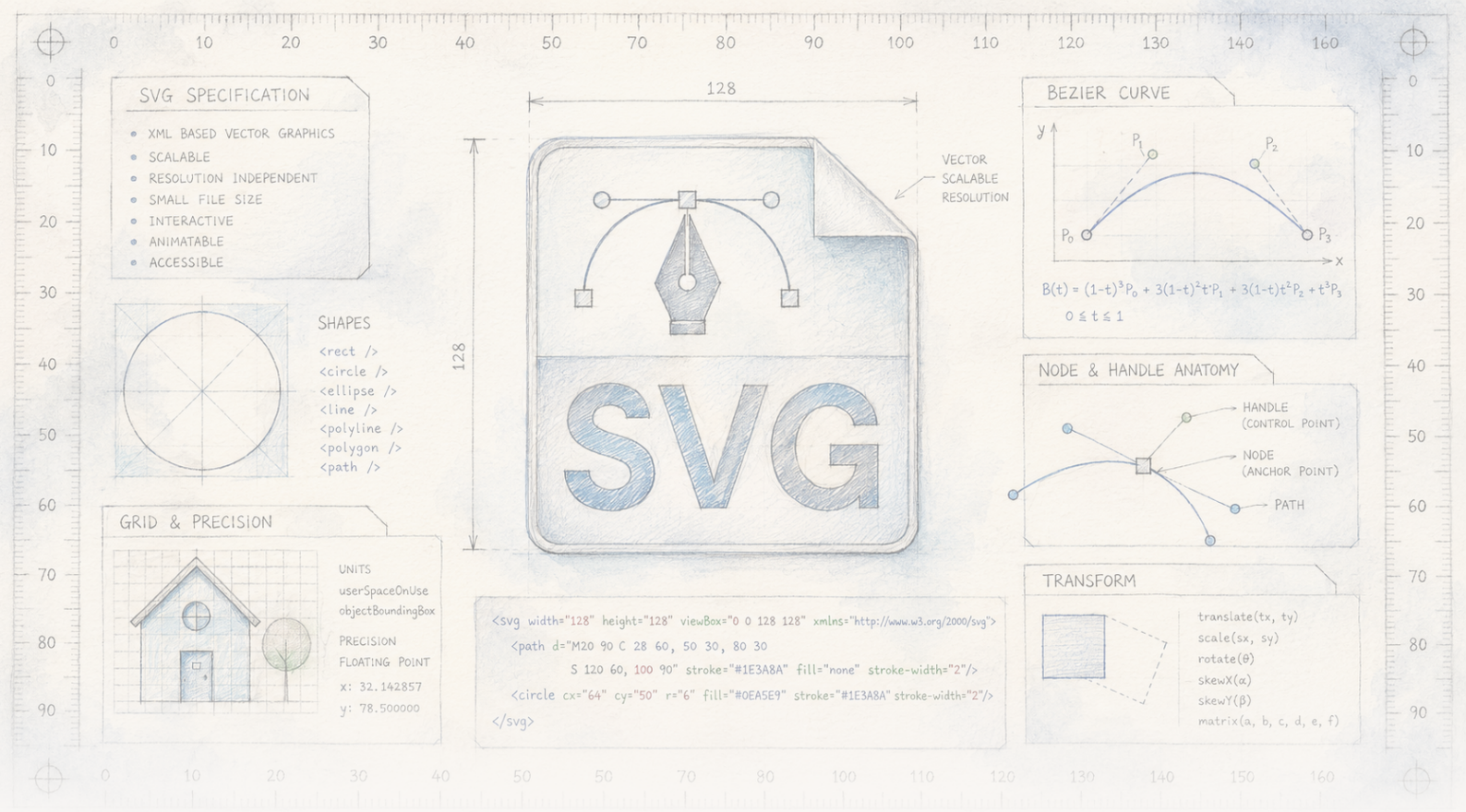



The Clean Paper

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

# Å lære en KI som tegner å se på arket

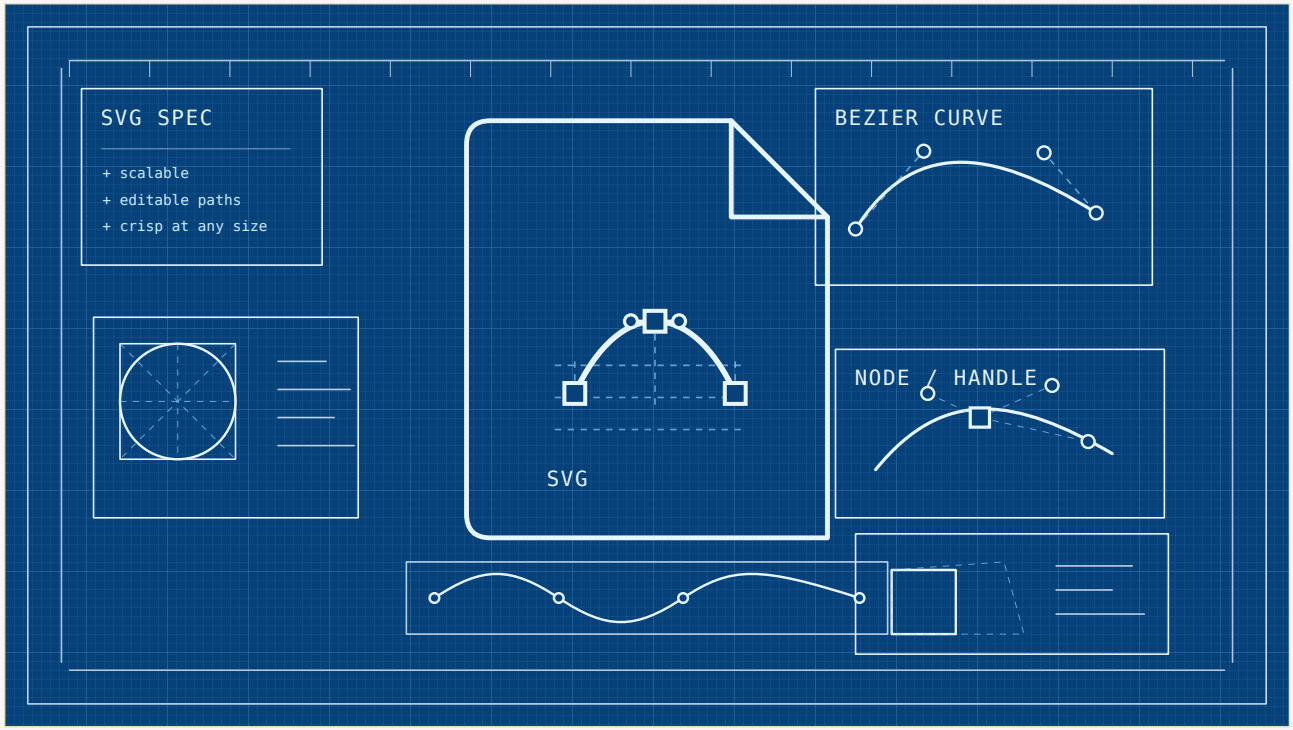
*Språkmodeller som genererer vektorgrafikk, gjør det i blinde — de skriver tegnekommandoene uten noen gang å se resultatet. En ny metode lar modellen se sitt eget lerret fylles, strek for strek, og den ærlige vendingen er at det blir dårligere bare å gi den øyne: Den må trenes på nytt for å bruke dem. Resultatet er en modell som matcher eller så vidt slår rivaler trent på opptil tjue ganger mer data — et reelt og nøkternt resultat på én standardtest, ikke en revolusjon.*

Written by Vera Rubin · Cross-checked by Michele Renda · Edited by Michele Renda · Figures by Laura Nesso · 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)

# Å tegne med øynene lukket

Be en av dagens KI-modeller tegne et bilde for deg, så skjer én av to svært forskjellige ting under panseret. De velkjente bildegeneratorene — de som tryller frem et fotografi fra en setning — maler piksler direkte, og de kan se lerretet mens de arbeider. Men det finnes en annen, stillere form for tegning der modellen ikke maler i det hele tatt. Den skriver instruksjoner: *tegn en sirkel her, en linje der, fyll denne formen med blått*. Instruksjonene er kode — samme Scalable Vector Graphics, eller SVG, som ligger bak de fleste ikoner og logoer på nettet — og fordelen er reell: Resultatet er ikke et fast rutenett av piksler, men et sett med former du kan skalere, fargelegge på nytt og redigere i det uendelige uten at det blir uskarpt.




Original teknisk SVG-tegning: Bildet er selv en redigerbar vektorfil, bygget opp av baner, rutenettlinjer, noder og håndtak i stedet for faste piksler.

Laura Nesso / The Clean Paper Permission on file

Haken er at en modell som skrev denne tegnekoden, inntil nylig gjorde det i blinde. Den sendte ut hele sekvensen av instruksjoner i én omgang — sirkel, linje, fyll — uten noen gang å rendre dem for å se hva den hadde laget. Se for deg at du skisserer et ansikt med øynene lukket: Du får kanskje øynene omtrent der øyne skal være, men du har ingen mulighet til å oppdage at det andre havnet på kinnet, eller at formen du tegnet som hår, nå ligger over nesen. Det var mer eller mindre slik disse modellene fungerte, og derfor produserte de så ofte kode som var helt gyldig, men visuelt rotete.

Et team ledet av Guotao Liang har nå gjort det nesten komisk opplagte: De lot modellen åpne øynene. Metoden deres, *Render-in-the-Loop*, rendrer den halvferdige tegningen etter hvert trinn og gir bildet tilbake til modellen før den tegner neste strek. Det virkelig interessante er ikke at dette hjelper — men hva de måtte gjøre for at det i det hele tatt skulle hjelpe.

**Hvordan vurderer man en tegning?**

Når en forskningsartikkel sier at modellen dens «slår» en annen, er det rimelig å spørre: Slår den på hva, målt hvordan? Det er genuint vanskelig å vurdere et generert bilde — det finnes ikke ett riktig svar på «tegn en bærbar datamaskin» — så fagfeltet støtter seg på en håndfull automatiske mål, som alle er ufullkomne erstatninger for at et menneske ser på resultatet.
Noen av dem brukes i denne artikkelen. **FID** sammenligner det overordnede statistiske preget i en hel gruppe genererte bilder med virkelige bilder; lavere er bedre, men målet beskriver gruppen, ikke ett enkelt bilde. **CLIP-poeng** spør en separat KI om et bilde samsvarer med ordene i prompten — nyttig, men ikke skarpere enn den dommeren. **DINO**, **SSIM** og **LPIPS** sammenligner et rekonstruert bilde med et målbilde på nivåer som spenner fra rå piksler til innlærte egenskaper.
Ingen av disse målene er sannheten. De er indirekte mål, og de beveger seg gjerne i små trinn. Når du leser at én modell får 127.6 der en annen får 128.8, er det en reell forskjell i den tilsiktede retningen — men det er et lite dytt, ikke et jordskred, og et dytt i et tall som bare løselig følger det øynene dine ville sagt. Det er verdt å ha i bakhodet når overskriften lyder «slår en modell trent på tjue ganger så mye data».

# Hva forfatterne gjorde

Problemet forfatterne ville løse, er selve tegningen i blinde. Eksisterende modeller behandler det å skrive SVG som en ren tekstoppgave: forutsi neste kodebit ut fra koden så langt, og aldri rendre den. Dermed står de kraftige «øynene» — synskoderen — som moderne multimodale modeller allerede har, ubrukt. Render-in-the-Loop gjør oppgaven om til en trinnvis visuell prosess. Etter hver bit med tegnekode rendres den delvise SVG-en til et bilde og mates tilbake til modellen som et bilde, slik at neste bit velges mens modellen faktisk ser på lerretet slik det er så langt.

Det første funnet deres maner til forsiktighet, og til deres ære rapporterer de det tydelig: Det fungerer ikke bare å koble denne sløyfen til en eksisterende standardmodell. Da de ga mellomliggende rendrede bilder til sterke generelle modeller uten noen spesiell trening, ble ikke kvaliteten bedre — den ble *dårligere* over hele linjen. En modell som aldri har lært å bruke øynene til dette, vet ikke plutselig hvordan den skal gjøre det.

Det meste av arbeidet ligger derfor i opplæringen. De bygger om treningsdataene slik at hver tegning deles inn i mange små, visuelt meningsfulle trinn — komplekse former deles opp i enklere biter, slik at det er noe nytt å se på hvert stadium — og finjusterer deretter en åpen modell med åtte milliarder parametere (bygget på Qwen3-VL) på disse trinnvise sekvensene. De kaller dette Visual Self-Feedback. De legger til en annen mekanisme under tegningen, Render-and-Verify: Før modellen godtar hver nye strek, rendrer den streken og sjekker om den faktisk endret bildet, eller bare gjentok den forrige. Streker som ikke tilfører noe, forkastes, og når ingenting mer hjelper, får modellen beskjed om å stoppe. Det er verdt å merke seg at alt dette kjøres på et ganske lite datasett — rundt 850,000 eksempler, mindre enn halvparten av det én konkurrent brukte og en liten brøkdel av en annens.

# Hva de fant

- Når modellen trenes på denne måten, tegner den bedre enn sin blinde motpart — mest synlig i feiltilfellene, der en blind modell plasserer et øye på et kinn eller hopper over et etterspurt stolpediagram og i stedet lager en generisk skjerm.
- På standardtesten (MMSVGBench) er metoden konkurransedyktig med sterke rivaler og ligger på flere mål litt foran dem — blant annet OmniSVG, som er trent på mer enn dobbelt så mye data, og InternSVG, som er trent på rundt tjue ganger så mye.
- Marginene er små. På ikonsettet er for eksempel hovedmålet for bildekvalitet 127.6 mot 128.8 for den beste rivalen, og poengsummen for samsvar med prompten er 0.293 mot 0.291 — reelle og konsekvente forskjeller, men knappe.
- Begge tilleggene forsvarer sin plass: Slå av spesialtreningen eller verifiseringen under tegningen, så faller tallene målbart. Verifiseringstrinnet hindrer særlig modellen i å bli sittende fast og tegne det samme på nytt om og om igjen.
- Resultatet forfatterne selv fremhever, er effektiviteten — å komme så langt med langt mindre treningsdata enn lederne brukte.

# Hva dette ikke beviser

- Det viser **ikke** at det å la en modell «se» er en gratis gevinst. Tvert imot: Artikkelens eget eksperiment viser at visuell tilbakemelding *uten* ny trening gjør resultatet dårligere. Gevinsten kommer fra den nye treningen, ikke fra øynene alene.
- Det fastslår **ikke** en stor eller avgjørende ledelse. På de fleste mål ligger metoden jevnt med rivalene; «slår en modell trent på 20× så mye data» er sant, men det dreier seg om små utslag i indirekte mål på én standardtest.
- Det demonstrerer **ikke** generell kunstnerisk evne. Dette er en forskningsmodell med åtte milliarder parametere som tegner ikoner og enkle illustrasjoner i en liten, fast oppløsning (224×224 piksler), ikke en designer for allmenn bruk.
- Sammenligningen med en stor generell modell (GPT-5) skjer **ikke** på like vilkår: Den modellen er ikke bygget eller finjustert for denne snevre oppgaven med å tegne i kode, så det å slå den her sier lite om noen av modellene generelt.
- Det er **ikke** gratis. Å rendre og lese lerretet på nytt for hvert trinn gjør genereringen langsommere enn å sende ut koden i én blind omgang — en kostnad forfatterne erkjenner.

# Hvor sterke er bevisene?

- **Solide** når det gjelder en kontrollert sammenligning på metodens egne premisser. Ablasjonsforsøkene er tydelige: Fjern treningen eller verifiseringen, så faller tallene — de to bestanddelene gjør altså virkelig det arbeidet forfatterne hevder.
- **Ærlig om sin egen overraskelse.** Funnet om at naiv visuell tilbakemelding *skader*, blir rapportert,

ikke gjemt bort, og det er det mest interessante i artikkelen — en nyttig korreksjon av intuisjonen om at mer informasjon alltid er bedre.

- **Tynne når det gjelder påstanden om resultatlisten.** Seirene over rivaler med større datamengder er små og finnes på én enkelt standardtest, som ble bygget av forfatterne bak en av disse rivalene. Små marginer i indirekte mål på ett testsett er antydende, ikke avgjørende.
- **Utestet i stor skala og i praksis.** Alt her handler om ikoner og enkle illustrasjoner i lav oppløsning. Om den samme ideen holder for komplekst designarbeid i høy oppløsning eller i virkelige anvendelser, må undersøkes i fremtidig arbeid — forfatterne sier det selv.

# Hvorfor det betyr noe

Ideen i sentrum av denne forskningsartikkelen er nesten pinlig enkel, og den strekker seg langt utover tegning. Hvis et program skal generere noe ved å skrive kode — en nettside, en graf, et diagram, en 3D-scene — kan det enten skrive alt i blinde og håpe på det beste, eller rendre underveis og korrigere kursen. For et menneske er det helt naturlig å lukke denne sløyfen; vi kaster stadig blikk på siden. For disse modellene er det et overraskende nytt grep.

Det som gjør artikkelen verdt å lese, er forbeholdet den legger til. Å gi en modell mulighet til å se er ikke det samme som å lære den å se etter. Øynene må trenes, og først da gir sløyfen uttelling — og selv da er gevinsten reell, men nøktern: en stødigere tegner, ikke en annen type kunstner. For alle som bygger verktøy som gjør en beskrivelse om til redigerbar grafikk — den typen som ender opp bak ikonene og illustrasjonene i en app — er den stillferdige, praktiske lærdommen den nyttige. Gevinsten her kom fra billigere og smartere trening, ikke fra mer data. Det er en bedre ting å lære enn enda et poeng på en resultatliste.

# Kort oppsummering

Språkmodeller som genererer vektorgrafikk — den redigerbare, skalerbare koden bak de fleste ikoner og logoer på nettet — har tradisjonelt gjort det «i blinde» ved å skrive ut alle tegnekommandoene uten noen gang å rendre dem for å se resultatet. Et team ledet av Guotao Liang foreslår Render-in-the-Loop: rendre den halvferdige tegningen etter hvert trinn og mate den tilbake, slik at modellen tegner neste strek mens den ser på lerretet. Det sentrale og ærlige funnet deres er at det å bare gjøre dette med en eksisterende modell gjør den *dårligere*; gevinsten kommer først etter at modellen trenes på nytt til å bruke den visuelle tilbakemeldingen, hjulpet av en kontroll under tegningen som forkaster streker som ikke endrer noe. Den nytrente modellen med åtte milliarder parametere matcher eller slår med liten margin rivaler som er trent på opptil tjue ganger mer data på en standardtest — et reelt resultat, men med små marginer på indirekte mål, for ikoner og enkle illustrasjoner i lav oppløsning. Konklusjonen forfatterne fremhever, og som er verdt å ta med seg, handler om effektivitet: Å se sitt eget arbeid ordentlig kan veie opp for ren datamengde.

# No-BS-sjekk

**Hva forskningsartikkelen viser:** Å trene en vektorgrafikkmodell på nytt til å rendre og se på sin egen halvferdige tegning, trinn for trinn, gir bedre og mer fullstendige resultater enn å tegne i blinde — konkurransedyktig med og litt bedre enn rivaler med større datamengder på én standardtest, med mindre treningsdata.

**Hva som er plausibelt, men ikke bevist:** At «det å se sitt eget arbeid» generelt er en bedre oppskrift enn å skalere opp datamengden; at den samme ideen vil hjelpe med kompleks grafikk i høy oppløsning eller i virkelige anvendelser; at de små marginene på standardtesten tilsvarer en forskjell et menneske faktisk ville legge merke til.

**Hva den ikke viser:** At visuell tilbakemelding hjelper i seg selv (uten ny trening skader den); at forspranget på rivalene er stort eller avgjørende; generell tegneevne; en rettferdig direkte sammenligning med generelle modeller som GPT-5, som ikke er bygget for denne oppgaven.

**Viktigste begrensninger:** Én standardtest, delvis bygget av forfatterne bak en rival; små marginer i indirekte mål; en 8B-modell, ikoner og enkle illustrasjoner i 224×224; langsommere generering på grunn av sløyfen som rendrer etter hvert trinn.

**Hvor stor tillit bør en vanlig leser ha?** Høy tillit til at det å lukke sløyfen — rendre og lese på nytt mens man tegner — virkelig hjelper, og at det må trenes inn, ikke bare slås på. Lav til moderat tillit til at akkurat denne modellen avgjørende slår rivalene sine; se på formuleringen «slår 20× så mye data» som et reelt, men beskjedent resultat fra én standardtest. Høy tillit til den ene ideen det er verdt å huske: For kode som tegner, er det bedre å se underveis enn å tegne i blinde.

## Sources

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/