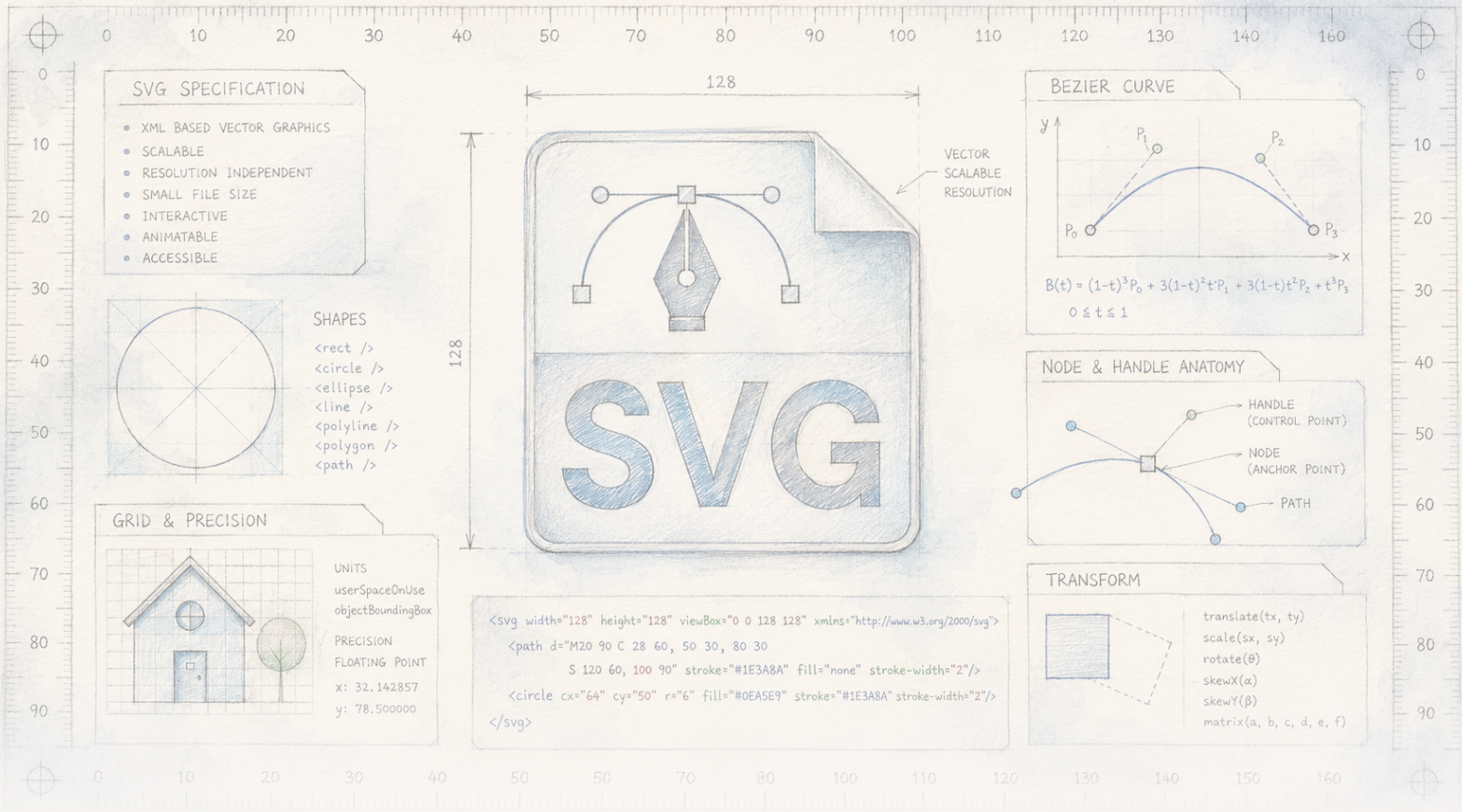


The Clean Paper

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

Սովորեցնել նկարչական AI-ին նայել իր իսկ էջին

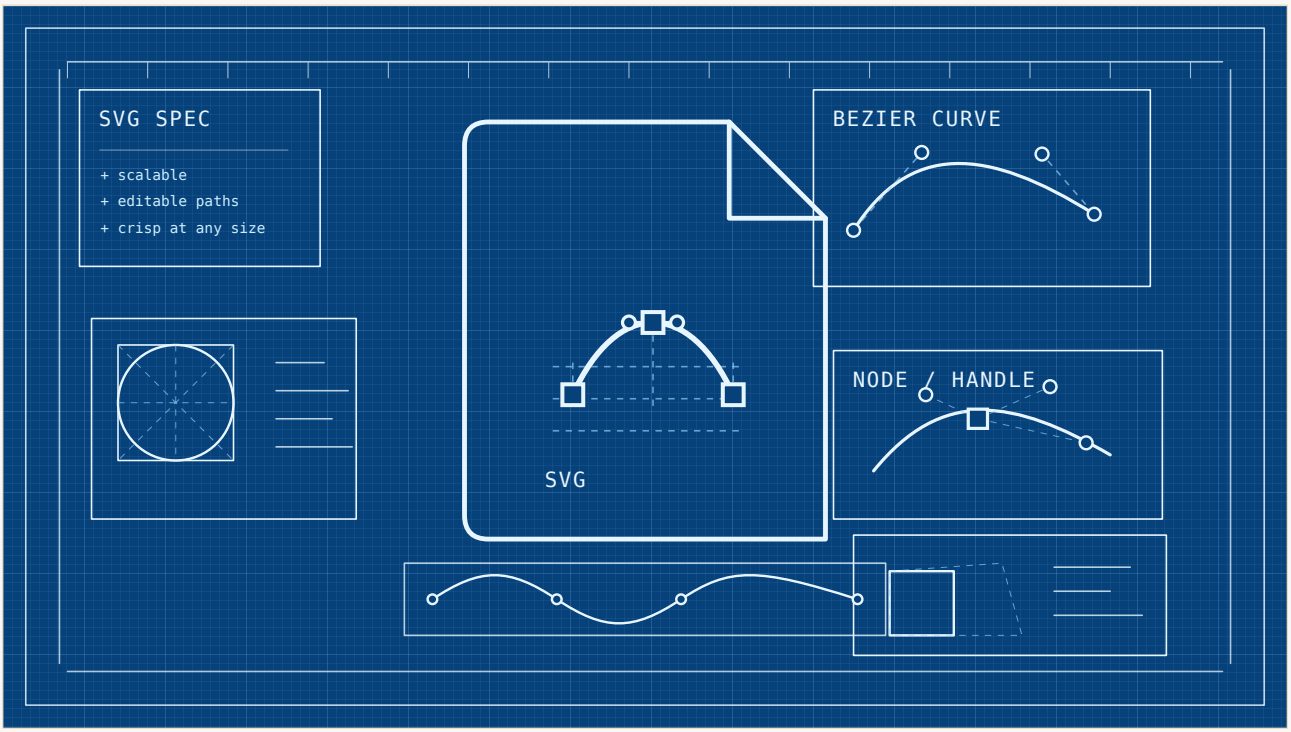
Վեկտորային գրաֆիկա ստեղծող լեզվային մոդելները սովորաբար գրում են նկարչական հրահանգները՝ չտեսնելով արդյունքը: Նոր մեթոդը յուրաքանչյուր քայլից հետո ցույց է տալիս կիսատ կտավը, բայց հետաքրքիր շրջադարձն այն է, որ պարզապես «աչքեր տալը» վատացնում է արդյունքը. մոդելը պետք է հատուկ վերամարզվի տեսածն օգտագործելու համար: Մեկ benchmark-ում վերամարզված մոդելը հավասարվում կամ փոքր-ինչ գերազանցում է մինչև 20 անգամ ավելի շատ տվյալներով մարզված մրցակիցներին՝ իրական, բայց ոչ հեղափոխական արդյունք:

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)

Նկարել՝ փակ աչքերով

Այսօրվա ԱԲ մոդելներից մեկին խնդրեք պատկեր նկարել, և ներսում կարող է տեղի ունենալ երկու բավական տարբեր գործընթացներից մեկը: Ծանոթ պատկերագեներատորները՝ այն համակարգերը, որոնք մեկ նախադասությունից «լուսանկար» են ստեղծում, անմիջապես պիքսելներ են ձևավորում և աշխատանքի ընթացքում փաստորեն տեսնում են կտավը: Բայց կա նաև երկրորդ, ավելի քիչ տեսանելի մոտեցում, որտեղ մոդելը ընդհանրապես չի «ներկում»: Այն գրում է հրահանգներ՝ *այստեղ շրջան գծիր, այնտեղ գիծ քաշիր, այս ձևը կապույտով լցրու*: Այդ հրահանգները կոդ են՝ նույն Scalable Vector Graphics-ը կամ SVG-ն, որը ընկած է համացանցի պատկերակների ու լոգոտիպների մեծ մասի հիմքում: Առավելությունն ակնհայտ է. արդյունքը պիքսելների ֆիքսված ցանց չէ, այլ ձևերի հավաքածու, որը կարելի է անվերջ մեծացնել, փոքրացնել, վերագունավորել և խմբագրել՝ առանց պատկերը մշուշելու:



Բնօրինակ SVG «նախագծային գծագիր». պատկերն ինքնին խմբագրվող վեկտորային ֆայլ է՝ կառուցված ուղիներից, ցանցային գծերից, հանգույցներից և կառավարման բռնակներից, ոչ թե ֆիքսված պիքսելներից:

Laura Nesso / The Clean Paper Permission on file

Մինչ վերջերս խնդիրն այն էր, որ նկարչական կոդ գրող մոդելը դա անում էր գրեթե կույր: Այն մեկ անցումով արտածում էր հրահանգների ամբողջ հաջորդականությունը՝ շրջան, գիծ, լցում, առանց դրանք որևէ պահի պատկերելու և տեսնելու, թե իրականում ինչ ստացվեց: Պատկերացրեք դեմք եք նկարում փակ աչքերով. գուցե աչքերը մոտավորապես ճիշտ տեղում հայտնվեն, բայց չեք նկատի, եթե երկրորդ աչքը այտին նստի կամ մազերի համար գծած ձևը ծածկի քիթը: Այդպես էին աշխատում այս

մոդելները, և դրա համար էլ հաճախ ստացվում էր տեխնիկապես ճիշտ SVG կոդ, որը տեսողականորեն վատ էր։

Գուոտաո Լիանգի ղեկավարած թիմը հիմա կատարել է գրեթե չափազանց ակնհայտ քայլը՝ մոդելին թույլ տվել «բացել աչքերը»։ Նրանց *Render-in-the-Loop* մեթոդը յուրաքանչյուր քայլից հետո պատկերում է կիսատ նկարը և հաջորդ գիծը գծելուց առաջ այդ պատկերը վերադարձնում մոդելին։ Իրական հետաքրքիր մասը այն չէ, որ սա կարող է օգնել, այլ այն, թե ինչ է պետք եղել անել, որպեսզի այն ընդհանրապես օգնի։

Ինչպե՞ս են գնահատում նկարը

Երբ աշխատանքում գրվում է, որ մի մոդել «գերազանցում է» մյուսին, արժե հարցնել՝ ինչում և ինչ չափմամբ։ Գեներացված նկարը գնահատելն իսկապես դժվար է. «նկարիր նոութբուք» խնդրանքի մեկ ճիշտ պատասխան չկա։ Այդ պատճառով ոլորտը հենվում է մի քանի ավտոմատ չափորոշիչների վրա, որոնցից յուրաքանչյուրը միայն մոտավոր փոխարինում է մարդու գնահատականին։
Այս աշխատանքում հանդիպում են մի քանիսը։ **FID**-ը գեներացված պատկերների ամբողջ խմբի վիճակագրական հատկությունները համեմատում է իրական պատկերների հետ. ավելի փոքր թիվը լավ է, բայց այն նկարագրում է խմբաքանակը, ոչ կոնկրետ մեկ նկարը։ **CLIP գնահատական**-ը մեկ այլ ԱԲի օգնությամբ գնահատում է, թե պատկերը որքանով է համապատասխանում տեքստային հրահանգին. օգտակար է, բայց այնքան լավ, որքան այդ «դատավորը»։ **DINO**, **SSIM** և **LPIPS** չափումները վերականգնված պատկերը համեմատում են թիրախի հետ՝ հում պիքսելներից մինչև սովորած հատկանիշներ տարբեր մակարդակներում։
Դրանցից ոչ մեկը «ճշմարտությունը» չէ։ Դրանք proxy-ներ են և հաճախ փոխվում են փոքր քայլերով։ Եթե մեկ մոդել ստանում է 127.6, իսկ մյուսը՝ 128.8, դա կարող է լինել իրական տարբերություն ցանկալի ուղղությամբ, բայց փոքր տարբերություն է, ոչ մեծ հաղթանակ, և այն էլ մի թվի մեջ, որը մարդու տեսողական դատողությանը միայն անուղղակիորեն է կապված։ Սա կարևոր է հիշել, երբ վերնագիրն ասում է՝ «գերազանցել է 20 անգամ ավելի շատ տվյալներով մարզված մոդելին»։

Ի՞նչ արեցին հեղինակները

Հեղինակների հիմնական թիրախը հենց «կույր նկարելն» էր։ Գոյություն ունեցող մոդելները SVG գրելը հիմնականում դիտում են որպես տեքստային խնդիր՝ կանխատեսել կոդի հաջորդ հատվածը՝ ելնելով մինչ այդ գրված կոդից, և երբեք չպատկերել արդյունքը։ Այդպիսով անգործուն է մնում ժամանակակից բազմամոդալ մոդելների հզոր «աչքը»՝ տեսողական encoder-ը։ Render-in-the-Loop-ը խնդիրը վերակազմակերպում է որպես քայլ առ քայլ տեսողական գործընթաց։ Յուրաքանչյուր կոդային հատվածից

հետո մասնակի SVG-ն պատկեր է դառնում և վերադարձվում մոդելին, որպեսզի հաջորդ հատվածը ընտրվի՝ արդեն տեսնելով մինչ այդ եղած կտավը։

Նրանց առաջին արդյունքը զգուշացնող է, և հեղինակները այն բացահայտ ներկայացնում են. այս օղակը պարզապես պատրաստ մոդելի վրա ավելացնելը չի աշխատում։ Երբ միջանկյալ render-ները տրվել են ուժեղ ընդհանուր մոդելներին՝ առանց հատուկ մարզման, որակը չի բարձրացել, այլ *վատացել է* գրեթե բոլոր չափումներով։ Եթե մոդելին նախկինում չեն սովորեցրել այս խնդրի համար օգտագործել իր տեսողական մուտքը, այն ինքնաբերաբար չի սովորում դրանից օգտվել։

Ուստի աշխատանքի հիմնական մասը հենց ուսուցումն է։ Հեղինակները վերակառուցել են ուսուցման տվյալները այնպես, որ յուրաքանչյուր նկար բաժանվի բազմաթիվ փոքր, տեսողական իմաստ ունեցող քայլերի. բարդ ձևերը մասնատվել են ավելի պարզ բաղադրիչների, որպեսզի յուրաքանչյուր փուլում իրոք նոր բան հայտնվի։ Այնուհետև նրանք fine-tune են արել ութ միլիարդ պարամետր ունեցող բաց մոդել՝ Qwen3-VL-ի հիման վրա, այս քայլային հաջորդականությունների վրա։ Մոտեցումը անվանում են Visual Self-հետադարձ կապ։ Նկարչության պահին ավելացվել է նաև երկրորդ մեխանիզմ՝ Render-and-Verify. յուրաքանչյուր նոր stroke ընդունելուց առաջ այն render է արվում և ստուգվում, արդյոք իրականում փոխել է պատկերը, թե պարզապես կրկնել նախորդը։ Ոչինչ չավելացնող stroke-ները մերժվում են, իսկ երբ նոր քայլերն այլևս չեն օգնում, մոդելին հուշվում է կանգ առնել։ Հատկանշական է, որ ամբողջ ուսուցումն իրականացվել է համեմատաբար փոքր տվյալների հավաքածուով՝ մոտ 850,000 օրինակ, այսինքն՝ մեկ մրցակցի տվյալների կեսից էլ պակաս և մյուսի օգտագործած ծավալի փոքր մասնիկը։

Ի՞նչ գտան

- Այս եղանակով մարզված մոդելը ավելի լավ է նկարում, քան իր «կույր» տարբերակը։ Տարբերությունն առավել տեսանելի է ձախողման դեպքերում՝ երբ կույր մոդելը, օրինակ, աչքը տեղադրում է այտին կամ խնդրված սյունակային գրաֆիկի փոխարեն սովորական monitor է նկարում։
- Ստանդարտ MMSVGBench չափանիշային թեստում մեթոդը մրցունակ է ուժեղ մրցակիցների հետ և մի շարք չափումներով փոքր առավելություն ունի, այդ թվում՝ OmniSVG նկատմամբ, որը մարզվել է ավելի քան երկու անգամ շատ տվյալներով, և InternSVG նկատմամբ, որը մոտ 20 անգամ ավելի շատ տվյալ է օգտագործել։
- Տարբերությունները փոքր են։ Օրինակ՝ icon հավաքածուում հիմնական պատկերիորակի միավորը 127.6 է՝ լավագույն մրցակցի 128.8-ի դիմաց, իսկ հրահանգ համապատասխանությ միավորը՝ 0.293՝ 0.291-ի դիմաց։ Դրանք իրական և նույն ուղղությամբ տարբերություններ են, բայց փոքր։
- Երկու նոր բաղադրիչներն էլ կարևոր են։ Եթե անջատվում է հատուկ ուսուցումը կամ նկարչության պահին կատարվող verification-ը, միավորները չափելիորեն ընկնում են։ Verification-ը հատկապես կանխում է այն, որ մոդելը անվերջ նույն բանն է վերագծում։

- Հեղինակների ընդգծած գլխավոր արդյունքը տվյալների արդյունավետությունն է՝ մրցունակ մակարդակի հասնել շատ ավելի քիչ ուսուցման տվյալներով:

Ի՞նչ սա չի ապացուցում

- Սա **չի ցույց տալիս**, որ մոդելին «տեսնելու» հնարավորություն տալը ինքնուրույն անվճար առավելություն է: Ընդհակառակը, աշխատանքի սեփական փորձը ցույց է տալիս, որ տեսողական հետադարձ կապը *առանց վերամարզման* վատացնում է արդյունքը: Օգուտը գալիս է ուսուցումից, ոչ պարզապես տեսողական մուտքի առկայությունից:
- Սա **չի հաստատում մեծ կամ վճռական առավելություն**: Չափումների մեծ մասում մեթոդը մրցակիցներին շատ մոտ է: «Գերազանցում է 20× ավելի շատ տվյալներով մարզված մոդելին» ձևակերպումը ճշմարիտ է, բայց խոսքը մեկ չափանիշային թեստի proxy չափումներում փոքր տարբերությունների մասին է:
- Սա **չի ցուցադրում ընդհանուր գեղարվեստական կարողություն**: Սա 8 միլիարդ պարամետր ունեցող հետազոտական մոդել է, որը 224×224 պիքսել փոքր լուծաչափով նկարում է հիմնականում պատկերակներ և պարզ նկարագարդումներ, ոչ ընդհանուր նպատակների դիզայներ:
- GPT-5-ի նման մեծ ընդհանուր մոդելի հետ համեմատությունը **ուղիղ և հավասար պայմաններով չէ**. այդ մոդելը չի ստեղծվել կամ հատուկ մարզվել այս նեղ կոդային նկարչության խնդրի համար, ուստի այստեղ այն գերազանցելը քիչ բան է ասում ընդհանուր կարողությունների մասին:
- Մոտեցումը **անվճար չէ նաև հաշվարկային առումով**: Յուրաքանչյուր քայլում render անելն ու պատկերը նորից կարդալը դանդաղեցնում է գեներացիան՝ համեմատած կոդը մեկ կույր անցումով արտածելու հետ: Հեղինակները դա ընդունում են:

Որքա՞ն ուժեղ է ապացույցը

- **Ամուր** է վերահսկվող համեմատության շրջանակում: Ablation փորձերը պարզ են. հեռացրեք հատուկ ուսուցումը կամ verification-ը, և ցուցանիշները վատանում են: Այսինքն՝ երկու բաղադրիչներն էլ իրականում կատարում են այն աշխատանքը, որը հեղինակները վերագրում են դրանց:
- **Անկեղծ է անսպասելի բացասական արդյունքի մասին**: Այն փաստը, որ միամիտ տեսողական հետադարձ կապը *վնասում է*, չի թաքցվել: Դա աշխատանքի ամենահետաքրքիր արդյունքներից մեկն է և օգտակար հակակշիռ է «ավելի շատ input միշտ ավելի լավ է» ինտուիցիային:
- **Ավելի թույլ է leaderboard պնդման մասում**: Ավելի մեծ տվյալներով մրցակիցների նկատմամբ առավելությունները փոքր են և ստացվել են մեկ չափանիշային թեստում, որը մասամբ ստեղծվել է հենց մրցակից համակարգերից մեկի հեղինակների կողմից: Proxy չափումների փոքր տարբերությունները մեկ թեստ set-ում խոստումնալից

են, բայց վերջնական պատասխան չեն։
- **Չի փորձարկվել մեծ մասշտաբում ու իրական աշխատանքային միջավայրում**։ Այստեղ հիմնականում փոքր icon-ներ և պարզ նկարազարդումներ են։ Բարդ, բարձր լուծաչափով կամ իրական դիզայնի խնդիրների վրա նույն գաղափարի արդյունավետությունը ապագա աշխատանքի հարց է, և հեղինակներն էլ դա նշում են։

Ինչո՞ւ է սա կարևոր

Աշխատանքի կենտրոնական գաղափարը գրեթե անհարմար չափով պարզ է և նկարելուց շատ ավելի լայն կիրառություն ունի։ Եթե ծրագիրը ինչոր բան ստեղծում է կոդ գրելով՝ web page, chart, diagram, 3D scene, այն կարող է ամբողջ կոդը գրել կույր և հուսալ, որ արդյունքը ճիշտ կլինի, կամ յուրաքանչյուր քայլից հետո render անել և ուղղել ընթացքը։ Մարդու համար երկրորդը բնական է. մենք անընդհատ նայում ենք էջին։ Մոդելների համար այդ փակ օղակը համեմատաբար նոր մոտեցում է։

Աշխատանքն արժեքավոր է հենց այն աստղանիշի պատճառով, որը դնում է այս պարզ գաղափարի կողքին։ Մոդելին տեսնելու հնարավորություն տալը նույնը չէ, ինչ նրան *նայել սովորեցնելը*։ Տեսողական մուտքը պետք է դառնա ուսուցման մաս, և միայն դրանից հետո օղակը օգուտ է տալիս։ Նույնիսկ այդ ժամանակ շահույթը իրական է, բայց չափավոր՝ ավելի կայուն գծագրող, ոչ նոր տեսակի նկարիչ։ Նկարագրությունը խմբագրվող գրաֆիկայի վերածող գործիքներ կառուցողների համար գործնական դասը կարևոր է. այստեղ հաջողությունը եկել է ավելի խելացի և տվյալների առումով էժան ուսուցումից, ոչ պարզապես ավելի մեծ տվյալների հավաքածուից։

Կարճ ամփոփում

Վեկտորային գրաֆիկա ստեղծող լեզվային մոդելները՝ web պատկերակների և լոգոտիպների հիմքում ընկած խմբագրվող, անսահման մասշտաբվող կոդը, սովորաբար աշխատել են «կույր»՝ ամբողջ նկարչական կոդը գրելով առանց արդյունքը render անելու։ Գուոտաո Լիանգի ղեկավարած թիմի Render-in-the-Loop մեթոդը յուրաքանչյուր քայլից հետո պատկերում է կիսատ աշխատանքը և այն վերադարձնում մոդելին, որպեսզի հաջորդ stroke-ը ընտրվի տեսած կտավի հիման վրա։ Հիմնական և ազնիվ արդյունքը այն է, որ այդ հետադարձ կապը պարզապես պատրաստ մոդելին տալը արդյունքը *վատացնում է*։ Օգուտն առաջանում է միայն այն ժամանակ, երբ մոդելը վերամարզվում է տեսողական հետադարձ կապը օգտագործելու համար, և երբ նկարչության պահին ստուգվում ու մերժվում են ոչինչ չփոխող stroke-ները։ Վերամարզված 8B մոդելը մեկ ստանդարտ չափանիշային թեստում հավասարվում կամ փոքրինչ գերազանցում է մինչև 20 անգամ ավելի շատ տվյալներով մարզված

մրցակիցներին։ Սա իրական, բայց փոքր տարբերություն է proxy չափումներում՝ ցածր լուծաչափով icon-ների ու պարզ պատկերների վրա։ Հեղինակների կարևոր դասը արդյունավետությունն է. սեփական աշխատանքը լավ տեսնել սովորեցնելը կարող է որոշ չափով փոխարինել պարզապես տվյալների ծավալը մեծացնելուն։

Առանց չափազանցության

Ի՞նչ է ցույց տալիս աշխատանքը։ Վեկտորային գրաֆիկա ստեղծող մոդելին քայլ առ քայլ render անել և իր կիսատ նկարը տեսնել *ուսուցանելը* ավելի լավ ու ամբողջական արդյունք է տալիս, քան կույր կոդ գեներացնելը։ Մեկ ստանդարտ չափանիշային թեստում այն մրցունակ է և փոքր առավելություն ունի շատ ավելի մեծ տվյալների հավաքածուներով մարզված որոշ մրցակիցների նկատմամբ։

Ի՞նչն է հավանական, բայց ապացուցված չէ։ Որ «աշխատանքը տեսնելը» լայնորեն ավելի լավ ռազմավարություն է, քան տվյալների հավաքածուի մասշտաբավորումը, որ նույն գաղափարը կօգնի բարդ բարձրլուծաչափով կամ իրական աշխարհի գրաֆիկայի վրա, կամ որ փոքր չափանիշային թեստ տարբերությունները մարդու աչքին նկատելի կլինեն։

Ի՞նչ չի ցույց տալիս։ Որ տեսողական հետադարձ կապը ինքնուրույն օգնում է՝ առանց վերաուսուցման այն վնասում է։ Չի ցույց տալիս մեծ ու վճռական առավելություն մրցակիցների նկատմամբ, ընդհանուր նպատակների նկարչական կարողություն կամ արդար ուղիղ համեմատություն GPT-5-ի նման ընդհանուր մոդելների հետ։

Գլխավոր սահմանափակումները։ Մեկ չափանիշային թեստ, որը մասամբ ստեղծվել է մրցակցի հեղինակների կողմից, proxy չափումների փոքր տարբերություններ, 8B մոդել, 224×224 icon-ներ ու պարզ նկարազարդումներ և ավելի դանդաղ գեներացիա՝ յուրաքանչյուր քայլում render անելու պատճառով։

Որքա՞ն վստահություն պետք է ունենա ընդհանուր ընթերցողը։ Բարձր վստահություն, որ օղակը փակելը՝ նկարելու ընթացքում render անելն ու արդյունքը նորից կարդալը, իրոք օգնում է, և որ դա պետք է *ուսուցանել*, ոչ պարզապես միացնել։ Ցածրից միջին վստահություն այն պնդման նկատմամբ, որ հենց այս մոդելը վճռականորեն գերազանցում է մրցակիցներին. «20× ավելի քիչ/շատ տվյալների» վերնագրային համեմատությունը իրական է, բայց համեստ և մեկ չափանիշային թեստից։ Ամենահուսալի գաղափարը պարզ է՝ նկարչական կոդի դեպքում ընթացքը տեսնելը ավելի լավ է, քան կույր աշխատելը։

Աղբյուրներ

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/