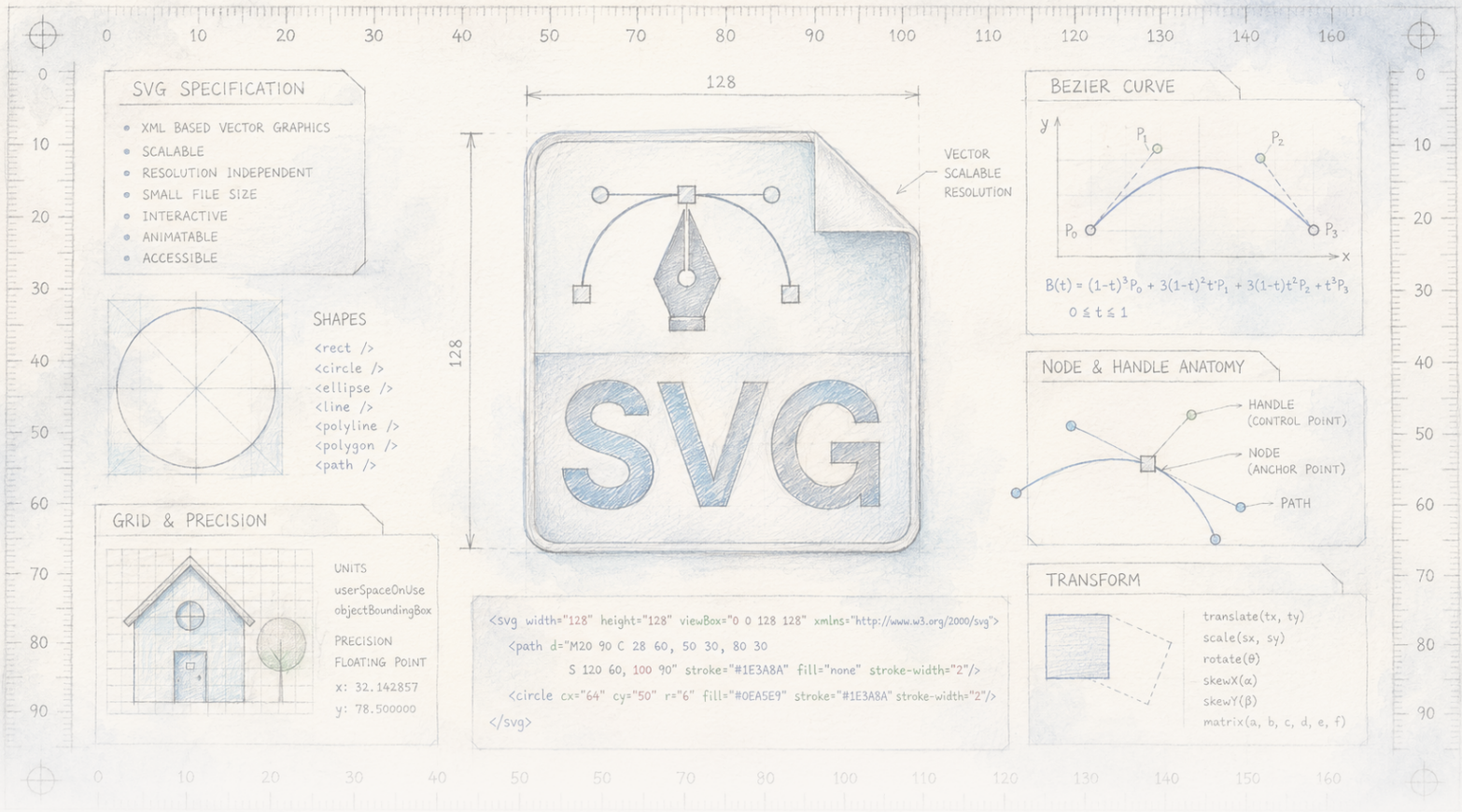


The Clean Paper

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

Kako naučiti AI koji crta da gleda stranicu

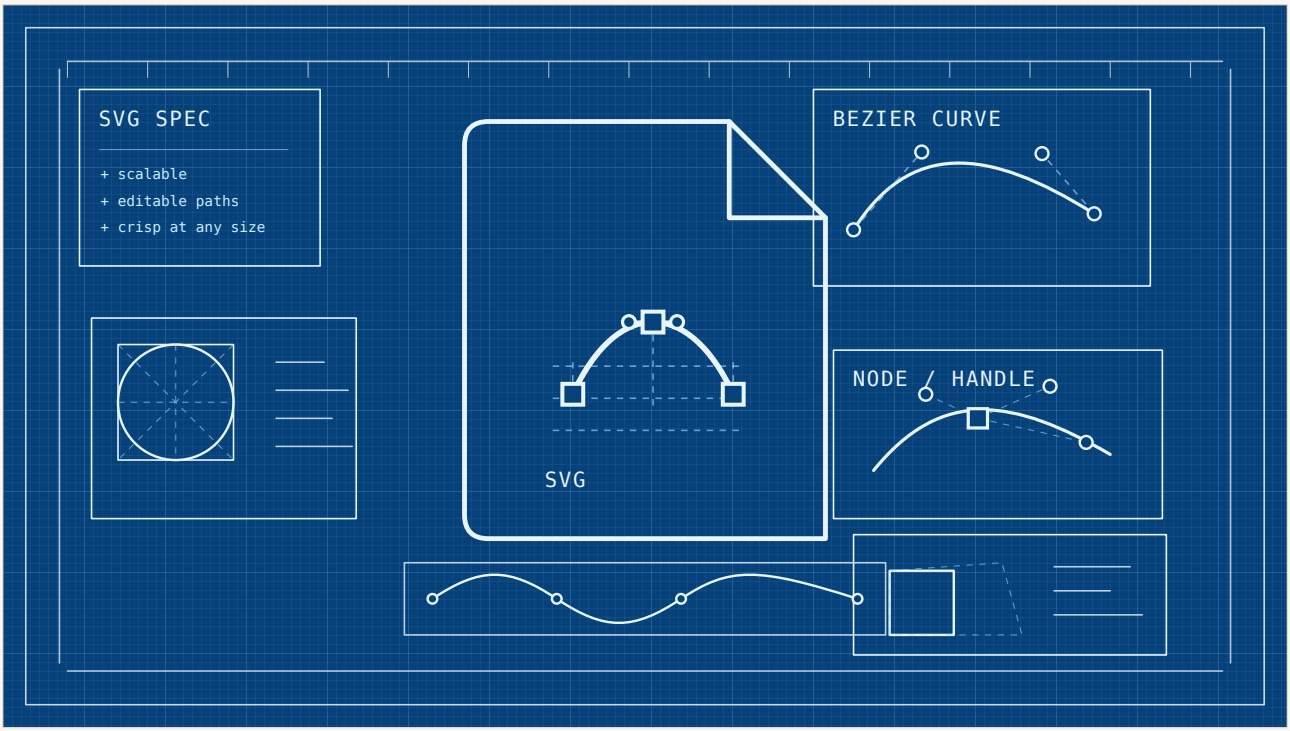
Jezični modeli koji generiraju vektorsku grafiku rade to naslijepo — ispisuju naredbe za crtanje a da nikada ne vide rezultat. Nova metoda dopušta modelu da gleda kako se njegovo platno ispunjava potez po potez, a pošteni obrat jest da samo davanje očiju pogoršava rezultat: model se mora ponovno trenirati da ih koristi. Dobit je model koji se izjednačuje ili blago nadmašuje suparnike trenirane na do dvadeset puta više podataka — stvaran, pažljiv rezultat na jednom benchmarku, ne revolucija.

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)

Crtanje zatvorenih očiju

Zatražite li od današnjeg AI modela da nacrta sliku, ispod površine događa se jedna od dvije vrlo različite stvari. Poznati generatori slika — oni koji iz jedne rečenice stvore fotografiju — izravno slikaju piksele i mogu "vidjeti" platno dok rade. Ali postoji i drugi, tiši način crtanja, u kojem model uopće ne slika. Piše upute: *nacrtaj krug ovdje, crtu ondje, ispuni ovaj oblik plavom*. Te su upute kod — isti Scalable Vector Graphics, odnosno SVG, koji stoji iza većine ikona i logotipa na webu — a prednost je stvarna: rezultat nije fiksna mreža piksela nego skup oblika koje možete beskonačno povećavati, mijenjati im boju i uređivati bez zamućivanja.



Izvorna SVG ilustracija tehničkog nacrta: sama slika je uređiva vektorska datoteka, izgrađena od putanja, mrežnih linija, čvorova i ručica umjesto fiksnih piksela.

Laura Nesso / The Clean Paper Permission on file

Problem je u tome što je donedavno model takav kod za crtanje pisao slijepo. Izbacivao je cijeli niz uputa u jednom prolazu — krug, crta, ispuna — a da ih nikada ne bi renderirao i pogledao što je zapravo napravio. Zamislite da skicirate lice zatvorenih očiju: možda biste oči otprilike stavili gdje trebaju biti, ali ne biste mogli primijetiti da je drugo završilo na obrazu ili da se oblik kose sada nalazi preko nosa. Otprilike tako su radili ti modeli, zbog čega su često proizvodili savršeno valjan kod koji je vizualno bio nered.

Tim koji vodi Guotao Liang sada je učinio gotovo komično očitu stvar: dopustio je modelu da otvori oči. Njihova metoda, *Render-in-the-Loop*, nakon svakog koraka renderira napola dovršen crtež i vraća tu sliku modelu prije nego što nacrta sljedeći potez. Doista zanimljiv dio nije to što ovo pomaže — nego što su morali učiniti da bi uopće počelo pomagati.

Kako se boduje crtež?

Kada rad kaže da njegov model "pobjeđuje" drugi, pošteno je pitati: pobjeđuje u čemu i kako se to mjeri? Ocjenjivanje generirane slike doista je teško — ne postoji jedan jedini ispravan odgovor na "nacrtaj prijenosno računalo" — pa se područje oslanja na nekoliko automatskih metrika, od kojih je svaka nesavršena zamjena za osobu koja gleda rezultat.
Nekoliko ih se pojavljuje u ovom radu. **FID** uspoređuje ukupni statistički karakter cijele skupine generiranih slika sa stvarnima; niže je bolje, ali opisuje skupinu, ne pojedinu sliku. **CLIP score** pita zasebni AI odgovara li slika riječima u upitu — korisno, ali samo koliko je dobar taj sudac. **DINO**, **SSIM** i **LPIPS** uspoređuju rekonstruiranu sliku s ciljem, na razinama od sirovih piksela do naučenih značajki.
Ništa od toga nije istina sama po sebi. To su zamjenske metrike i obično se pomiču u malim koracima. Kada pročitate da jedan model ima 127.6, a drugi 128.8, to je stvarna razlika u željenom smjeru — ali mali pomak, ne odron, i to u broju koji tek labavo prati ono što bi vaše oko reklo. Vrijedi to zadržati na umu kada naslov glasi "pobjeđuje model treniran na dvadeset puta više podataka".

Što su autori učinili

Problem koji su autori pokušali riješiti upravo je slijepo crtanje. Postojeći modeli pisanje SVG-a tretiraju kao čisti tekstualni zadatak: predvidi sljedeći komad koda iz dosadašnjeg koda i nikada ga ne renderiraj. Time ostaju neiskorištene moćne "oči" — vizualni enkoder — koje suvremeni multimodalni modeli već imaju. Render-in-the-Loop posao preoblikuje u vizualni postupak korak po korak. Nakon svakog dijela koda za crtanje djelomični SVG renderira se u sliku i vraća modelu, pa sljedeći dio bira dok stvarno gleda dotadašnje platno.

Prvi nalaz je upozoravajući i, njima u prilog, autori ga navode otvoreno: jednostavno priključiti tu petlju na postojeći gotovi model ne radi. Kada su snažnim općim modelima bez posebnog treniranja davali međurezultate renderiranja, kvaliteta se nije poboljšala — *pogoršala se* u svim slučajevima. Model koji nikada nije naučen koristiti oči za ovaj zadatak ne zna to odjednom sam od sebe.

Zato je većina rada u podučavanju. Ponovno grade podatke za treniranje tako da je svaki crtež razloml-jen na mnogo malih, vizualno smislenih koraka — složene oblike rastavljaju na jednostavnije dijelove kako bi u svakoj fazi postojalo nešto novo za vidjeti — a zatim na tim sekvencama dodatno treniraju otvoreni model s osam milijardi parametara (izgrađen na Qwen3-VL). To nazivaju Visual Self-Feedback. Dodaju i drugi mehanizam tijekom crtanja, Render-and-Verify: prije prihvaćanja svakog novog poteza model ga renderira i provjerava je li doista promijenio sliku ili samo ponovio prethodni. Potezi koji ništa ne dodaju odbacuju se, a kada ništa više ne pomaže, model dobiva uputu da stane. Važno je da sve to radi na relativno malom skupu — oko 850,000 primjera, manje od polovice količine koju je koristio jedan konkurent i mali dio podataka drugoga.

Što su pronašli

- Ovako treniran model crta bolje od svojeg slijepog pandana — najvidljivije u slučajevima pogreške, kada slijepi model stavi oko na obraz ili preskoči traženi stupčasti grafikon i umjesto njega nacrta generički monitor.
- Na standardnom benchmarku (MMSVGBench) metoda je konkurentna snažnim suparnicima i prema nekoliko mjera malo ispred njih — uključujući OmniSVG, treniran na više nego dvostruko podataka, i InternSVG, treniran na približno dvadeset puta više.
- Razlike su male. Na skupu ikona, primjerice, glavna ocjena kvalitete slike iznosi 127.6 prema 128.8 najboljeg suparnika, a ocjena podudaranja s upitom 0.293 prema 0.291 — stvarno i dosljedno, ali tijesno.
- Oba dodatka opravdavaju svoje mjesto: isključite li posebno treniranje ili provjeru tijekom crtanja, brojke mjerljivo padaju. Korak provjere osobito sprječava model da zapne u ponovnom crtanju iste stvari.
- Rezultat koji sami autori naglašavaju jest učinkovitost — doći ovako daleko s mnogo manje podataka za treniranje nego što su ih koristili vodeći modeli.

Što ovo ne dokazuje

- **Ne** pokazuje da je dopustiti modelu da "vidi" besplatna pobjeda. Upravo suprotno: vlastiti eksperiment rada pokazuje da vizualna povratna informacija *bez* ponovnog treniranja pogoršava rezultat. Dobit dolazi iz treniranja, ne iz samih očiju.
- **Ne** utvrđuje veliku ili odlučujuću prednost. Prema većini metrika metoda je gotovo izjednačena sa suparnicima; "pobjeđuje model treniran na 20× više podataka" točno je, ali riječ je o malim pomacima zamjenskih metrika na jednom benchmarku.
- **Ne** pokazuje opću umjetničku sposobnost. Ovo je istraživački model s osam milijardi parametara koji crta ikone i jednostavne ilustracije u maloj fiksnoj rezoluciji (224×224 piksela), ne općenamjenski dizajner.
- Usporedba s velikim općim modelom (GPT-5) **nije** usporedba istih uvjeta: taj model nije izrađen ni podešen za uski zadatak crtanja kodom, pa njegovo nadmašivanje ovdje malo govori o bilo kojem od modela općenito.
- **Ne** dolazi bez troška. Renderiranje i ponovno čitanje platna pri svakom koraku čini generiranje sporijim od izbacivanja cijelog koda u jednom slijepom prolazu — cijena koju autori priznaju.

Koliko su dokazi jaki

- **Čvrsti** su ondje gdje je riječ o kontroliranoj usporedbi pod vlastitim uvjetima. Ablacije su jasne: uklonite treniranje ili provjeru i brojke padaju — dakle ta dva sastojka doista rade ono što autori tvrde.
- **Pošteni prema vlastitom iznenađenju.** Nalaz da naivna vizualna povratna informacija *šteti* ob-

javljen je, ne skriven, i možda je najzanimljivija stvar u radu — koristan ispravak intuicije da je više ulaza uvijek bolje.
- **Tanki** su ondje gdje se tvrdnja svodi na ljestvicu rezultata. Dobici nad suparnicima treniranima na više podataka mali su i postoje na jednom benchmarku koji su izgradili autori jednog od tih suparnika. Male razlike zamjenskih metrika na jednom testnom skupu sugestivne su, ne konačne.
- **Neispitani pri velikom razmjeru i u stvarnom svijetu.** Ovdje su sve ikone i jednostavne ilustracije male rezolucije. Hoće li ista ideja vrijediti za složen, visokorezolucijski ili stvarni dizajnerski rad ostaje budući posao — autori to i sami kažu.

Zašto je važno

Ideja u središtu rada gotovo je neugodno jednostavna i ide mnogo dalje od crtanja. Ako program nešto generira pisanjem koda — web-stranicu, grafikon, dijagram, 3D scenu — može ili napisati sve naslijepo i nadati se ili renderirati tijekom rada i ispravljati smjer. Zatvaranje te petlje čovjeku je druga priroda; stalno bacamo pogled na stranicu. Za ove modele to je iznenađujuće nov potez.

Vrijednim čitanja rad čini zvjezdica koju dodaje toj ideji. Dati modelu način da vidi nije isto što i naučiti ga gledati. Oči moraju biti istrenirane, i tek tada petlja počinje donositi korist — a i tada je korist stvarna, ali odmjerena: pouzdaniji crtač, ne druga vrsta umjetnika. Za svakoga tko gradi alate koji opis pretvaraju u uređivu grafiku — onu vrstu koja završi iza ikona i ilustracija neke aplikacije — korisna je upravo tiha, praktična lekcija. Dobit je ovdje došla iz jeftinijeg i pametnijeg treniranja, ne iz više podataka. To je bolja pouka od još jednog boda na ljestvici.

Sažetak bez prenaglašavanja

Jezični modeli koji generiraju vektorsku grafiku — uređiv i skalabilan kod iza većine web-ikona i logotipa — tradicionalno su to radili "slijepo", ispisujući sve naredbe za crtanje a da ih nikada ne renderiraju i pogledaju rezultat. Tim Guotao Lianga predlaže Render-in-the-Loop: nakon svakog koraka renderirati napola dovršen crtež i vratiti ga modelu, tako da sljedeći potez nacrta dok gleda platno. Njihov središnji, pošteni nalaz jest da jednostavno dodavanje toga postojećem modelu čini rezultat *lošijim*; korist se pojavljuje tek nakon što je model ponovno treniran da koristi vizualnu povratnu informaciju, uz pomoć provjere tijekom crtanja koja odbacuje poteze koji ništa ne mijenjaju. Ponovno trenirani model s osam milijardi parametara izjednačuje se ili blago nadmašuje suparnike trenirane na do dvadeset puta više podataka na standardnom benchmarku — stvaran rezultat, ali s malim razlikama na zamjenskim metrikama, za ikone i jednostavne ilustracije niske rezolucije. Zaključak koji autori naglašavaju, i koji vrijedi zadržati, tiče se učinkovitosti: dobro gledati vlastiti rad može nadomjestiti dio gole skale podataka.

Provjera bez uljepšavanja

Što rad pokazuje: Ponovno treniranje modela vektorske grafike da korak po korak renderira i gleda vlastiti napola dovršeni crtež daje bolje i potpunije rezultate nego crtanje naslijepo — konkurentno, i malo bolje od suparnika s više podataka na jednom standardnom benchmarku, uz manje podataka za treniranje.

Što je moguće, ali nije dokazano: Da je "gledanje vlastitog rada" općenito bolji recept od skaliranja podataka; da će ista ideja pomoći složenoj, visokorezolucijskoj ili stvarnoj grafici; da male razlike na benchmarku predstavljaju razliku koju bi čovjek doista primijetio.

Što ne pokazuje: Da vizualna povratna informacija pomaže sama od sebe (bez ponovnog treniranja šteti); da je prednost nad suparnicima velika ili odlučujuća; općenamjensku sposobnost crtanja; poštenu izravnu usporedbu s općim modelima poput GPT-5, koji nisu izrađeni za ovaj zadatak.

Glavna ograničenja: Jedan benchmark, djelomično napravljen od autora suparničkog modela; male razlike na zamjenskim metrikama; 8B model, ikone i jednostavne ilustracije pri 224×224; sporije generiranje zbog petlje koja renderira svaki korak.

Koliko povjerenja treba imati opći čitatelj? Visoko da zatvaranje petlje — renderiranje i ponovno gledanje tijekom crtanja — doista pomaže i da se to mora istrenirati, a ne samo uključiti. Nisko do srednje da ovaj konkretni model odlučno nadmašuje suparnike; tvrdnju "pobjeđuje s 20× manje podataka" treba tretirati kao stvaran, ali skroman rezultat na jednom benchmarku. Visoko za ideju koju vrijedi zapamtiti: kod koji crta radi bolje kada gleda tijekom rada nego kada crta naslijepo.

Izvori

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/