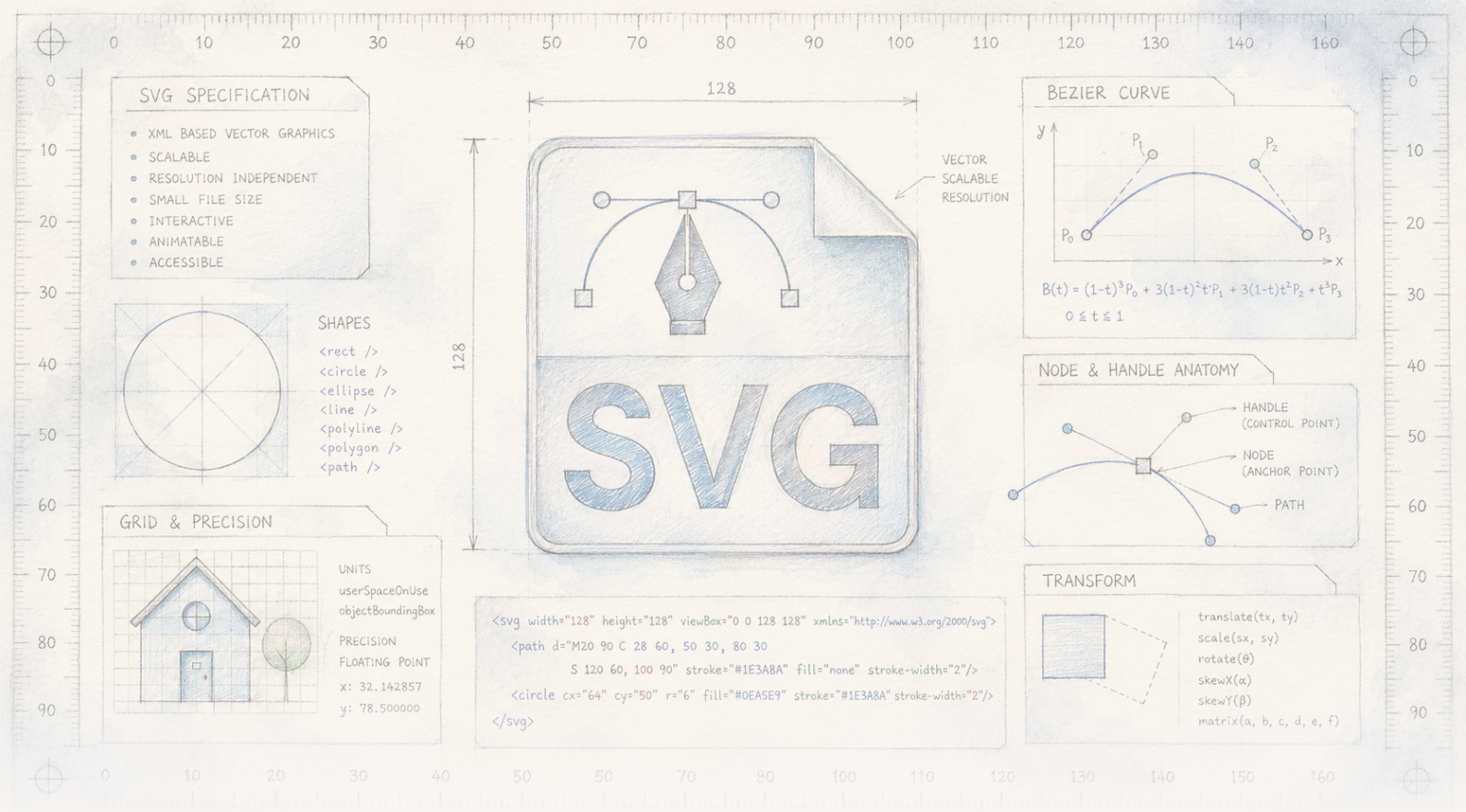


The Clean Paper

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

Iemācīt zīmējošam MI skatīties uz lapu

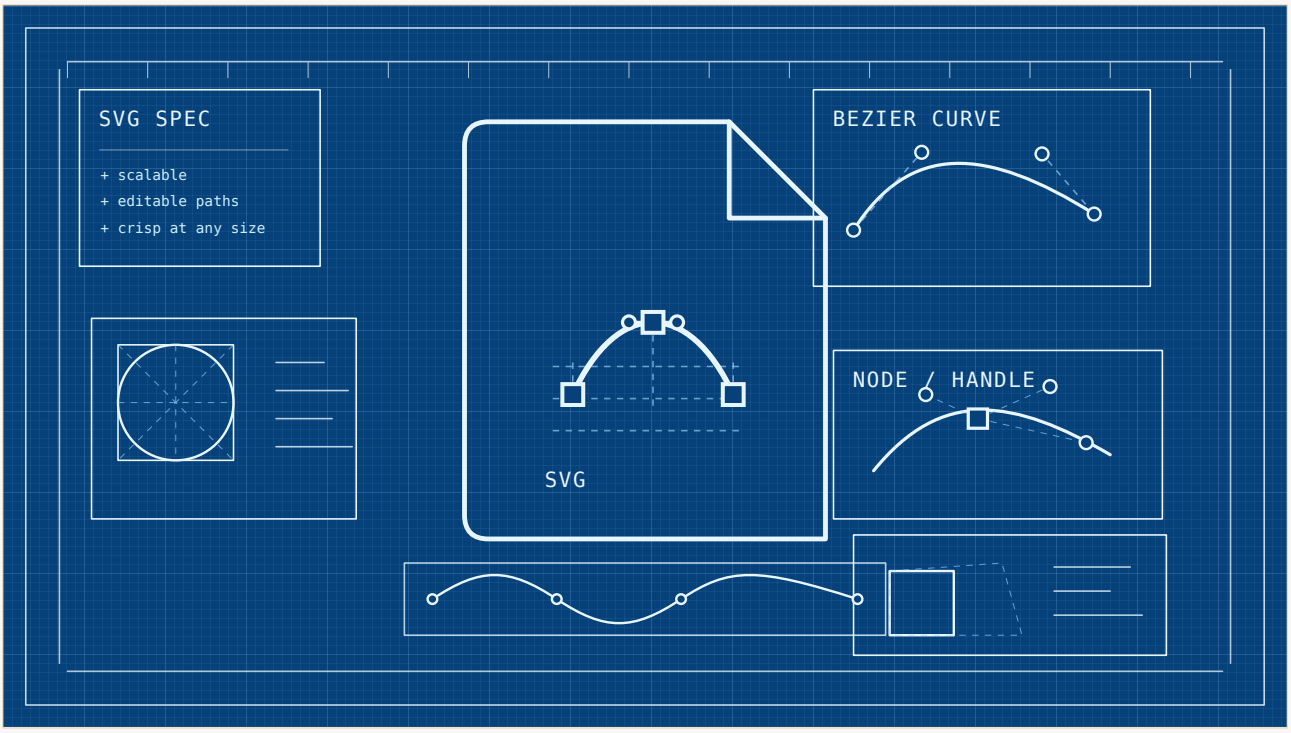
Valodas modeļi, kas ģenerē vektorgrafiku, tradicionāli to dara akli — raksta zīmēšanas komandas, neredzot rezultātu. Jauna metode ļauj modelim vērot, kā audekls piepildās soli pa solim, taču svarīgākais pavērsiens ir tas, ka vienkārši "iedot acis" rezultātus pasliktina: modelis jāpārtrenē, lai tas spētu redzi izmantot. Pēc tam tas vienā etalonā sasniedz vai nedaudz pārspēj konkurentus, kas trenēti ar līdz pat divdesmit reižu vairāk datu — īsts, rūpīgs rezultāts, nevis revolūcija.

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)

Zīmēt ar aizvērtām acīm

Palūdzot mūsdienu MI modelim uzzīmēt attēlu, "zem motora pārsega" var notikt viena no divām ļoti atšķirīgām lietām. Pazīstamie attēlu ģeneratori — tie, kas no viena teikuma rada fotogrāfijai līdzīgu ainu — tieši veido pikseļus un darba gaitā redz savu audeklu. Taču ir arī klusāks zīmēšanas veids, kurā modelis vispār neko "nekrāso". Tas raksta instrukcijas: *šeit uzzīmē apli, tur līniju, šo formu aizpildi zilā krāsā*. Šīs instrukcijas ir kods — tas pats Scalable Vector Graphics jeb SVG formāts, kas ir pamatā daudziem tīmekļa ikonām un logotipiem. Priekšrocība ir īsta: rezultāts nav fiksēts pikseļu režģis, bet formu kopums, ko var bezgalīgi mērogot, pārkrāsot un rediģēt, nezaudējot asumu.



Oriģināls SVG tehniskais rasējums: pats attēls ir rediģējams vektoru fails, kas veidots no ceļiem, režģa līnijām, mezgliem un vadības punktiem, nevis fiksētiem pikseļiem.

Laura Nesso / The Clean Paper Permission on file

Problēma bija tā, ka vēl pavisam nesen modelis šo zīmēšanas kodu rakstīja akli. Tas vienā piegājienā izdeva visu instrukciju secību — aplis, līnija, aizpildījums —, nekad to neatveidojot, lai redzētu, kas īsti sanācis. Iedomājieties, ka mēģināt skicēt seju ar aizvērtām acīm: aptuveni zināt, kur acīm vajadzētu atrasties, taču nevarat pamanīt, ka otra acs nonākusi uz vaiga vai ka matiem paredzētā forma tagad aizsedz degunu. Aptuveni tā šie modeļi strādāja, un tāpēc tie tik bieži radīja pilnīgi derīgu kodu, kas vizuāli izskatījās pēc jucekļa.

Guotao Liang vadīta komanda tagad izdarījusi gandrīz komiski pašsaprotamo: ļāvusi modelim atvērt acis. Viņu metode *Render-in-the-Loop* pēc katra soļa atveido nepabeigto zīmējumu un parāda šo attēlu modelim, pirms tas ģenerē nākamo vilcienu. Patiesi interesantais nav tas, ka šāda atgriezeniskā saite palīdz, bet gan tas, kas bija jādara, lai tā vispār sāktu palīdzēt.

Kā novērtēt zīmējumu?

Ja pētījums saka, ka tā modelis "pārspēj" citu, ir godīgi jautāt: ko tieši un pēc kāda mērījuma? Ģenerēta attēla kvalitāti novērtēt ir grūti — uzdevumam "uzzīmē klēpjdatoru" nav vienas vienīgas pareizas atbildes —, tāpēc joma balstās uz vairākiem automātiskiem rādītājiem, no kuriem katrs ir tikai nepilnīgs aizstājējs cilvēka skatienam.
Daži no tiem izmantoti arī šajā darbā. **FID** salīdzina veselas ģenerētu attēlu kopas statistiskās īpašības ar īstu attēlu kopu; zemāka vērtība ir labāka, taču tā raksturo kopu, nevis atsevišķu attēlu. **CLIP score** lūdz citam MI modelim novērtēt, vai attēls atbilst teksta uzvednei — tas ir noderīgi, taču tikai tik labi, cik labs ir pats vērtētājs. **DINO**, **SSIM** un **LPIPS** salīdzina rekonstruētu attēlu ar mērķa attēlu dažādos līmeņos — no pikseļiem līdz iemācītām pazīmēm. Neviens no šiem rādītājiem nav patiesības mērs. Tie ir aizstājējrādītāji, un atšķirības bieži ir nelielas. Ja lasāt, ka viens modelis iegūst 127,6, bet cits 128,8, tā ir reāla atšķirība paredzētajā virzienā — taču drīzāk neliels grūdiens nekā nogruvums, turklāt skaitlī, kas tikai aptuveni atspoguļo to, ko teiktu cilvēka acs. To ir vērts paturēt prātā, kad virsrakstā parādās "pārspēj modeli, kas trenēts ar divdesmit reižu vairāk datu".

Ko autori darīja

Autori mēģināja novērst pašu aklo zīmēšanas procesu. Līdzšinējie modeļi SVG rakstīšanu uztver kā tīri teksta uzdevumu: paredz nākamo koda fragmentu no līdzšinējā koda un neko neatveido. Tādējādi neizmantots paliek spēcīgais "redzes" komponents — vizuālais kodētājs —, kas jau ir iebūvēts mūsdienu multimodālajos modeļos. Render-in-the-Loop pārveido uzdevumu par pakāpenisku vizuālu procesu. Pēc katra zīmēšanas koda fragmenta daļējais SVG tiek atveidots attēlā un padots modelim atpakaļ, lai nākamais fragments tiktu izvēlēts, faktiski skatoties uz līdz šim uzzīmēto.

Pirmais rezultāts ir labs brīdinājums, un autori to godīgi pasaka: vienkārši pievienot šādu cilpu gatavam modelim nestrādā. Kad starprezultātu attēlus deva spēcīgiem vispārīgiem modeļiem bez īpašas apmācības, kvalitāte nevis uzlabojās, bet *pasliktinājās* visos testos. Modelis, kuram nekad nav mācīts izmantot redzi šim uzdevumam, pēkšņi neiemācās to darīt tikai tāpēc, ka attēls tagad ir pieejams.

Tāpēc lielākā darba daļa ir apmācība. Pētnieki pārveido treniņdatus tā, lai katrs zīmējums būtu sadalīts daudzos mazos, vizuāli jēgpilnos soļos — sarežģītas formas sadalot vienkāršākos elementos, lai katrā posmā parādītos kaut kas jauns, ko redzēt —, un pēc tam papildus trenē astoņu miljardu parametru atvērto modeli, kura pamatā ir Qwen3-VL, uz šīm pakāpeniskajām secībām. Viņi šo pieeju sauc par Visual Self-Feedback. Zīmēšanas laikā tiek pievienots vēl viens mehānisms — Render-and-Verify: pirms katra jaunā vilciena pieņemšanas modelis to atveido un pārbauda, vai attēls patiešām mainījies, nevis tikai atkārtots iepriekšējais solis. Vilcieni, kas neko nepievieno, tiek atmesti, bet, kad turpināšana vairs nepalīdz, modelim tiek dots signāls apstāties. Zīmīgi, ka viss tas trenēts ar samērā nelielu datu kopu — aptuveni 850 000 piemēru, mazāk nekā puse no viena konkurenta datiem un tikai neliela daļa no otra izmantotā apjoma.

Ko viņi atklāja

- Šādi apmācīts modelis zīmē labāk nekā tā "aklā" versija — īpaši skaidri kļūdu piemēros, kur aklais modelis novieto aci uz vaiga vai pieprasītas joslu diagrammas vietā izveido vispārīgu monitoru.
- Standarta etalonā MMSVGBench metode ir konkurētspējīga un vairākos rādītājos nedaudz pārspēj spēcīgus konkurentus — tostarp OmniSVG, kas trenēts ar vairāk nekā divreiz lielāku datu kopu, un InternSVG, kam treniņdatu bija aptuveni divdesmit reižu vairāk.
- Atšķirības ir mazas. Piemēram, ikonu kopā galvenais attēla kvalitātes rādītājs ir 127,6 pret labākā konkurenta 128,8, bet uzvednes atbilstības rādītājs — 0,293 pret 0,291. Atšķirības ir reālas un konsekventas, taču nelielas.
- Abas pievienotās sastāvdaļas dod izmērāmu ieguldījumu: izslēdzot īpašo apmācību vai ģenerēšanas laikā veikto pārbaudi, rezultāti pasliktinās. Pārbaudes solis īpaši palīdz novērst situāciju, kur modelis iestrēgst un atkal un atkal pārzīmē vienu un to pašu.
- Paši autori visvairāk uzsver efektivitāti — līdzīgu rezultātu sasniegšanu ar ievērojami mazāku treniņdatu apjomu nekā vadošajiem konkurentiem.

Ko tas nepierāda

- Tas **neparāda**, ka iespēja "redzēt" automātiski dod priekšrocību. Patiesībā darba pašu eksperiments rāda pretējo: vizuāla atgriezeniskā saite *bez* pārtrenēšanas pasliktina rezultātus. Ieguvumu dod apmācība izmantot redzi, nevis redze pati par sevi.
- Tas **nepierāda** lielu vai izšķirošu pārsvaru. Vairumā rādītāju metode ir ļoti tuvu konkurentiem; frāze "pārspēj modeli ar 20× vairāk datu" ir patiesa, taču atšķirības ir mazas, tās redzamas aizstājējrādītājos un vienā etalonā.
- Tas **neapliecina** vispārīgas māksliniecisk as spējas. Šis ir astoņu miljardu parametru pētniecības modelis, kas zīmē ikonas un vienkāršas ilustrācijas nelielā fiksētā 224×224 pikseļu izšķirtspējā, nevis universāls dizainers.
- Salīdzinājums ar lielu vispārīgu modeli (GPT-5) **nav līdzvērtīgs**: tas nav būvēts vai īpaši pielāgots šaurajam koda zīmēšanas uzdevumam, tāpēc uzvara šeit maz ko pasaka par abu modeļu vispārīgajām spējām.
- Metode nav **bez maksas** skaitļošanas ziņā. Audekla atveidošana un atkārtota nolasīšana pēc katra soļa padara ģenerēšanu lēnāku nekā visa koda izdošana vienā aklā piegājienā — izmaksas, ko autori atzīst.

Cik pārliecinoši ir pierādījumi?

- **Stabili** tajā, ko kontrolēts salīdzinājums patiešām pārbauda. Ablācijas ir skaidras: izņemot īpašo apmācību vai pārbaudes soli, rezultāti krītas. Tātad abas sastāvdaļas dara to, ko autori apgalvo.
- **Godīgi pret pašu pārsteigumu.** Fakts, ka naivi pievienota vizuāla atgriezeniskā saite *kaitē*, netiek paslēpts. Tas, iespējams, ir interesantākais darba rezultāts — labs pretpiemērs intuīcijai, ka vairāk

ievades vienmēr nozīmē labāk.

- **Vājāki**, ja runa ir par līderu tabulas secinājumu. Pārsvars pār konkurentiem ar lielākām datu kopām ir neliels un redzams vienā etalonā, kuru izveidojuši viena no šo konkurentu modeļu autori. Nelielas starpības aizstājējrādītājos vienā testu kopā ir daudzsološas, nevis galīgas.
- **Nav pārbaudīts mērogā un reālos apstākļos.** Šeit testētas ikonas un vienkāršas ilustrācijas nelielā izšķirtspējā. Vai tā pati ideja saglabā priekšrocības sarežģītā, augstas izšķirtspējas vai reālā dizaina darbā, paliek nākotnes pētījumiem — autori to skaidri atzīst.

Kāpēc tas ir svarīgi

Šā darba centrālā ideja ir gandrīz mulsinoši vienkārša un sniedzas tālu ārpus zīmēšanas. Ja programma ar kodu ģenerē kaut ko vizuālu — tīmekļa lapu, grafiku, diagrammu vai 3D ainu —, tā var uzrakstīt visu akli un cerēt uz labu rezultātu vai arī darba gaitā atveidot starprezultātu un koriģēt nākamos soļus. Cilvēkam šāda atgriezeniskā cilpa šķiet pašsaprotama; mēs nepārtraukti uzmetam skatienu lapai. Modeļiem tā ir pārsteidzoši jauna pieeja.

Darbu lasīt vērts tieši tāpēc, ka tam ir svarīga piebilde. Dot modelim iespēju redzēt nav tas pats, kas iemācīt tam skatīties. “Acis” vispirms jāapmāca, un tikai tad cilpa sāk palīdzēt — turklāt arī tad ieguvums ir īsts, bet mērens: stabilāks zīmētājs, nevis pilnīgi cita veida mākslinieks. Ikvienam, kas būvē rīkus, kuri no apraksta rada rediģējamu grafiku — piemēram, lietotņu ikonas un ilustrācijas —, noderīgākā ir klusā praktiskā atziņa. Šeit uzvara nāca no gudrākas un lētākas apmācības, nevis no vienkārši lielāka datu apjoma. Tas ir vērtīgāks secinājums par vēl vienu punktu līderu tabulā.

Īss kopsavilkums

Valodas modeļi, kas ģenerē vektorgrafiku — rediģējamu un mērogojamu kodu, kas ir daudzu tīmekļa ikonu un logotipu pamatā — tradicionāli to darījuši “akli”, izrakstot visas zīmēšanas komandas, nekad neatveidojot rezultātu. Guotao Liang vadīta komanda piedāvā Render-in-the-Loop: pēc katra soļa atveidot nepabeigto zīmējumu un padot to modelim atpakaļ, lai nākamais vilciens tiktu veidots, skatoties uz audeklu. Galvenais un godīgākais rezultāts ir tas, ka vienkārši pievienot šo iespēju gatavam modelim ir sliktāk, nevis labāk; ieguvums parādās tikai pēc pārtrenēšanas izmantot vizuālo atgriezenisko saiti un ar papildus pārbaudi, kas atmet vilcienus, kuri neko nemaina. Pārtrenētais astoņu miljardu parametru modelis standarta etalonā sasniedz vai nedaudz pārspēj konkurentus, kas trenēti ar līdz pat divdesmit reižu vairāk datu. Tas ir īsts rezultāts, bet ar nelielām starpībām aizstājējrādītājos, ikonām un vienkāršām ilustrācijām zemā izšķirtspējā. Vērtīgākais secinājums ir par efektivitāti: labi iemācīta spēja skatīties uz savu darbu var daļēji aizstāt vienkāršu datu apjoma palielināšanu.

Bez pārspīlējumiem

Ko pētījums parāda: pārtrenējot vektorgrafikas modeli tā, lai tas soli pa solim atveidotu un aplūkotu savu nepabeigto zīmējumu, tiek iegūti labāki un pilnīgāki rezultāti nekā aklā zīmēšanā — vienā standarta etalonā konkurētspējīgi un nedaudz labāki par modeļiem ar daudz lielākām treniņdatu kopām.

Kas ir ticams, bet nav pierādīts: ka "skatīšanās uz savu darbu" kopumā ir labāka stratēģija par datu apjoma palielināšanu; ka tā pati ideja palīdzēs sarežģītai, augstas izšķirtspējas vai reālai grafikai; ka nelielās etalona starpības būtu acīmredzami pamanāmas cilvēkam.

Ko tas neparāda: ka vizuāla atgriezeniskā saite palīdz pati par sevi (bez pārtrenēšanas tā kaitē); ka pārsvars pār konkurentiem ir liels vai izšķirošs; vispārīgas zīmēšanas spējas; taisnīgu salīdzinājumu ar vispārīgiem modeļiem, piemēram, GPT-5, kas nav veidoti šim uzdevumam.

Galvenie ierobežojumi: viens etalons, ko daļēji veidojuši konkurenta autori; nelielas starpības aizstājējrādītājos; 8B modelis, ikonas un vienkāršas ilustrācijas 224×224 izšķirtspējā; lēnāka ģenerēšana, jo pēc katra soļa jāveic atveidošana.

Cik lielai pārliecībai vajadzētu būt vispārīgam lasītājam? Augstai par to, ka cilpas noslēgšana — atveidot un vēlreiz apskatīt zīmējumu darba gaitā — patiešām palīdz, un ka šī prasme ir jāiemāca, nevis vienkārši jāieslēdz. Zemai līdz vidējai par to, ka konkrētais modelis izšķiroši pārspēj konkurentus; frāzi "pārspēj ar 20× vairāk datu trenētu modeli" uztveriet kā īstu, bet nelielu viena etalona rezultātu. Augstai par vienu ideju, ko vērts atcerēties: kodam, kas zīmē, skatīties darba gaitā ir labāk nekā zīmēt akli.

Avoti

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/`