizual, ajutat de o verificare în timpul desenului care aruncă trasele ce nu schimbă nimic. Modelul reantrenat de opt miliarde de parametri egalează sau depășește ușor rivali antrenați pe până la de douăzeci de ori mai multe date pe un test de referință standard — un rezultat autentic, dar cu marje mici pe indicatori indirecți, pentru pictograme și ilustrații simple la rezoluție joasă. Concluzia pe care autorii o subliniază, și cea care merită păstrată, este despre eficiență: să-ți vezi bine munca poate ține loc, parțial, de scara brută a datelor.

Verificare fără exagerări

Ce arată lucrarea: Reantrenarea unui model de grafică vectorială ca să randereze și să-și privească propriul desen pe jumătate terminat, pas cu pas, produce rezultate mai bune și mai complete decât desenul pe nevăzute — competitive cu, și ușor înaintea, unor rivali cu mai multe date pe un test de referință standard, din mai puține date de antrenament.

Ce este plauzibil, dar nedovedit: Că "a-ți vedea munca" este în general o rețetă mai bună decât scalarea datelor; că aceeași idee va ajuta pe grafică complexă, de rezoluție înaltă sau reală; că marjele mici din testul de referință reflectă o diferență pe care o persoană ar observa-o cu adevărat.

Ce nu arată: Că feedbackul vizual ajută singur (fără reantrenare dăunează); că avansul față de rivali este mare sau decisiv; abilitate de desen generală; o comparație corectă directă cu modele generale precum GPT-5, care nu sunt construite pentru această sarcină.

Limitări principale: Un singur test de referință, construit parțial de autorii unui rival; marje mici după indicatori indirecți; un model de 8B, pictograme și ilustrații simple la 224×224; generare mai lentă din cauza buclei de randare de la fiecare pas.

Câtă încredere ar trebui să aibă un cititor general? Mare că închiderea buclei — randare și recitire pe măsură ce desenezi — ajută cu adevărat, și că trebuie antrenată, nu doar pornită. Mică spre moderată că acest model specific își bate decisiv rivalii; tratează linia "bate 20× datele" ca pe un rezultat real, dar modest, pe un singur test de referință. Mare în ideea care merită ținută minte: pentru codul care desenează, să privești pe măsură ce mergi bate desenatul pe nevăzute.

Surse

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/