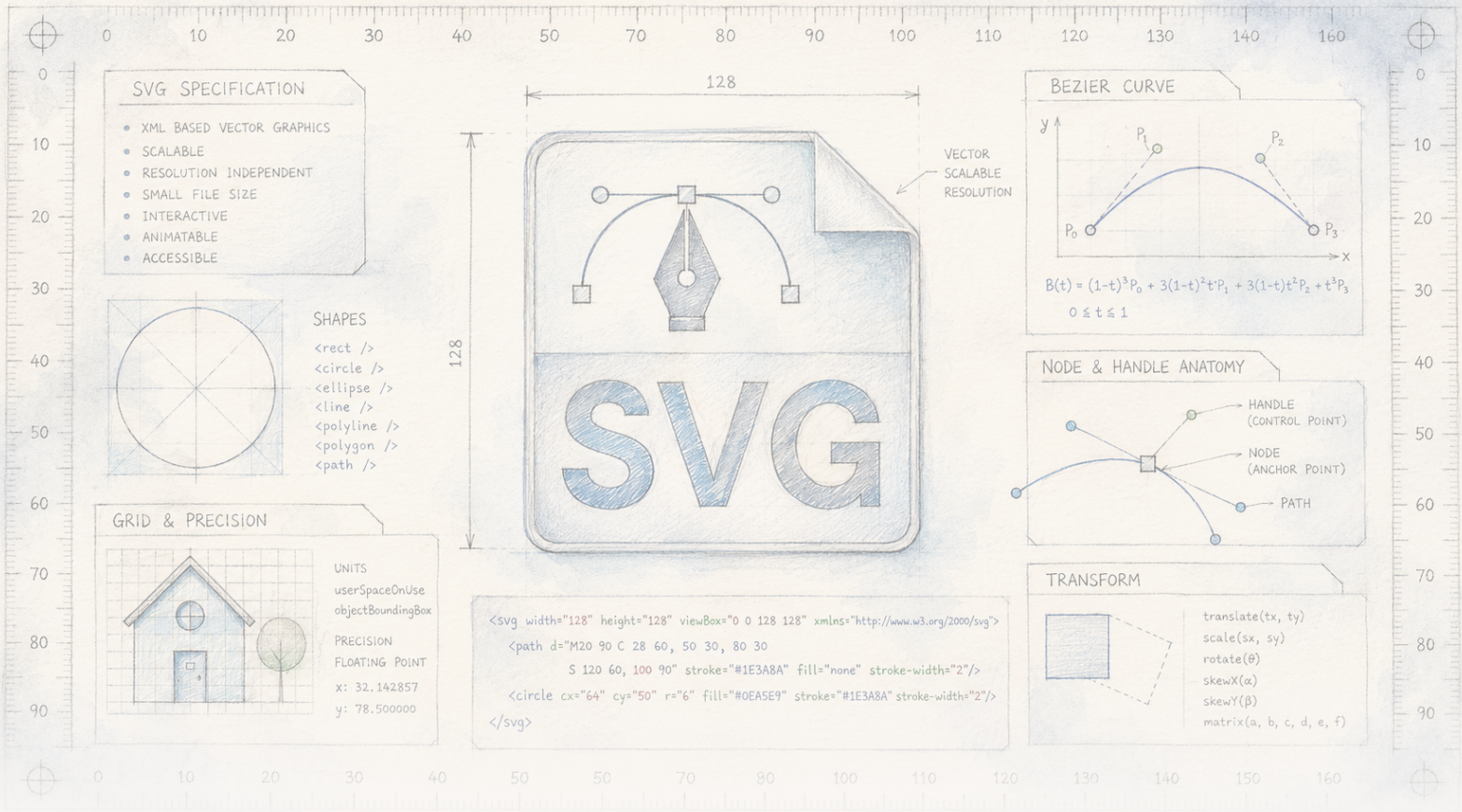


The Clean Paper

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

Joonistava AI õpetamine lehte vaatama

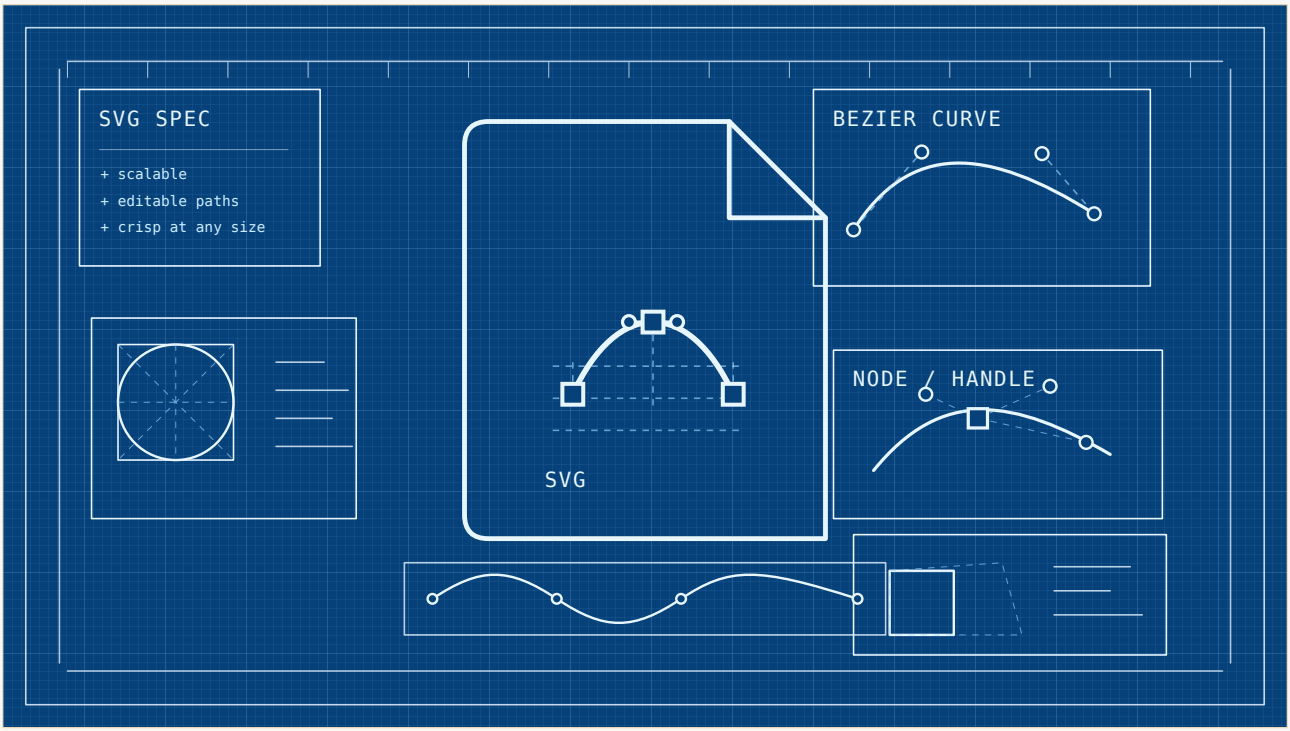
Vektorgraafikat genereerivad keelemudelid töötavad pimesi — kirjutavad joonistuskäsud välja ilma tulemust kunagi nägemata. Uus meetod laseb mudelil jälgida oma lõuendi täitumist tõmme tõmbe haaval ning aus pööre on see, et lihtsalt silmade andmine teeb tulemuse halvemaks: mudel tuleb neid kasutama ümber treenida. Tasuks saadakse mudel, mis jõuab samale tasemele või napilt ette konkurentidest, mida treeniti kuni kakskümmend korda suurema andmehulgaga — päris ja hoolikas ühe benchmark'i tulemus, mitte revolutsioon.

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)

Joonistamine kinnisilmi

Palu ühel tänapäeva AI-mudelil sulle pilt joonistada ja kapoti all toimub üks kahest väga erinevast asjast. Tuttavad pildigeneraatorid — need, mis loovad lausest foto — maalivad piksleid otse ning näevad töö käigus lõuendit. Kuid on ka teine, vaiksem joonistamisviis, kus mudel ei maali üldse. Ta kirjutab juhiseid: *joonista siia ring, sinna joon, täida see kujund sinisega*. Need juhised on kood — seesama Scalable Vector Graphics ehk SVG, millel põhineb suur osa veebis leiduvatest ikoonidest ja logodest — ning sellel on päris eelis: tulemus ei ole fikseeritud pikslivõrk, vaid kujundite kogum, mida saab lõputult skaleerida, ümber värvida ja muuta ilma uduseks minemata.



Originaalne SVG-tehnilise joonise kujundus: pilt ise on muudetav vektorfail, mis koosneb fikseeritud pikslite asemel radadest, ruudustikujoontest, sõlmedest ja juhthoobadest.

Laura Nesso / The Clean Paper Permission on file

Konks on selles, et kuni hiljutise ajani kirjutas mudel seda joonistuskoodi pimesi. Ta väljastas kogu juhiste jada ühe korraga — ring, joon, täide — ilma neid kordagi renderdamata, et näha, mida ta tegi. Kujutage ette näo visandamist kinniste silmadega: võib-olla satub esimene silm enam-vähem õigesse kohta, aga pole võimalik märgata, et teine maandus põsel või et juuste jaoks joonistatud kujund katab nüüd nina. Enam-vähem nii need mudelid töötasidki, mistõttu nad tekitasid sageli koodi, mis oli täiesti korrektne, kuid visuaalselt segadus.

Guotao Liangi juhitud töörühm tegi nüüd peaaegu koomiliselt ilmse asja: lasi mudelil silmad lahti teha. Nende meetod *Render-in-the-Loop* renderdab poolelioleva joonise pärast iga sammu ja annab pildi mudelile tagasi enne järgmise tõmbe tegemist. Päriselt huvitav osa pole see, et see aitab — vaid see, mida oli vaja teha, et see üldse aitaks.

Kuidas joonistust hinnata?

Kui artikkel ütleb, et üks mudel „võidab“ teist, on õiglane küsida: milles ja kuidas mõõdetuna? Genereeritud pildi hindamine on päriselt keeruline — küsimusele „joonista sülearvuti“ pole ühtainsat õiget vastust — seega kasutab valdkond käputäit automaatseid skoore, millest igaüks on inimese hinnangu ebatäiuslik asendus.
Selles artiklis kasutatakse mitut. **FID** võrdleb genereeritud piltide terve kogumi üldist statistilist iseloomu pärispiltidega; väiksem on parem, kuid see kirjeldab kogumit, mitte üht pilti. **CLIP-skoor** küsib eraldi AI-lt, kas pilt vastab prompti sõnadele — kasulik, kuid ainult nii terav kui hindaja ise. **DINO**, **SSIM** ja **LPIPS** võrdlevad rekonstrueeritud pilti sihtpildiga tasanditel, mis ulatuvad toorpikslitest õpitud tunnusteni.
Ükski neist pole tõde. Need on asendusnäitajad ja kipuvad liikuma väikeste sammudega. Kui loed, et üks mudel sai 127,6 ja teine 128,8, on see soovitud suunas päris erinevus — kuid väike nihe, mitte maalihe, ning nihe arvus, mis ainult kaudselt peegeldab seda, mida silm ütleks. Seda tasub meeles pidada, kui pealkiri kuulutab „võidab mudelit, mida treeniti kakskümmend korda rohkemate andmetega“.

Mida autorid tegid

Probleem, mida autorid püüdsid parandada, ongi pime joonistamine. Senised mudelid käsitlevad SVG kirjutamist puhta tekstiülesandena: ennusta senise koodi põhjal järgmine koodijupp ja ära kunagi renderda tulemust. Nii jäävad kasutamata võimsad „silmad“ — nägemiskodeerija — mis tänapäeva multimodaalsetel mudelitel juba olemas on. Render-in-the-Loop korraldab ülesande ümber samm-sammuliseks visuaalseks protsessiks. Pärast iga joonistuskoodijuppi renderdatakse osaline SVG pildiks ja söödetakse mudelile tagasi, nii et järgmine fragment valitakse tegelikult senist lõuendit vaadates.

Esimene leid on hoiatav ning autorite kiituseks öeldakse see otse välja: selle tsükli lihtsalt olemasoleva valmismudeli külge kinnitamine ei tööta. Kui nad andsid tugevatele üldmudelitele vahepealseid renderdusi ilma eritreeninguta, kvaliteet ei paranenud — see *halvenes* üle kogu rea. Mudel, mida pole õpetatud sellisel viisil silmi kasutama, ei oska seda äkki iseenesest.

Seega on suurem osa tööst õpetamises. Nad ehitavad treeningandmed ümber nii, et iga joonistus jagatakse paljudeks väikesteks visuaalselt tähenduslikeks sammudeks — keerulised kujundid lõhutakse lihtsamateks osadeks, et igal etapil oleks midagi uut näha — ning seejärel peenhäälestatakse nende jadasid kasutades kaheksa miljardi parameetriga avatud mudelit (Qwen3-VL-i põhjal). Nad nimetavad seda Visual Self-Feedback’iks. Joonistamise ajal lisatakse teine mehhanism, Render-and-Verify: enne iga uue tõmbe aktsepteerimist renderdab mudel selle ja kontrollib, kas pilt tegelikult muutus või kordas uus käsk lihtsalt eelmist. Midagi lisamata tõmbed visatakse minema ning kui edasine lisamine enam ei aita, palutakse mudelil lõpetada. Märkimisväärselt töötab kõik see üsna väikese andmestiku peal — umbes 850 000 näidet, vähem kui pool ühe konkurendi ja väike murdosa teise kasutatud mahust.

Mida nad leidsid

- Selliselt treenitud mudel joonistab paremini kui tema pime vaste — eriti nähtavalt ebaõnnestumiste puhul, kus pime mudel paneb silma põsele või jätab palutud tulpdiagrammi tegemata ja väljastab selle asemel üldise monitori.
- Standardsel võrdlustestil MMSVGBench on meetod tugevate konkurentidega võrreldav ja mitme näitaja järgi neist veidi ees — sealhulgas OmniSVG-st, mida treeniti rohkem kui kaks korda suurema andmehulgaga, ja InternSVG-st, mida treeniti ligikaudu kakskümmend korda suurema andmehulgaga.
- Vahed on väikesed. Ikoonide kogumis on näiteks peamine pildikvaliteedi skoor 127,6 võrreldes parima konkurendi 128,8-ga ning prompti sobivuse skoor 0,293 võrreldes 0,291-ga — päris ja järjekindel, kuid napp eelis.
- Mõlemad lisakomponendid õigustavad end: eritreeningu või joonistamisaegse kontrolli väljalülitamisel langevad näitajad mõõdetavalt. Eriti kontrollietapp takistab mudelil takerdumast sama asja üha uuesti joonistama.
- Tulemus, mida autorid ise rõhutavad, on tõhusus — nii kaugele jõutakse palju väiksema treeningandmehulgaga kui liidritel.

Mida see ei tõesta

- See **ei näita**, et mudelile „nägemise“ andmine on tasuta võit. Vastupidi: artikli enda katse näitab, et visuaalne tagasiside *ilma* ümbertreeninguta teeb tulemuse halvemaks. Võit tuleb treeningust, mitte silmadest üksinda.
- See **ei kehtesta suurt ega otsustavat edumaad**. Enamiku skooride järgi on meetod konkurentidega kael-kuklas; „võidab mudelit, mida treeniti 20× suurema andmehulgaga“ on tõsi, kuid väikeste nihetega asendusmõõdikutes ja ühel võrdlustestil.
- See **ei demonstreeri üldist kunstilist võimet**. Tegemist on kaheksa miljardi parameetriga uurimismudeliga, mis joonistab ikoone ja lihtsaid illustratsioone väikesel fikseeritud lahutusel (224×224 pikslit), mitte üldotstarbelise disaineriga.
- Võrdlus suure üldmudeliga (GPT-5) **ei ole üks-ühele õiglane**: see mudel pole ehitatud ega häälestatud kitsale koodijoonistamise ülesandele, seega ütleb siinne võit üldiselt kummagi mudeli kohta vähe.
- See **ei tule tasuta**. Lõuendi renderdamine ja uuesti lugemine igal sammul teeb genereerimise aeglasemaks kui kogu koodi väljastamine ühe pimeda passiga — autorid tunnistavad seda kulu.

Kui tugev on tõendusmaterjal

- **Tugev** seal, kus tegu on kontrollitud võrdlusega töö enda tingimustel. Ablatsioonid on puhtad: eemaldad treeningu või kontrolli ja numbrid langevad — seega teevad mõlemad komponendid tõepoolest seda, mida autorid väidavad.

- **Aus omaenda üllatuse suhtes.** Leid, et naiivne visuaalne tagasiside *kahjustab*, on välja toodud, mitte peidetud, ning see on artikli kõige huvitavam tulemus — kasulik parandus intuitsioonile, et rohkem sisendit on alati parem.
- **Õhuke edetabeliväitena.** Võidud suurema andmehulgaga konkurentide ees on väikesed ja pärinevad ühelt võrdlustestilt, mille lõid ühe konkurendi autorid. Väikesed vahed asendusmõõdikutes ühel testkogumil on vihjavad, mitte lõplikud.
- **Suures skaalas ja pärismaailmas testimata.** Kõik siin on ikoonid ja lihtsad illustratsioonid väikese lahutusega. Kas sama idee töötab keeruka, suure lahutusega või päris disainitöö puhul, jääb tulevikuks — autorid ütlevad seda ka ise.

Miks see oluline on

Artikli keskne idee on peaaegu piinlikult lihtne ja ulatub joonistamisest palju kaugemale. Kui programm hakkab midagi koodi kirjutades genereerima — veebilehte, graafikut, diagrammi, 3D-stseeni — võib ta kirjutada kogu asja pimesi ja loota parimat või renderdada töö käigus ning kurssi parandada. Inimese jaoks on sellise tagasisideahela sulgemine loomulik; me heidame lehele pidevalt pilgu. Nende mudelite jaoks on see üllatavalt hiljutine samm.

Artikli teeb lugemist väärt tärn, mille see ideele lisab. Mudelile nägemise võimaldamine pole sama mis selle õpetamine vaatama. Silmi tuleb treenida ja alles siis tasub tsükkel ära — ning ka siis on võit päris, kuid mõõdukas: kindlama käega joonistaja, mitte uut tüüpi kunstnik. Kõigile, kes ehitavad tööriistu, mis muudavad kirjelduse redigeeritavaks graafikaks — näiteks rakenduste ikoonideks ja illustratsioonideks — on vaikne praktiline õppetund kõige kasulikum. Võit tuli odavamast ja nutikamast treeningust, mitte lihtsalt rohkematest andmetest. See on väärtuslikum teadmine kui järjekordne punkt edetabelis.

Lühidalt

Vektorgraafikat — enamiku veebiikoonide ja logode taga olevat redigeeritavat ja skaleeritavat koodi — genereerivad keelemudelid on traditsiooniliselt töötanud „pimesi“, kirjutades kõik joonistuskäsud välja ilma tulemust kunagi renderdamata. Guotao Liangi juhitud töörühm pakub Render-in-the-Loop meetodit: pärast iga sammu renderdatakse pooleliolev joonis ja antakse tagasi mudelile, nii et järgmine tõmme tehakse lõuendit vaadates. Nende keskne ja aus leid on, et selle lihtsalt olemasoleva mudeli külge lisamine teeb mudeli *halvemaks*; kasu ilmub alles pärast seda, kui mudel on visuaalset tagasisidet kasutama ümber treenitud, abiks joonistamisaegne kontroll, mis viskab ära pildi muutmata jätvad tõmbed. Ümbertreenitud kaheksa miljardi parameetriga mudel jõuab standardsel võrdlustestil samale tasemele või veidi ette konkurentidest, mida treeniti kuni kakskümmend korda suurema andmehulgaga — päris tulemus, kuid väikeste vahedega asendusmõõdikutes, ikonide ja lihtsate väikese lahutusega illustratsioonide puhul. Autorite rõhutatud ja säilitamist väärt järeldus puudutab tõhusust: oma töö hea nägemine võib osaliselt asendada paljast andmemastaapi.

Kaine hinnang

Mida artikkel näitab: Vektorgraafikamudeli ümbertreenimine nii, et ta renderdab ja vaatab sammsammult oma pooleliolevat joonist, annab pimedast joonistamisest paremaid ja täielikumaid tulemusi — ühe standardse võrdlustesti järgi konkurentsivõimelisi ja veidi paremaid kui suurema andmehulgaga konkurentidel, kasutades vähem treeningandmeid.

Mis on usutav, kuid tõestamata: Et „vaata oma tööd“ on üldiselt parem retsept kui andmete suurendamine; et sama idee aitab keeruka, suure lahutusega või pärismaailma graafika puhul; et väikesed võrdlustesti vahed vastavad erinevusele, mida inimene tegelikult märkaks.

Mida see ei näita: Et visuaalne tagasiside aitab iseenesest (ilma ümbertreeninguta kahjustab see tulemust); et eelis konkurentide ees on suur või otsustav; üldotstarbelist joonistusvõimet; õiglast üksühele võrdlust üldmudelitega nagu GPT-5, mis pole selle ülesande jaoks ehitatud.

Peamised piirangud: Üks võrdlustest, mille loomises osales ühe konkurendi autorirühm; väikesed vahed asendusmõõdikutes; 8B mudel, ikoonid ja lihtsad illustratsioonid 224×224 lahutusel; aeglasem genereerimine, sest iga samm renderdatakse.

Kui kindel võiks tavalugeja olla? Väga kindel, et ahela sulgemine — töö renderdamine ja uuesti vaatamine joonistamise käigus — tõepoolest aitab ning et seda tuleb sisse treenida, mitte lihtsalt sisse lülitada. Madal kuni keskmine kindlus, et just see mudel konkurente otsustavalt võidab; „võidab 20× rohkemate andmetega mudelit“ on päris, kuid tagasihoidlik ühe võrdlustesti tulemus. Väga kindel tasub olla ühes idees: joonistava koodi puhul on töö käigus vaatamine parem kui pimesi joonistamine.

Allikad

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/