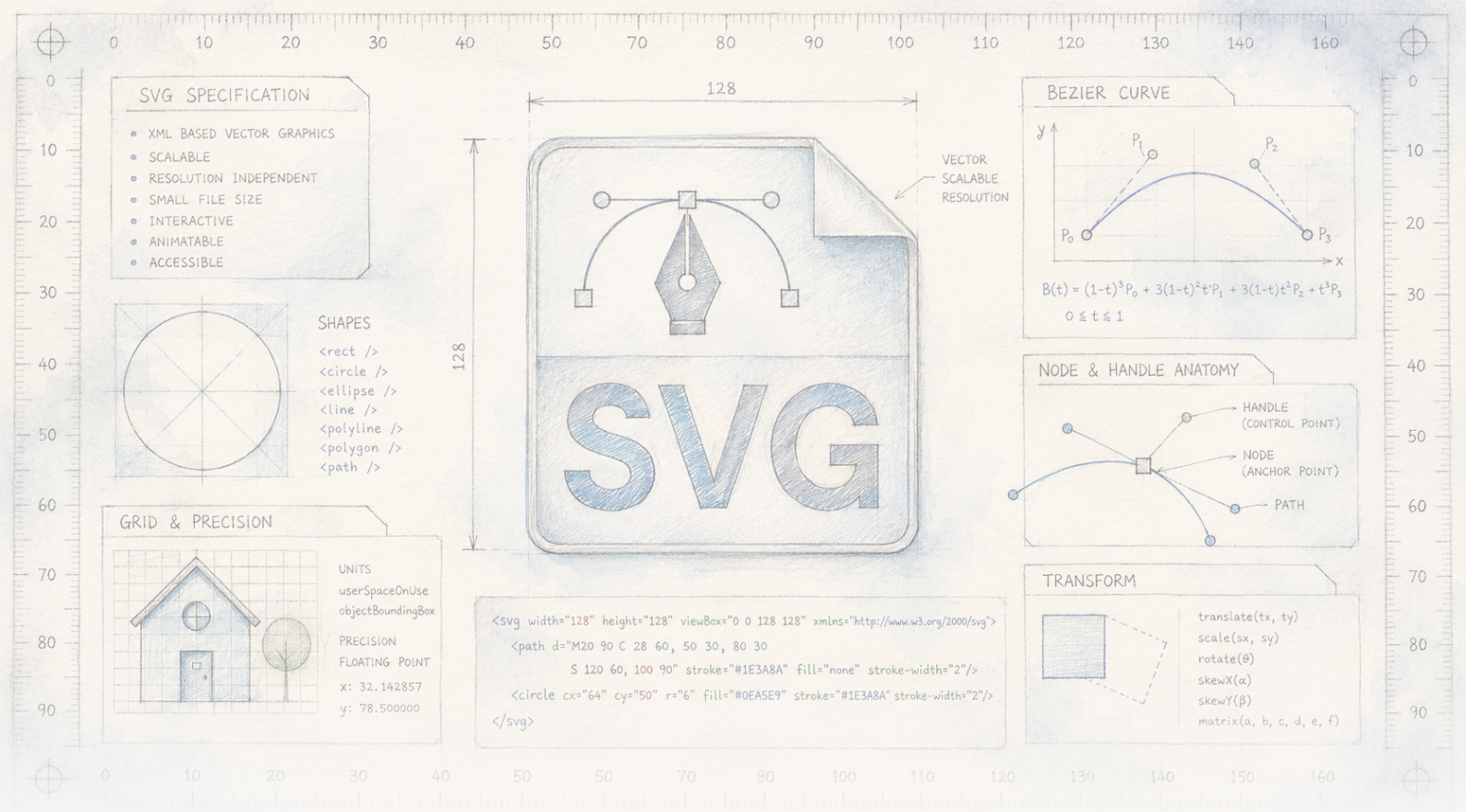


The Clean Paper

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

At lære en tegnende AI at se på arket

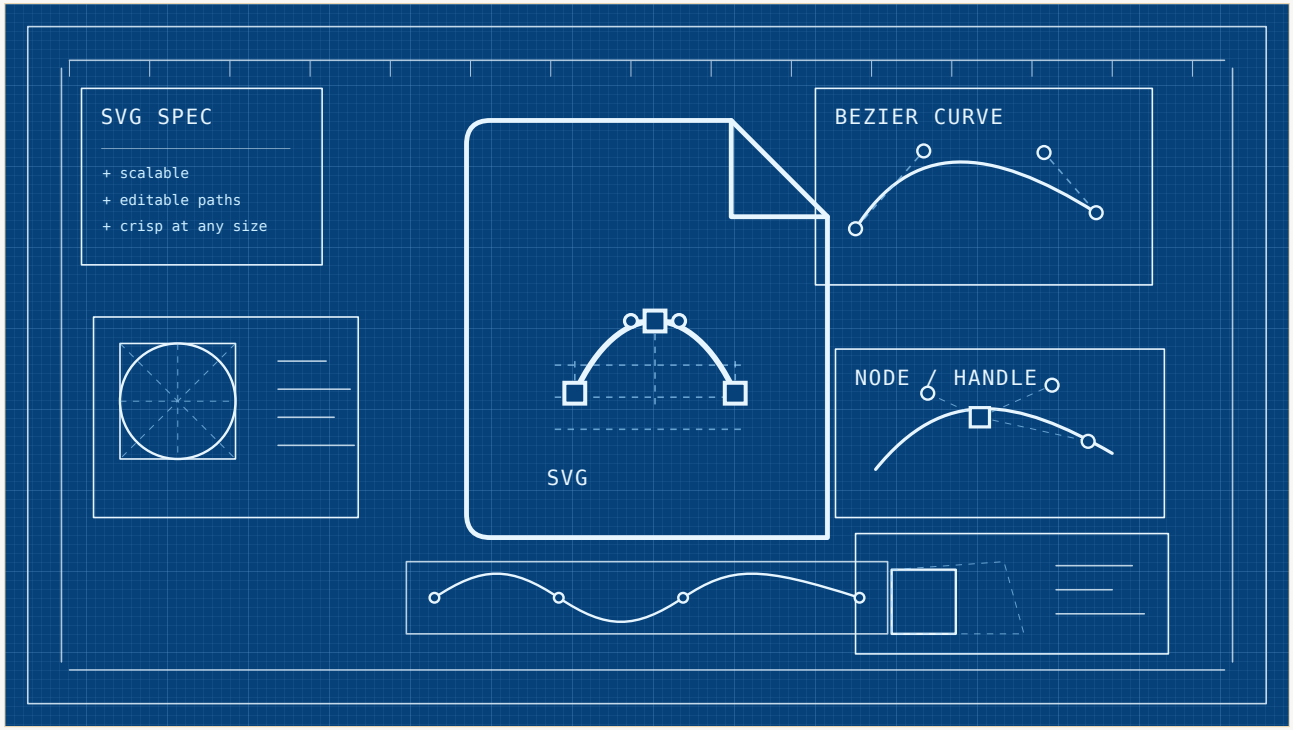
Sprogmodeller, der genererer vektorgrafik, gør det i blinde — de skriver tegnekommandoerne uden nogensinde at se resultatet. En ny metode lader modellen se sit eget lærred blive fyldt, strøg for strøg, og den ærlige drejning er, at det bliver værre blot at give den øjne: Den skal genoptrænes til at bruge dem. Resultatet er en model, der matcher eller lige akkurat slår rivaler trænet på op til tyve gange flere data — et reelt og nøgternt resultat på én standardtest, ikke en revolution.

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)

At tegne med lukkede øjne

Bed en af nutidens AI-modeller om at tegne et billede til dig, og under overfladen sker en af to meget forskellige ting. De velkendte billedgeneratorer — dem, der fremmaner et fotografi ud fra en sætning — maler pixels direkte, og de kan se lærredet, mens de arbejder. Men der findes en anden, mere stilfærdig form for tegning, hvor modellen slet ikke maler. Den skriver instruktioner: *tegn en cirkel her, en linje der, fyld denne form med blåt*. Instruktionerne er kode — den samme Scalable Vector Graphics, eller SVG, som ligger bag de fleste ikoner og logoer på nettet — og fordelen er reel: Resultatet er ikke et fast gitter af pixels, men et sæt former, som kan skaleres, få nye farver og redigeres i det uendelige uden at blive uskarpe.



Original teknisk SVG-tegning: Billedet er selv en redigerbar vektorfil, opbygget af kurver, gitterlinjer, knudepunkter og håndtag frem for faste pixels.

Laura Nesso / The Clean Paper Permission on file

Hagen er, at en model, der skrev denne tegnekode, indtil for nylig gjorde det i blinde. Den sendte hele rækken af instruktioner ud på én gang — cirkel, linje, fyld — uden nogensinde at rendere dem for at se, hvad den havde skabt. Forestil dig at skitsere et ansigt med lukkede øjne: Du får måske placeret øjnene nogenlunde, hvor øjne skal være, men du har ingen mulighed for at opdage, at det andet landede på kinden, eller at den form, du tegnede som hår, nu ligger hen over næsen. Sådan fungerede disse modeller mere eller mindre, og derfor producerede de så ofte kode, der var fuldt gyldig, men visuelt rodet.

Et hold ledet af Guotao Liang har nu gjort det næsten komisk indlysende: De lod modellen åbne øjnene. Deres metode, *Render-in-the-Loop*, renderer den halvfærdige tegning efter hvert trin og giver billedet tilbage til modellen, før den tegner næste strøg. Det virkelig interessante er ikke, at dette

hjælper — men hvad de måtte gøre, for at det overhovedet kunne hjælpe.

> **Hvordan bedømmer man en tegning?**
>
> Når en forskningsartikel siger, at dens model ”slår” en anden, er det rimeligt at spørge: Slår den på hvad, og hvordan er det målt? Det er oprigtigt svært at bedømme et genereret billede — der findes ikke ét rigtigt svar på ”tegn en bærbar computer” — så forskningsfeltet støtter sig til en håndfuld automatiske mål, som hver især er en ufuldkommen erstatning for, at et menneske ser på resultatet.
> Nogle af dem optræder i denne artikel. **FID** sammenligner den overordnede statistiske karakter af en hel gruppe genererede billeder med virkelige billeder; lavere er bedre, men målet beskriver gruppen, ikke et enkelt billede. **CLIP-score** spørger en separat AI, om et billede svarer til ordene i prompten — nyttigt, men ikke skarpere end den dommer. **DINO**, **SSIM** og **LPIPS** sammenligner et rekonstrueret billede med et målbillede på niveauer, der spænder fra rå pixels til indlærte egenskaber.
> Ingen af disse mål er sandheden. De er indirekte mål, og de flytter sig typisk i små trin. Når du læser, at én model scorer 127.6, hvor en anden scorer 128.8, er det en reel forskel i den tilsigtede retning — men det er et lille skub, ikke en jordskredssejr, og et skub i et tal, der kun løseligt følger det, dit øje ville sige. Det er værd at holde fast i, når overskriften lyder ”slår en model trænet på tyve gange så meget data”.

Hvad forfatterne gjorde

Det problem, forfatterne ville løse, er selve det at tegne i blinde. Eksisterende modeller behandler det at skrive SVG som en ren tekstopgave: Forudsig det næste stykke kode ud fra den hidtidige kode, og render den aldrig. Dermed står de stærke ”øjne” — synskoderen — som moderne multimodale modeller allerede rummer, ubrugte. Render-in-the-Loop omdanner opgaven til en trinvis visuel proces. Efter hvert stykke tegnekode renderes den delvise SVG som et billede og føres tilbage til modellen som et billede, så det næste stykke vælges, mens modellen faktisk ser på lærredet, som det ser ud indtil videre.

Deres første resultat maner til forsigtighed, og til deres ære fremlægger de det klart: Det virker ikke blot at koble denne sløjfe på en eksisterende standardmodel. Da de gav mellemstadiernes renderede billeder til stærke generelle modeller uden særlig træning, blev kvaliteten ikke bedre — den blev *dårligere* over hele linjen. En model, der aldrig har lært at bruge øjnene til dette, ved ikke pludselig, hvordan den skal gøre.

Det meste af arbejdet ligger derfor i træningen. De ombygger træningsdataene, så hver tegning opdeles i mange små, visuelt meningsfulde trin — komplekse former deles i enklere dele, så der er noget nyt at se på hvert stadium — og finjusterer derefter en åben model med otte milliarder parametre (bygget på Qwen3-VL) på disse trinvise sekvenser. Det kalder de Visual Self-Feedback. De tilføjer en anden mekanisme under tegningen, Render-and-Verify: Før modellen accepterer hvert nyt strøg, renderer den det og kontrollerer, om det faktisk ændrede billedet eller blot gentog det forrige. Strøg,

der ikke tilføjer noget, kasseres, og når intet mere hjælper, bliver modellen bedt om at stoppe. Bemærkelsesværdigt nok kører alt dette på et ret lille datasæt — omkring 850,000 eksempler, mindre end halvdelen af det, én konkurrent brugte, og en lille brøkdel af en andens.

Hvad de fandt

- Trænet på denne måde tegner modellen bedre end sin blinde modpart — tydeligst i fejltilfældene, hvor en blind model placerer et øje på en kind eller springer et ønsket søjlediagram over og i stedet leverer en generisk skærm.
- På standardtesten (MMSVGBench) er metoden konkurrencedygtig med stærke rivaler og ligger på flere mål lidt foran dem — herunder OmniSVG, der er trænet på mere end dobbelt så meget data, og InternSVG, der er trænet på omtrent tyve gange så meget.
- Margenerne er små. På ikonsættet er modellens vigtigste mål for billedkvalitet eksempelvis 127.6 mod den bedste rivals 128.8, og dens score for overensstemmelse med prompten er 0.293 mod 0.291 — reelle og konsekvente forskelle, men beskedne.
- Begge tilføjelser fortjener deres plads: Slå den særlige træning eller kontrollen under tegningen fra, og tallene falder målbart. Kontroltrinnet forhindrer især modellen i at sidde fast i at tegne det samme igen og igen.
- Det resultat, forfatterne selv fremhæver, er effektiviteten — at nå så langt med langt færre træningsdata, end de førende modeller brugte.

Hvad dette ikke beviser

- Det viser **ikke**, at det er en gratis gevinst at lade en model "se". Tværtimod: Artiklens eget eksperiment viser, at visuel feedback *uden* genoptræning gør resultatet dårligere. Gevinsten kommer fra genoptræningen, ikke fra øjnene alene.
- Det fastslår **ikke** et stort eller afgørende forspring. På de fleste mål ligger metoden side om side med rivalerne; "slår en model trænet på 20× så meget data" er sandt, men det drejer sig om små udsving i indirekte mål på én standardtest.
- Det demonstrerer **ikke** generel kunstnerisk kunnen. Dette er en forskningsmodel med otte milliarder parametre, som tegner ikoner og enkle illustrationer i en lille fast opløsning (224×224 pixels), ikke en designer til alle formål.
- Sammenligningen med en stor generel model (GPT-5) foregår **ikke** på lige vilkår: Den model er hverken bygget eller finjusteret til denne snævre opgave med at tegne i kode, så det at slå den her siger ikke meget om nogen af modellerne generelt.
- Det er **ikke** gratis. At rendere og genlæse lærredet ved hvert trin gør genereringen langsommere end at sende koden ud på én gang i blinde — en omkostning, forfatterne anerkender.

Hvor stærk er evidensen?

- **Solid** dér, hvor der er tale om en kontrolleret sammenligning på metodens egne præmisser. Ablationsforsøgene er klare: Fjern træningen eller kontrollen, og tallene falder — de to bestanddele udfører altså reelt det arbejde, forfatterne hævder.
- **Ærlig om sin egen overraskelse.** Resultatet om, at naiv visuel feedback *skader*, bliver fremlagt, ikke gemt væk, og det er det mest interessante i artiklen — en nyttig korrektion af intuitionen om, at mere input altid er bedre.
- **Spinkel som påstand om ranglisten.** Sejrene over rivaler med større datamængder er små og findes på en enkelt standardtest, som blev bygget af forfatterne bag en af disse rivaler. Små margener i indirekte mål på ét testsæt er antydende, ikke afgørende.
- **Uafprøvet i stor skala og i praksis.** Alt her handler om ikoner og enkle illustrationer i lav opløsning. Om den samme idé holder til komplekst designarbejde i høj opløsning eller til virkelige anvendelser, står tilbage som fremtidigt arbejde — det siger forfatterne selv.

Hvorfor det er vigtigt

Ideen i centrum af denne forskningsartikel er næsten pinligt enkel, og den rækker langt ud over tegning. Hvis et program skal generere noget ved at skrive kode — en webside, en graf, et diagram, en 3D-scene — kan det enten skrive det hele i blinde og håbe eller rendere undervejs og korrigere kursen. For et menneske er det helt naturligt at lukke denne sløjfe; vi kaster hele tiden et blik på siden. For disse modeller er det et overraskende nyt greb.

Det, der gør artiklen værd at læse, er det forbehold, den føjer til. At give en model mulighed for at se er ikke det samme som at lære den at se efter. Øjnene skal trænes, og først da giver sløjfen udbytte — og selv da er udbyttet reelt, men afmålt: en mere sikker tegner, ikke en anden slags kunstner. For alle, der bygger værktøjer, som forvandler en beskrivelse til redigerbar grafik — den slags, der ender bag ikonerne og illustrationerne i en app — er den stilfærdige, praktiske lærdom den nyttige. Gevinsten her kom fra billigere og smartere træning, ikke fra flere data. Det er mere værd at lære end at hente endnu et point på en rangliste.

Kort sammenfatning

Sprogmodeller, der genererer vektorgrafik — den redigerbare, skalerbare kode bag de fleste ikoner og logoer på nettet — har traditionelt gjort det "i blinde" ved at skrive alle tegnekommandoerne ud uden nogensinde at rendere dem for at se resultatet. Et hold ledet af Guotao Liang foreslår Render-in-the-Loop: Render den halvfærdige tegning efter hvert trin, og før den tilbage, så modellen tegner det næste strøg, mens den ser på lærredet. Deres centrale og ærlige resultat er, at det at gøre dette med en eksisterende model ganske enkelt gør den *dårligere*; gevinsten viser sig først, når modellen genoptrænes til at bruge den visuelle feedback, hjulpet af en kontrol under tegningen, som kasserer

strøg, der ikke ændrer noget. Den genoptrænede model med otte milliarder parametre matcher eller slår med en lille margin rivaler, der er trænet på op til tyve gange flere data, på en standardtest — et reelt resultat, men med små margener på indirekte mål, for ikoner og enkle illustrationer i lav opløsning. Den konklusion, forfatterne fremhæver, og som er værd at tage med, handler om effektivitet: At se sit eget arbejde ordentligt kan opveje ren datamængde.

Nøgtern vurdering

Hvad forskningsartiklen viser: At genoptræne en model til vektorgrafik, så den trin for trin renderer og ser på sin egen halvfærdige tegning, giver bedre og mere fuldstændige resultater end at tegne i blinde — konkurrencedygtigt med og lidt bedre end rivaler med større datamængder på én standardtest, med færre træningsdata.

Hvad der er plausibelt, men ikke bevist: At ”se sit eget arbejde” generelt er en bedre opskrift end at opskalere datamængden; at den samme idé vil hjælpe med kompleks grafik i høj opløsning eller i virkelige anvendelser; at de små margener på standardtesten afspejler en forskel, som et menneske faktisk ville bemærke.

Hvad den ikke viser: At visuel feedback hjælper af sig selv (uden genoptræning skader den); at forspringet til rivalerne er stort eller afgørende; generel tegneevne; en fair direkte sammenligning med generelle modeller som GPT-5, der ikke er bygget til denne opgave.

Vigtigste begrænsninger: Én standardtest, delvist bygget af forfatterne bag en rival; små margener i indirekte mål; en 8B-model, ikoner og enkle illustrationer i 224×224; langsommere generering på grund af sløjfen, der renderer efter hvert trin.

Hvor stor tillid bør en almindelig læser have? Høj tillid til, at det at lukke sløjfen — rendere og genlæse, mens man tegner — reelt hjælper, og at det skal trænes ind, ikke blot slås til. Lav til moderat tillid til, at netop denne model afgørende slår sine rivaler; betragt formuleringen ”slår 20× så meget data” som et reelt, men beskedent resultat fra en enkelt standardtest. Høj tillid til den ene idé, der er værd at huske: For kode, der tegner, er det bedre at se efter undervejs end at tegne i blinde.

Kilder

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/`