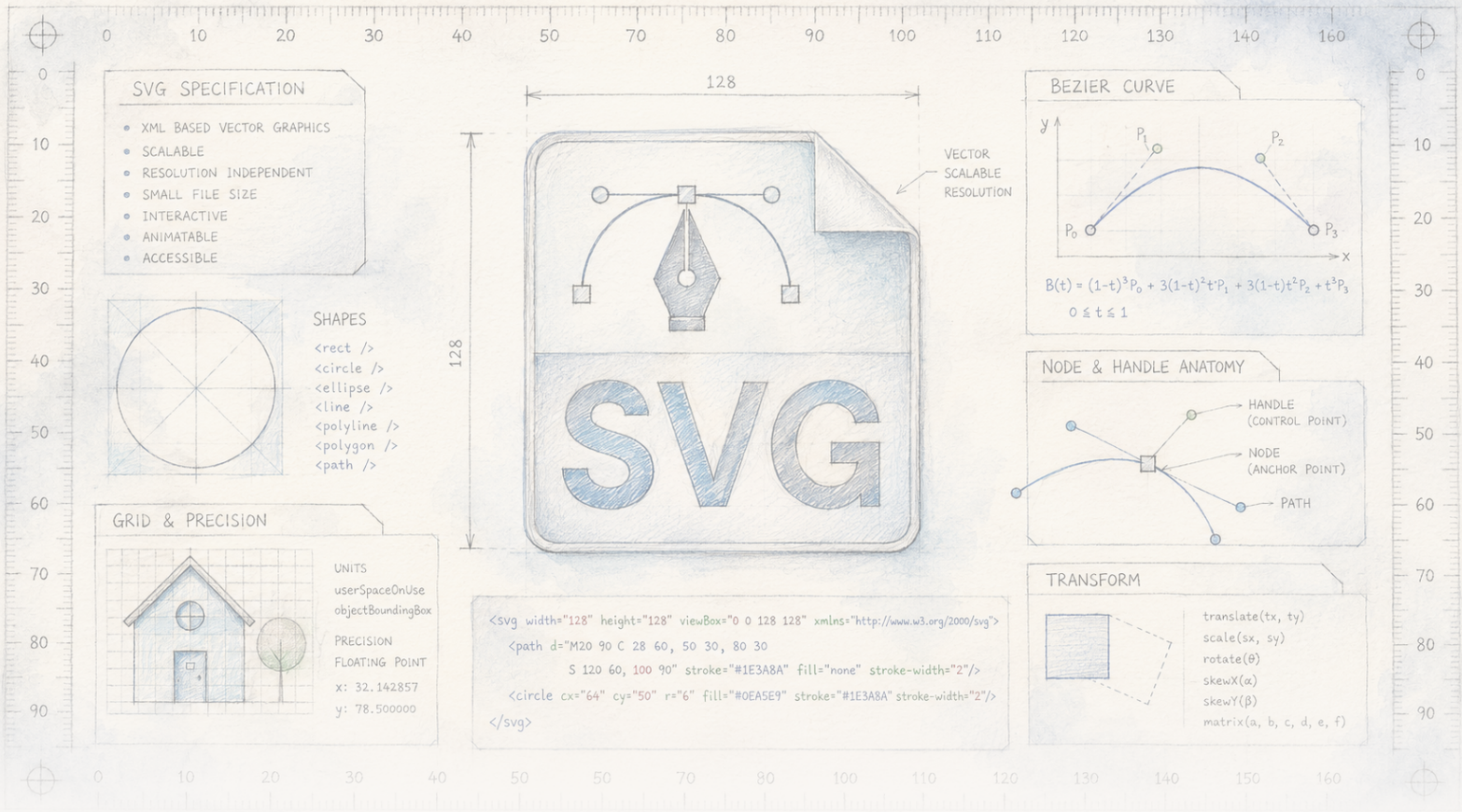


The
CLEAN PAPER

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

T'i mësosh një AI-je që vizaton të shohë faqen

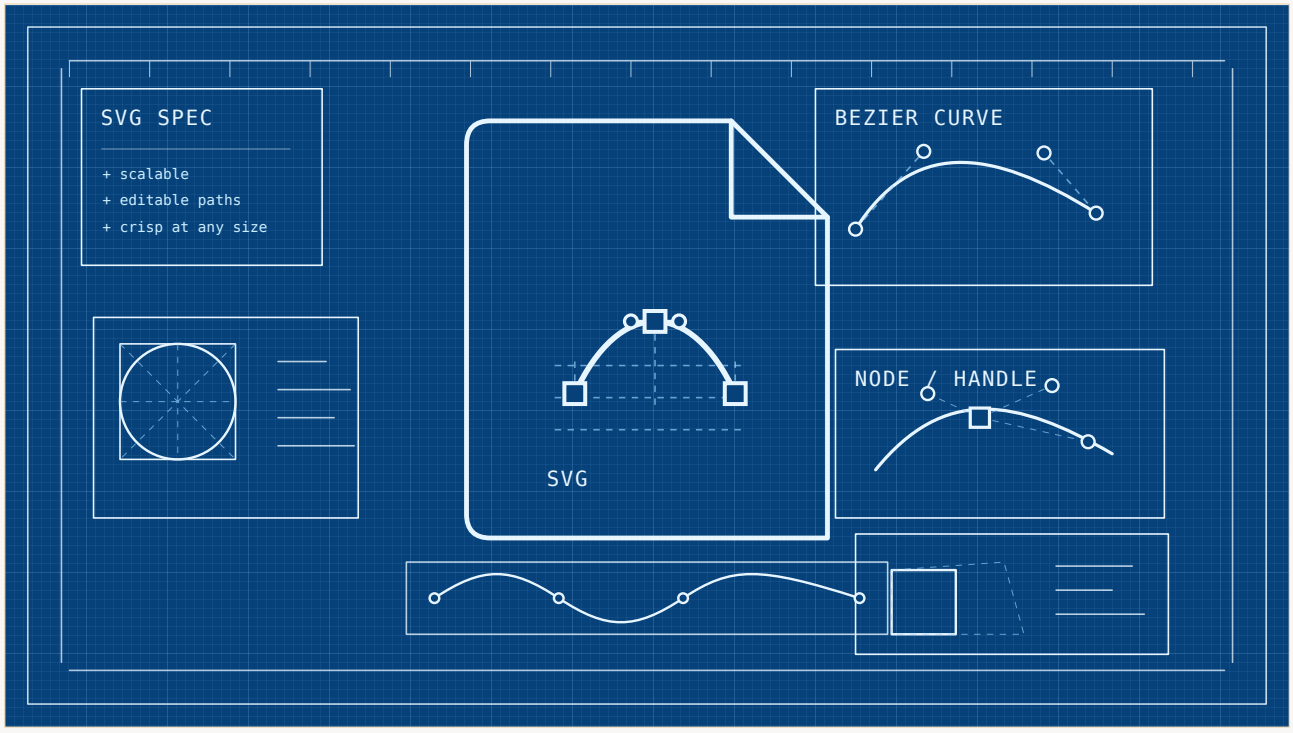
Modelet gjuhësore që gjenerojnë grafikë vektoriale zakonisht punojnë verbërisht — shkruajnë komandat e vizatimit pa parë kurrë rezultatin. Një metodë e re e lejon modelin ta shohë kanavacën hap pas hapi, por kthesa e ndershme është se thjesht t'i japësh sy e përkeqëson performancën: duhet ritrajnuar që t'i përdorë. Pas kësaj, një model 8B barazon ose kalon lehtë rivalë të trajnuar me deri në njëzet herë më shumë të dhëna — rezultat real dhe i kujdesshëm në një benchmark, jo revolucion.

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)

Të vizatosh me sy mbyllur

Kur i kërkon një modeli modern AI të vizatojë një figurë, nën kapak ndodh njëra nga dy gjëra shumë të ndryshme. Gjeneratorët e njohur të imazheve — ata që krijojnë fotografi nga një fjali — pikturojnë drejtpërdrejt pikselat dhe mund ta "shohin" kanavacën gjatë punës. Por ka edhe një lloj të dytë, më të qetë vizatimi, ku modeli nuk pikturon fare. Ai shkruan udhëzime: *vizato një rreth këtu, një vijë aty, mbushe këtë formë me blu*. Udhëzimet janë kod — i njëjti Scalable Vector Graphics, ose SVG, që qëndron pas shumicës së ikonave dhe logove në web — dhe avantazhi është real: rezultati nuk është rrjet fiks pikselash, por një grup formash që mund t'i zmadhosh, rilyesh dhe redaktosh pafund pa u bërë të turbullta.



Vizatim origjinal SVG në stil projekti teknik: vetë imazhi është skedar vektorial i redaktueshëm, i ndërtuar nga shtigje, vija rrjeti, nyje dhe doreza, jo pikselë fiks.

Laura Nesso / The Clean Paper Permission on file

Problemi është se, deri vonë, një model që shkruante këtë kod vizatimi e bënte verbërisht. Nxirrte gjithë sekuencën e udhëzimeve me një kalim — rreth, vijë, mbushje — pa i renderuar për të parë çfarë kishte bërë. Imagjino të skicosh një fytyrë me sy mbyllur: mund t'i vendosësh sytë afërsisht aty ku duhet, por nuk ke si të vësh re se i dyti përfundoi në faqe, apo se forma e flokëve mbuloi hundën. Kështu punonin këto modele dhe prandaj shpesh prodhonin kod krejtësisht valid, por vizualisht të çrregullt.

Një ekip i udhëhequr nga Guotao Liang bëri tani gjënë pothuajse qesharakisht të dukshme: ia hapi modelit sytë. Metoda e tyre, *Render-in-the-Loop*, renderon vizatimin gjysmë të përfunduar pas çdo hapi dhe ia jep imazhin modelit para goditjes tjetër. Pjesa vërtet interesante nuk është që kjo ndihmon — është çfarë duhej bërë që të ndihmonte fare.

Si vlerësohet një vizatim?

Kur një punim thotë se modeli "mund" një tjetër, duhet pyetur: në çfarë dhe si u mat? Vlerësimi automatik i një figure të gjeneruar është i vështirë, sepse nuk ka një përgjigje të vetme të saktë për "vizato një laptop". Prandaj fusha përdor disa tregues automatikë, secili vetëm zëvendësues i papërsosur i një njeriu që sheh rezultatin.
Në këtë punim shfaqen disa. **FID** krahason karakteristikat statistikore të një grupi të tërë imazhesh të gjeneruara me ato reale; më i ulët është më mirë, por përshkruan grupin, jo një imazh të vetëm. **CLIP score** pyet një AI tjetër nëse imazhi përputhet me tekstin e kërkesës — i dobishëm, por i kufizuar nga gjykatësi. **DINO**, **SSIM** dhe **LPIPS** krahasojnë një imazh të rindërtuar me objektivin, nga pikselat e papërpunuar te veçoritë e mësuara.
Asnjë nuk është e vërteta vetë. Janë proxy dhe shpesh lëvizin me hapa të vegjël. Kur një model shënon 127.6 dhe një tjetër 128.8, ndryshimi është real në drejtimin e synuar — por është një shtytje e vogël, jo rrëshqitje masive, dhe në një numër që vetëm përafërsisht ndjek atë që do të thoshte syri yt. Kjo vlen të mbahet mend kur titulli është "mund një model të trajnuar me njëzet herë më shumë të dhëna".

Çfarë bënë autorët

Problemi që synuan të rregullonin ishte vizatimi i verbër. Modelet ekzistuese e trajtojnë shkrimin SVG si detyrë thjesht tekstuale: parashikojnë copën tjetër të kodit nga kodi i deritanishëm dhe nuk e renderojnë. Kështu, "sytë" e fuqishëm — enkoderi vizual — që modelet multimodale moderne e kanë tashmë, mbeten pa u përdorur. Render-in-the-Loop e ristrukturon detyrën si proces vizual hap pas hapi. Pas çdo fragmenti kodi, SVG-ja e pjesshme renderohet në imazh dhe i jepet modelit, kështu që fragmenti tjetër zgjidhet duke parë kanavacën aktuale.

Gjetja e parë është paralajmëruese dhe autorët e raportojnë hapur: thjesht t'ia shtosh këtë cikël një modeli ekzistues nuk funksionon. Kur i ushqyen modele të forta të përgjithshme me renderime të ndërmjetme pa trajnim të posaçëm, cilësia nuk u përmirësua — **u përkeqësua** në të gjitha rastet. Një model që nuk është mësuar t'i përdorë sytë për këtë detyrë nuk di papritur ta bëjë.

Prandaj puna kryesore është trajnimi. Autorët rindërtojnë të dhënat e trajnimit që çdo vizatim të ndahet në shumë hapa të vegjël me kuptim vizual, duke ndarë forma komplekse në pjesë më të thjeshta. Pastaj fine-tune-ojnë një model të hapur me tetë miliardë parametra, të ndërtuar mbi Qwen3-VL, në këto sekuenca hap pas hapi. Këtë e quajnë Visual Self-Feedback. Shtojnë edhe Render-and-Verify gjatë gjenerimit: para se të pranojë një goditje të re, modeli e renderon dhe kontrollon nëse ajo ndryshoi vërtet figurën apo thjesht përsëriti hapin e mëparshëm. Goditjet që nuk shtojnë gjë hidhen dhe, kur asgjë tjetër nuk ndihmon, modeli nxitet të ndalojë. E gjithë kjo përdor një dataset relativisht të vogël — rreth 850,000 shembuj, më pak se gjysma e një rivali dhe një pjesë e vogël e një tjetri.

Çfarë gjetën

- Pas këtij trajnimi, modeli vizaton më mirë se versioni i tij i verbër — sidomos në rastet e dështimit, ku një model i verbër vendos një sy në faqe ose harron një grafik me shtylla të kërkuar dhe nxjerr monitor të përgjithshëm.
- Në benchmark-un standard MMSVGBench, metoda është konkurruese dhe në disa masa pak përpara rivalëve të fortë — përfshirë OmniSVG, të trajnuar me mbi dy herë më shumë të dhëna, dhe InternSVG, me rreth njëzet herë më shumë.
- Diferencat janë të vogla. Në setin e ikonave, për shembull, rezultati kryesor i cilësisë është 127.6 kundrejt 128.8 të rivalit më të mirë, ndërsa rezultati i përputhjes me prompt-in 0.293 kundrejt 0.291 — ndryshime reale e konsistente, por të holla.
- Të dy përbërësit shtesë kanë efekt: hiq trajnimin e posaçëm ose verifikimin gjatë vizatimit dhe rezultatet bien. Verifikimi në veçanti e pengon modelin të ngecë duke rivizatuar të njëjtën gjë.
- Rezultati që autorët theksojnë është efikasiteti — arritja e kësaj performance me shumë më pak të dhëna trajnimi se liderët.

Çfarë nuk provon kjo

- Nuk tregon se ta lejosh modelin "të shohë" është fitore falas. Përkundrazi: vetë punimi tregon se feedback-u vizual **pa** retrajnim i përkeqëson gjërat. Fitimi vjen nga trajnimi për ta përdorur feedback-un, jo nga sytë vetëm.
- Nuk vendos epërsi të madhe ose vendimtare. Në shumicën e matjeve metoda është shumë pranë rivalëve; "mund një model me 20× më shumë të dhëna" është e vërtetë, por me diferenca të vogla në proxy metrics dhe në një benchmark.
- Nuk demonstron aftësi artistike të përgjithshme. Është model kërkimor 8B që vizaton ikona dhe ilustrime të thjeshta në rezolucion të vogël fiks 224×224, jo dizajner universal.
- Krahasimi me një model të madh të përgjithshëm (GPT-5) **nuk** është apples-to-apples: ai model nuk është ndërtuar apo akorduar për këtë detyrë të ngushtë të vizatimit me kod.
- Nuk vjen pa kosto. Renderimi dhe rileximi pas çdo hapi e ngadalëson gjenerimin krahasuar me nxjerrjen e gjithë kodit me një kalim të verbër.

Sa e fortë është prova?

- **E fortë** si krahasim i kontrolluar në kushtet e veta. Ablation-et janë të pastra: hiq trajnimin ose verifikimin dhe rezultatet bien, pra të dy përbërësit bëjnë punën që pretendohet.
- **E ndershme për surprizën e vet.** Fakti që feedback-u naiv vizual *dëmton* raportohet hapur dhe është ndoshta pjesa më interesante: më shumë input nuk do të thotë automatikisht më mirë.
- **Më e hollë si pretendim leaderboard-i.** Fitoret ndaj rivalëve me më shumë të dhëna janë të vogla dhe jetojnë në një benchmark të vetëm, i ndërtuar nga autorët e njërit prej rivalëve. Diferencat e vogla në proxy metrics sugjerojnë diçka, por nuk e mbyllin çështjen.

- **E patestuar në shkallë të madhe dhe në përdorim real.** Këtu kemi ikona dhe ilustrime të thjeshta në rezolucion të ulët. Nëse ideja vlen njësoj për dizajn kompleks, me rezolucion të lartë ose punë reale, mbetet për të ardhmen.

Pse ka rëndësi

Ideja qendrore është pothuajse tepër e thjeshtë dhe shkon përtej vizatimit. Nëse një program do të gjenerojë diçka duke shkruar kod — faqe web, grafik, diagram, skenë 3D — mund ta shkruajë të gjithë verbërisht dhe të shpresojë, ose ta renderojë gjatë rrugës dhe të korrigjojë kursin. Për një njeri kjo është instinktive; ne i hedhim vazhdimisht sytë faqes. Për këto modele është një hap çuditërisht i ri.

Ajo që e bën punimin të vlefshëm është ylli i vogël pranë idesë. T'i japësh një modeli mundësi për të parë nuk është e njëjta gjë me ta mësosh të shikojë. "Sytë" duhen trajnuar; vetëm atëherë cikli jep rezultat — dhe edhe atëherë përfitimi është real, por i matur: një vizatues më i qëndrueshëm, jo një lloj tjetër artisti. Për këdo që ndërton mjete që kthejnë përshkrimin në grafikë të redaktueshme, mësimi praktik është i vlefshëm. Fitimi këtu erdhi nga trajnim më i zgjuar dhe më ekonomik, jo nga më shumë të dhëna.

Përmbledhje e shkurtër

Modelet gjuhësore që gjenerojnë grafikë vektoriale — kodi i redaktueshëm dhe i shkallëzueshëm pas shumicës së ikonave dhe logove web — tradicionalisht e bëjnë "verbërisht", duke nxjerrë komandat pa renderuar rezultatin. Ekipi i Guotao Liang propozon Render-in-the-Loop: rendero vizatimin e papërfunduar pas çdo hapi dhe ktheja modelit, që goditjen tjetër ta bëjë duke parë kanavacën. Gjetja qendrore dhe e ndershme është se ta bësh këtë thjesht te një model ekzistues e bën **më keq**; përfitimi shfaqet vetëm pasi modeli ritrajnohet për ta përdorur feedback-un vizual, i ndihmuar nga një kontroll që heq goditjet që nuk ndryshojnë asgjë. Modeli 8B i ritrajnuar barazon ose kalon pak rivalë të trajnuar me deri në njëzet herë më shumë të dhëna në një benchmark standard — rezultat real, por me diferenca të vogla në proxy scores, për ikona dhe ilustrime të thjeshta në rezolucion të ulët. Mesazhi më i vlefshëm është efikasiteti: të mësosh të shohësh mirë punën tënde mund të kompensojë një pjesë të shkallës së të dhënave.

Kontroll pa teprime

Çfarë tregon punimi: Ritajnimi i një modeli grafikash vektoriale për ta renderuar dhe parë vizatimin gjysmë të përfunduar hap pas hapi prodhon rezultate më të mira e më të plota se vizatimi i verbër — konkurrues dhe pak përpara rivalëve me më shumë të dhëna në një benchmark, me më pak trajnim.

Çfarë është e besueshme, por jo e provuar: Që "ta shohësh punën" është përgjithësisht recetë më e

mirë se shkallëzimi i dataset-it; që ideja do të ndihmojë në grafikë komplekse, rezolucion të lartë ose punë reale; që diferencat e vogla në benchmark do t'i vërente një njeri.

Çfarë nuk tregon: Që feedback-u vizual ndihmon vetë (pa ritrajnim dëmton); që epërsia është e madhe; aftësi vizatimi të përgjithshme; ose krahasim të drejtë me modele të përgjithshme si GPT-5.

Kufizimet kryesore: Një benchmark, i ndërtuar pjesërisht nga autorët e një rivali; diferenca të vogla në proxy metrics; model 8B, ikona dhe ilustrime të thjeshta 224×224; gjenerim më i ngadaltë nga cikli render-every-step.

Sa besim duhet të ketë një lexues i përgjithshëm? Të lartë se mbyllja e ciklit — renderimi dhe rileximi gjatë vizatimit — ndihmon realisht kur është trajnuar. Të ulët deri mesatar se ky model i mund bindshëm rivalët; "mund 20× më shumë të dhëna" është rezultat real, por modest dhe në një test të vetëm. Të lartë për idenë që ia vlen të mbahet mend: për kod që vizaton, të shohësh gjatë punës është më mirë se të vizatosh verbërisht.

Burimet

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/