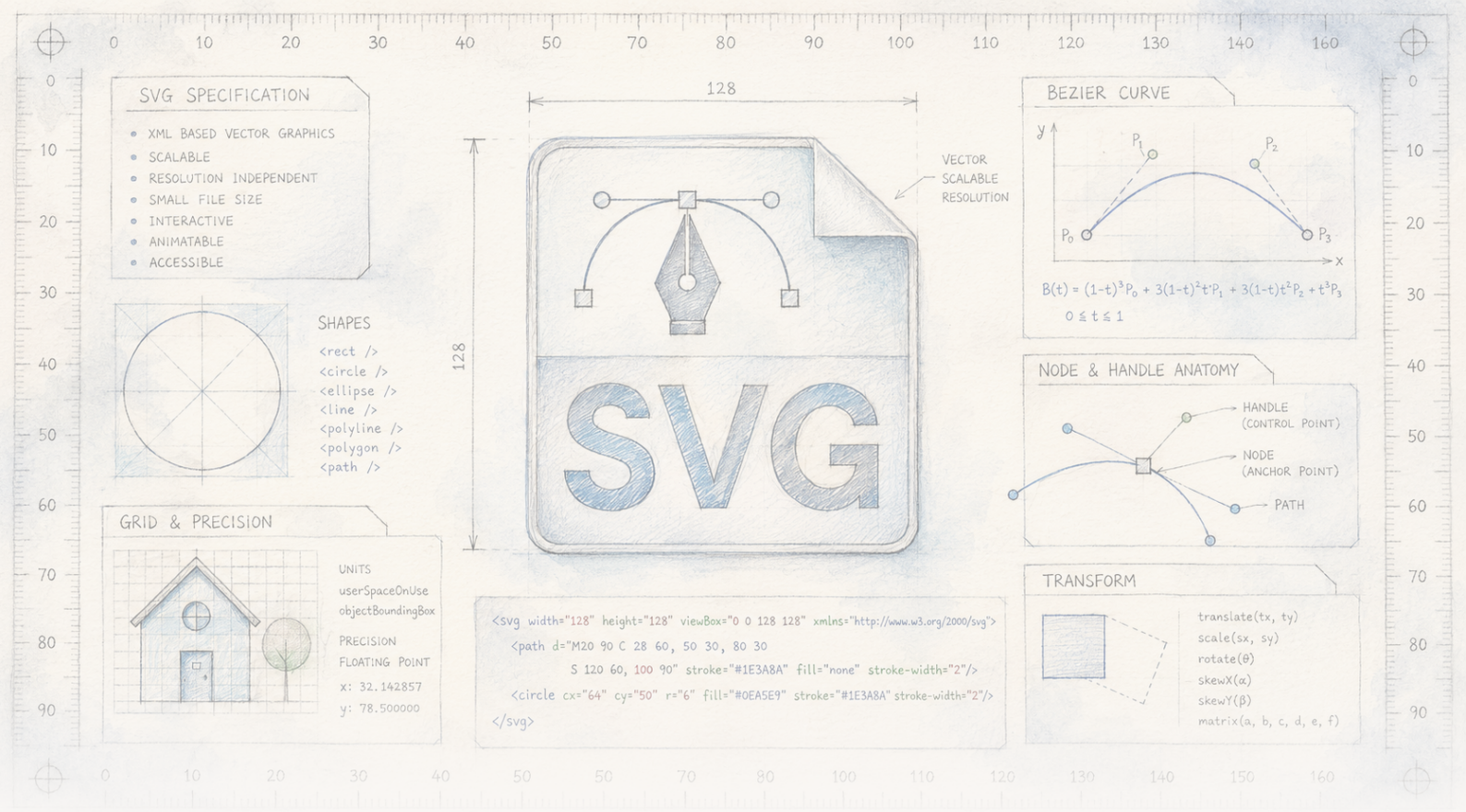


The Clean Paper

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

Att lära en tecknande AI att titta på sidan

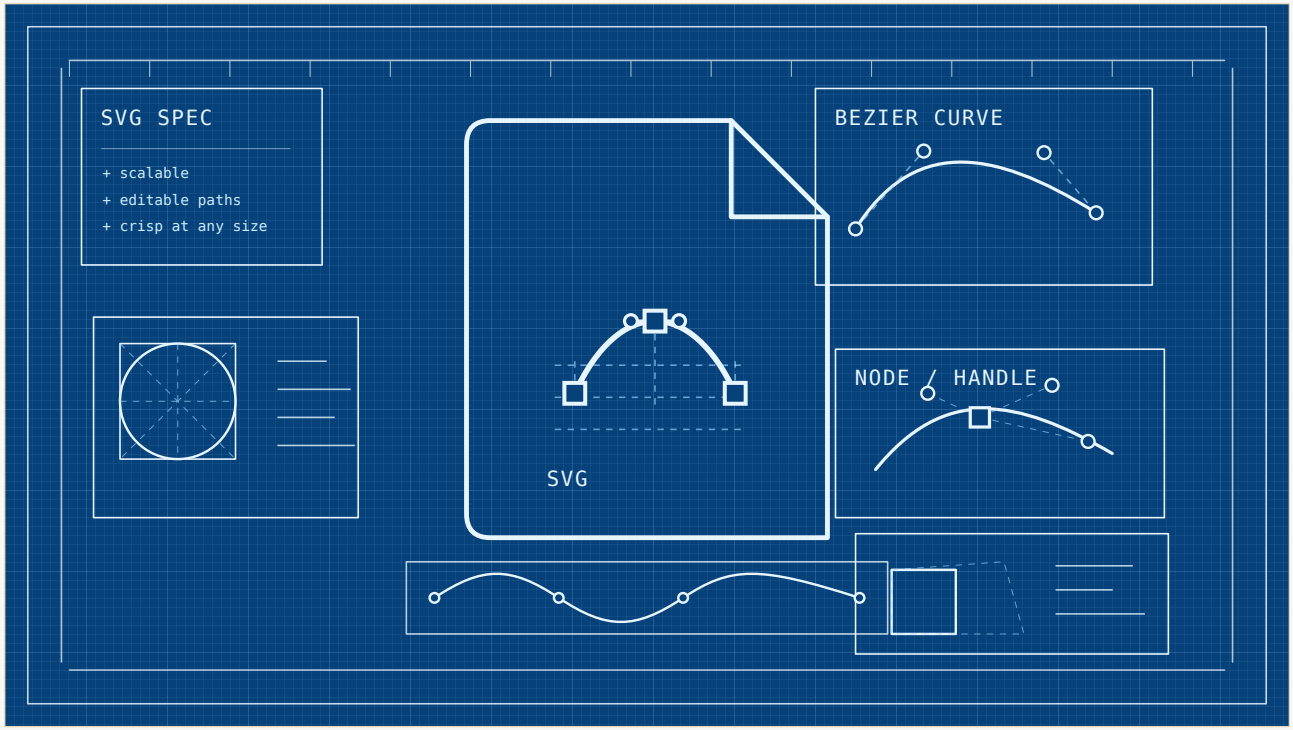
Språkmodeller som genererar vektorgrafik gör det i blindo — de skriver ritkommandona utan att någonsin se resultatet. En ny metod låter modellen se sin egen duk fyllas, streck för streck, och den ärliga vändningen är att det blir sämre om man bara ger den ögon: den måste tränas om för att använda dem. Resultatet är en modell som matchar eller med knapp marginal slår konkurrenter tränade på upp till tjugo gånger mer data — ett verkligt, försiktigt resultat på ett enda standardtest, inte 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)

Att rita med slutna ögon

Be en av dagens AI-modeller att rita en bild åt dig, så händer en av två mycket olika saker under ytan. De välkända bildgeneratorerna — de som trollar fram ett fotografi ur en mening — målar pixlar direkt och kan se duken medan de arbetar. Men det finns ett annat, mer lågmält slags tecknande där modellen inte målar alls. Den skriver instruktioner: *rita en cirkel här, ett streck där, fyll den här formen med blått*. Instruktionerna är kod — samma Scalable Vector Graphics, eller SVG, som ligger bakom de flesta ikoner och logotyper på webben — och fördelen är påtaglig: resultatet är inte ett fast rutnät av pixlar, utan en samling former som kan skalas om, färgas om och redigeras hur länge som helst utan att bli suddiga.



Teknisk SVG-ritning i original: bilden är själv en redigerbar vektorfil, uppbyggd av banor, rutnätslinjer, noder och handtag i stället för fasta pixlar.

Laura Nesso / The Clean Paper Permission on file

Haken är att en modell som skrev sådan ritkod fram till nyligen gjorde det i blindo. Den matade ut hela följden av instruktioner i ett enda svep — cirkel, linje, fyllning — utan att någonsin rendera dem för att se vad den hade åstadkommit. Föreställ dig att skissa ett ansikte med slutna ögon: du kanske placerar ögonen ungefär där de ska sitta, men du kan inte upptäcka att det andra hamnade på kinden eller att formen du ritade som hår nu ligger över näsan. Ungefär så fungerade de här modellerna, vilket är anledningen till att de så ofta producerade kod som var helt giltig men visuellt rörig.

Ett forskarlag lett av Guotao Liang har nu gjort det nästan komiskt självklara: de lät modellen öppna ögonen. Deras metod, *Render-in-the-Loop*, renderar den halvfärdiga teckningen efter varje steg och skickar tillbaka bilden till modellen innan den ritar nästa streck. Det riktigt intressanta är inte att detta hjälper — utan vad de var tvungna att göra för att det över huvud taget skulle hjälpa.

Hur bedömer man en teckning?

När en forskningsartikel säger att dess modell ”slår” en annan är det rimligt att fråga: på vad, och hur mättes det? Det är genuint svårt att bedöma en genererad bild — det finns inget enda rätt svar på ”rita en bärbar dator” — så forskningsfältet använder en handfull automatiska mått, som vart och ett är en bristfällig ersättning för att en människa granskar resultatet.
Några av dem förekommer i den här artikeln. **FID** jämför den övergripande statistiska karaktären hos en hel uppsättning genererade bilder med verkliga bilder; lägre är bättre, men måttet beskriver uppsättningen, inte en enskild bild. **CLIP-poäng** låter en separat AI avgöra om en bild motsvarar orden i prompten — användbart, men inte skarpare än den domaren. **DINO**, **SSIM** och **LPIPS** jämför en rekonstruerad bild med en målbild, på nivåer som sträcker sig från råa pixlar till inlärda egenskaper.
Inget av detta är sanningen. Det är indirekta mått, och de brukar förändras i små steg. När du läser att en modell får 127.6 där en annan får 128.8 är det en verklig skillnad i avsedd riktning — men det är en liten knuff, inte en jordskredsseger, och dessutom en knuff i ett tal som bara löst följer vad ditt öga skulle säga. Det är värt att hålla i minnet när rubriken lyder ”slår en modell tränad på tjugo gånger så mycket data”.

Vad författarna gjorde

Problemet som författarna ville lösa är själva ritandet i blindo. Befintliga modeller behandlar skrivandet av SVG som en ren textuppgift: förutsäg nästa kodstycke utifrån koden hittills och rendera den aldrig. Därmed lämnas de kraftfulla ”ögon” — synkodaren — som moderna multimodala modeller redan har oanvända. Render-in-the-Loop gör om uppgiften till en stegvis visuell process. Efter varje fragment av ritkod renderas den ofullständiga SVG-filen till en bild och matas tillbaka till modellen som en bild, så att nästa fragment väljs medan modellen faktiskt tittar på duken så långt den har kommit.

Deras första resultat manar till försiktighet, och till deras heder redovisar de det tydligt: att bara haka fast den här slingan på en befintlig standardmodell fungerar inte. När de matade mellanliggande renderingar till starka generella modeller utan någon särskild träning förbättrades inte kvaliteten — den *försämrades* över hela linjen. En modell som aldrig har lärt sig att använda ögonen på det här sättet vet inte plötsligt hur den ska göra.

Det mesta av arbetet ligger alltså i undervisningen. De bygger om träningsdatan så att varje teckning delas upp i många små, visuellt meningsfulla steg — komplexa former delas i enklare delar så att det finns något nytt att se i varje skede — och finjusterar sedan en öppen modell med åtta miljarder parametrar (byggd på Qwen3-VL) på dessa stegvisa sekvenser. De kallar detta Visual Self-Feedback. De lägger också till en andra mekanism under ritandet, Render-and-Verify: innan modellen godtar varje nytt streck renderar den det och kontrollerar om bilden faktiskt förändrades eller om strecket bara upprepade det föregående. Streck som inte tillför något kasseras, och när inget mer hjälper uppmanas modellen att sluta. Anmärkningsvärt nog körs allt detta på en ganska liten datauppsättning — omkring 850,000 exempel, mindre än hälften av vad en konkurrent använde och en liten bråkdel av

en annans.

Vad de fann

- Tränad på det här sättet ritar modellen bättre än sin blinda motsvarighet — tydligast syns det i felfallen, där en blind modell placerar ett öga på en kind eller hoppar över ett efterfrågat stapeldiagram och i stället levererar en generell bildskärm.
- På standardtestet (MMSVGBench) står sig metoden väl mot starka konkurrenter och ligger enligt flera mått något före dem — däribland OmniSVG, som tränats på mer än dubbelt så mycket data, och InternSVG, som tränats på ungefär tjugo gånger så mycket.
- Marginalerna är små. På ikonuppsättningen är dess huvudsakliga mått på bildkvalitet till exempel 127.6 mot den bästa konkurrentens 128.8, och dess poäng för överensstämmelse med prompten är 0.293 mot 0.291 — verkliga och konsekventa skillnader, men knappa.
- Båda tilläggen fyller sin funktion: stäng av den särskilda träningen eller verifieringen under ritandet, så sjunker siffrorna mätbart. Verifieringssteget hindrar i synnerhet modellen från att fastna i att rita om samma sak gång på gång.
- Det resultat som författarna själva betonar är effektiviteten — att nå så här långt med betydligt mindre träningsdata än ledarna använde.

Vad detta inte bevisar

- Det visar **inte** att det är en kostnadsfri vinst att låta en modell ”se”. Tvärtom: forskningsartikelns eget experiment visar att visuell återkoppling *utan* omträning gör resultatet sämre. Vinsten kommer från omträningen, inte enbart från ögonen.
- Det fastställer **inte** ett stort eller avgörande försprång. Enligt de flesta mått ligger metoden sida vid sida med sina konkurrenter; ”slår en modell tränad på 20× så mycket data” är sant, men det handlar om små knuffar i indirekta mått på ett enda standardtest.
- Det visar **inte** någon allmän konstnärlig förmåga. Det här är en forskningsmodell med åtta miljarder parametrar som ritar ikoner och enkla illustrationer i en liten fast upplösning (224×224 pixlar), inte en designer för allmänna ändamål.
- Jämförelsen med en stor generell modell (GPT-5) är **inte** rättvisande: den modellen är varken byggd eller finjusterad för denna snäva uppgift att rita med kod, så att slå den här säger inte mycket om någon av modellerna i allmänhet.
- Det är **inte** kostnadsfritt. Att rendera och läsa av duken på nytt i varje steg gör genereringen långsammare än att mata ut koden i ett enda blint svep — en kostnad som författarna medger.

Hur starka är beläggen

- **Starka** när det gäller en kontrollerad jämförelse på metodens egna villkor. Ablationsstudierna är tydliga: ta bort träningen eller verifieringen, så sjunker siffrorna — de två beståndsdelarna utför alltså faktiskt det arbete som författarna hävdar.
- **Ärlig om sin egen överraskning.** Resultatet att naiv visuell återkoppling *skadar* redovisas i stället för att gömmas undan, och det är det mest intressanta i artikeln — en nyttig korrigering av intuitionen att mer indata alltid är bättre.
- **Tunt när det gäller påståendet om topplistan.** Vinsterna över konkurrenter med större datamängder är små och finns på ett enda standardtest, som skapades av författarna bakom en av dessa konkurrenter. Små marginaler i indirekta mått på en enda testuppsättning är antydande, inte avgörande.
- **Oprövad i stor skala och i verkligheten.** Allt här gäller ikoner och enkla illustrationer i låg upplösning. Huruvida samma idé fungerar för komplext designarbete i hög upplösning eller i verkliga tillämpningar lämnas åt framtida forskning — det säger författarna själva.

Varför det spelar roll

Idén i centrum för denna forskningsartikel är nästan pinsamt enkel, och den sträcker sig långt bortom tecknande. Om ett program ska skapa något genom att skriva kod — en webbsida, ett diagram, en schematisk bild, en 3D-scen — kan det antingen skriva alltsammans i blindo och hoppas på det bästa, eller rendera efter hand och korrigera kursen. För en människa är det en självklarhet att sluta återkopplingsslingan; vi kastar hela tiden blickar på sidan. För dessa modeller är det ett förvånansvärt nytt grepp.

Det som gör forskningsartikeln läsvärd är brasklappen den lägger till. Att ge en modell möjlighet att se är inte samma sak som att lära den att titta. Ögonen måste tränas, och först då ger slingan utdelning — och även då är utdelningen verklig men måttlig: en stadigare tecknare, inte ett annat slags konstnär. För alla som bygger verktyg som omvandlar en beskrivning till redigerbar grafik — sådant som hamnar bakom ikonerna och illustrationerna i en app — är den lågmälda, praktiska lärdomen den användbara. Vinsten här kom från billigare, smartare träning, inte från mer data. Det är bättre att lära sig än att få ytterligare en poäng på en topplista.

Kort sammanfattning

Språkmodeller som genererar vektorgrafik — den redigerbara, skalbara koden bakom de flesta ikoner och logotyper på webben — har traditionellt gjort det ”i blindo” genom att skriva ut alla ritkommandon utan att någonsin rendera dem för att se resultatet. Ett forskarlag lett av Guotao Liang föreslår Render-in-the-Loop: rendera den halvfärdiga teckningen efter varje steg och mata tillbaka den, så att modellen ritar nästa streck medan den tittar på duken. Deras centrala och ärliga resultat är att en

befintlig modell blir *sämre* om man bara gör detta; vinsten visar sig först när modellen tränas om för att använda den visuella återkopplingen, med hjälp av en kontroll under ritandet som kasserar streck som inte förändrar något. Den omtränade modellen med åtta miljarder parametrar matchar eller slår med liten marginal konkurrenter som tränats på upp till tjugo gånger mer data i ett standardtest — ett genuint resultat, men med små marginaler i indirekta mått, för ikoner och enkla illustrationer i låg upplösning. Slutsatsen som författarna betonar, och som är värd att ta med sig, handlar om effektivitet: att se sitt arbete ordentligt kan väga upp ren datavolym.

Rakt på sak

Vad forskningsartikeln visar: Att träna om en modell för vektorgrafik så att den steg för steg renderar och tittar på sin egen halvfärdiga teckning ger bättre och mer fullständiga resultat än att rita i blindo — konkurrenskraftigt med och något bättre än rivaler med större datamängder på ett standardtest, med mindre träningsdata.

Vad som är rimligt men inte bevisat: Att ”se sitt arbete” i stort är ett bättre recept än att skala upp datamängden; att samma idé hjälper för komplex grafik i hög upplösning eller i verkliga tillämpningar; att de små marginalerna i standardtestet motsvarar en skillnad som en människa faktiskt skulle märka.

Vad den inte visar: Att visuell återkoppling hjälper av sig själv (utan omträning skadar den); att försprånget över konkurrenterna är stort eller avgörande; en allmän teckningsförmåga; en rättvis direkt jämförelse med generella modeller som GPT-5, som inte är byggda för den här uppgiften.

Huvudsakliga begränsningar: Ett standardtest, delvis skapat av en konkurrents författare; små marginaler i indirekta mått; en 8B-modell, ikoner och enkla illustrationer i 224×224; långsammare generering på grund av slingan som renderar efter varje steg.

Hur stor tilltro bör en vanlig läsare ha? Hög tilltro till att den slutna slingan — att rendera och läsa av på nytt medan man ritar — verkligen hjälper, och att den måste tränas in, inte bara slås på. Låg till måttlig tilltro till att just den här modellen avgörande slår sina konkurrenter; se formuleringen ”slår 20× så mycket data” som ett verkligt men blygsamt resultat från ett enda standardtest. Hög tilltro till den enda idé som är värd att minnas: för kod som ritar är det bättre att titta efter hand än att rita i blindo.

Källor

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/