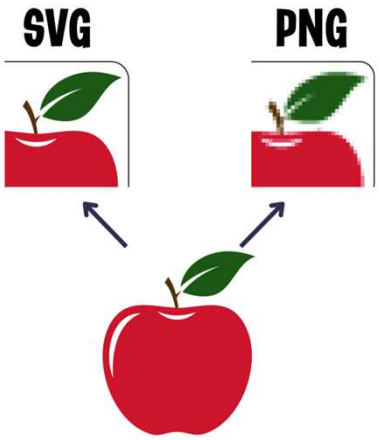
Week 7
Structured Vector Graphics Editing With VLMs

Matthew Perlman

James Beetham, Dr. Amrit Bedi, Dr. Mubarak Shah

Background

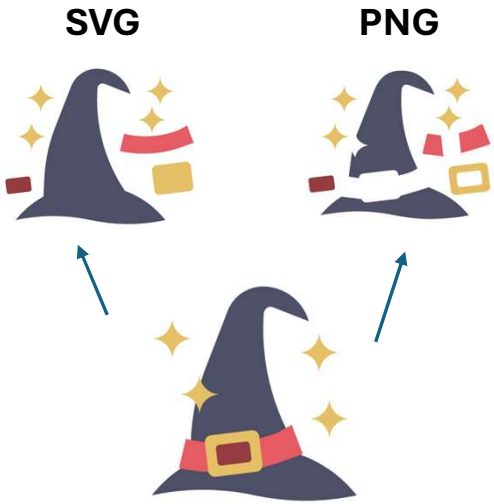
Why do we want to use SVG images instead of raster images?



Lossless Upscaling



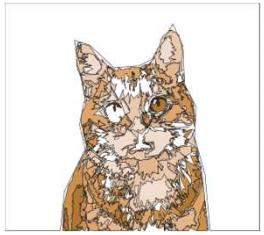
Reduced Storage



Human Editable

Existing Work

Combining the expressive power of frontier VLMs with diffusion models has "solved" generating rasters. For SVGs… not so much.



GPT-5.5



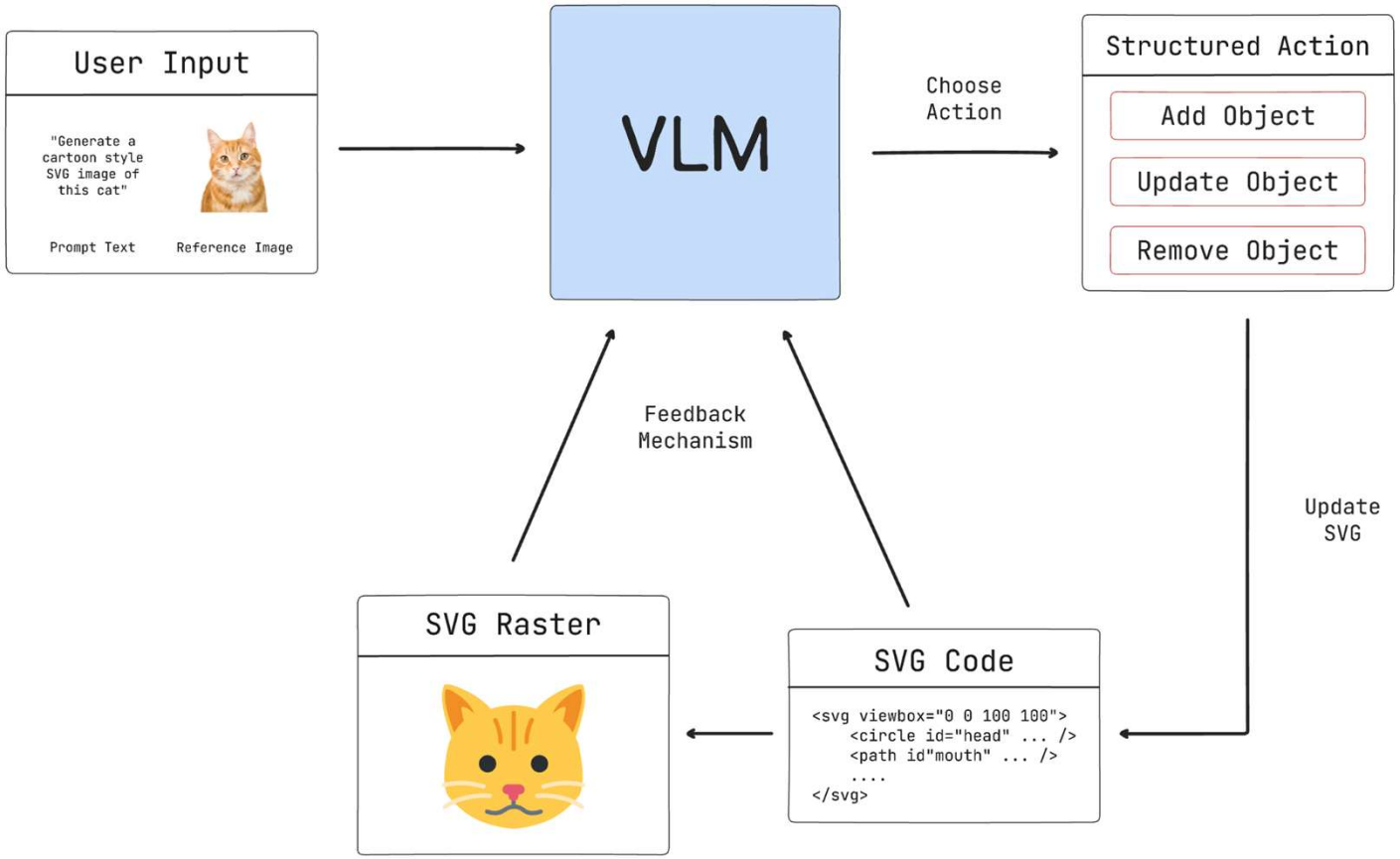
Sonnet-4.6

Dedicated raster-to-vector models perform better, but are **limited** to mostly icons and logos, often produce **overly complex** code, and require **separate training** pipelines.

Why Off-the-Shelf VLMs Fail

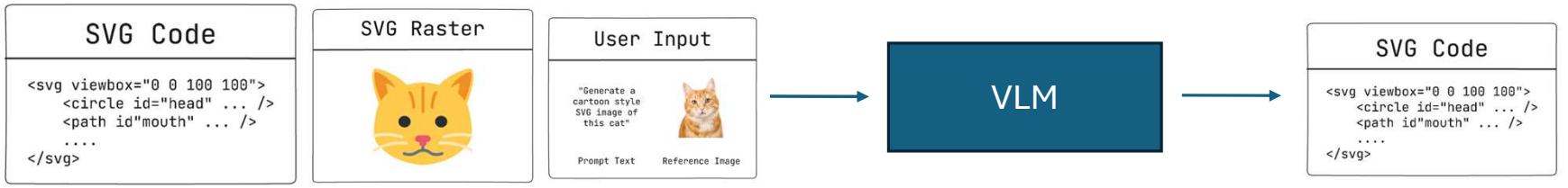
- Smaller models **lack reasoning skill** needed to produce full SVG code
- VLMs **never see** the result, they guess how code looks
- They produce code autoregressively, making choices **irreversible**

Our Solution – Iterative Refinement

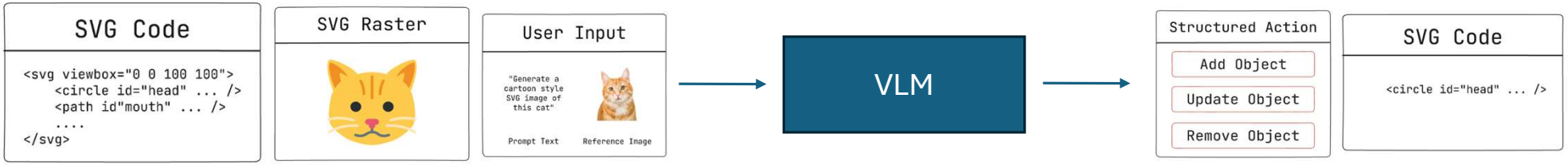


Pipeline Variants

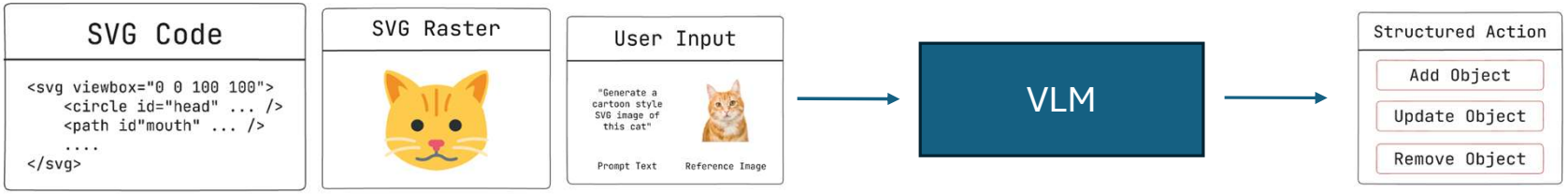
Unstructured Actions



Semi-Structured Actions



Structured Actions



Structured Outputs + Constrained Decoding

- Models natively produce text, but we want to **enforce structure**:
  - Enumerated Options (ex: "True" or "False", nothing else allowed)
  - JSON Template (ex: { name: string, age: number })
  - Context-Free-Grammar (ex: code syntax, XML, nested data)
- Modern language models support structured outputs implemented through a constrained decoding engine

Progress This Week

- **Sped up CFG** constrained decoding in VLLM (~36 hr → ~2hr)
- **Finalized results** across 3 models × 4 pipelines × 2 constraints × 2 tasks
- Pacific Graphics **Two Page Technical Paper** (due July 15th)
  - Abstract (Incomplete)
  - Introduction (Drafted)
  - Related Work (Complete)
  - System Overview (Drafted)
  - Results (Pending Results)
  - Limitations + Future Work (Complete)

Final Text-to-SVG Evaluation Metrics

Model	Decoding	Pipeline	Compile↑	CLIP↑	Aesthetic↑	# Tokens
Qwen3.5-2B	Unconstrained	Single Turn	**0.270**	**0.213**	4.49	487
		Structured	0.190	0.199	**4.53**	3423
		Semi-Structured	0.266	0.189	4.51	5268
		Unstructured	0.190	0.211	4.47	4685
	Constrained	Single Turn	0.150	0.199	4.35	433
		Structured	0.163	0.196	**4.54**	2446
		Semi-Structured	**0.603**	0.183	4.52	2908
		Unstructured	0.089	**0.217**	4.53	3754
Qwen3.5-9B	Unconstrained	Single Turn	**0.887**	**0.250**	**4.60**	1428
		Structured	0.627	0.216	4.46	3303
		Semi-Structured	0.190	0.195	4.33	7718
		Unstructured	0.46	0.234	4.47	6664
	Constrained	Single Turn	**0.997**	0.185	**4.55**	161
		Structured	0.627	0.209	4.43	2628
		Semi-Structured	0.473	0.196	4.44	2996
		Unstructured	0.400	**0.226**	4.45	5060
Qwen3.5-27B	Unconstrained	Single Turn	0.787	**0.260**	**4.71**	1379
		Structured	**0.943**	0.236	4.50	3248
		Semi-Structured	0.170	0.215	4.50	8090
		Unstructured	0.490	0.257	4.65	2871
	Constrained	Single Turn	0.663	0.255	4.51	1049
		Structured	**0.953**	0.227	4.45	2149
		Semi-Structured	0.550	0.215	4.45	2366
		Unstructured	0.797	**0.257**	**4.52**	5425

Final Image-to-SVG Evaluation Metrics

Model	Decoding	Pipeline	Compile↑	DINO↑	LPIPS↓	# Tokens
Qwen3.5-2B	Unconstrained	Single Turn	**0.360**	0.442	0.564	447
		Structured	0.137	0.422	**0.533**	3674
		Semi-Structured	0.157	0.447	0.543	6063
		Unstructured	0.123	**0.456**	0.584	5092
	Constrained	Single Turn	0.257	0.444	0.608	428
		Structured	0.110	0.429	0.556	3310
		Semi-Structured	**0.303**	0.378	0.580	2641
		Unstructured	0.077	**0.515**	**0.529**	4365
Qwen3.5-9B	Unconstrained	Single Turn	**0.593**	**0.654**	**0.421**	1769
		Structured	0.400	0.573	0.445	8152
		Semi-Structured	0.427	0.556	0.454	8457
		Unstructured	0.173	0.625	0.432	9350
	Constrained	Single Turn	**1.000**	0.264	0.639	156
		Structured	0.507	0.552	**0.449**	3398
		Semi-Structured	0.343	0.428	0.542	3019
		Unstructured	0.433	**0.578**	0.457	4675
Qwen3.5-27B	Unconstrained	Single Turn	0.667	0.753	0.370	1652
		Structured	**0.900**	0.692	0.380	8095
		Semi-Structured	0.573	0.677	0.382	9102
		Unstructured	0.053	**0.782**	**0.311**	12018
	Constrained	Single Turn	0.660	0.558	0.518	546
		Structured				
		Semi-Structured	0.407	0.500	0.469	3434
		Unstructured	**0.86**	**0.656**	**0.402**	5432

Our Model vs. SOTA Models

Text-to-SVG	**CLIP**↑	**Aesthetic**↑	**# Tokens**
Best SOTA	0.299	4.94	0.6k
Worst SOTA	0.185	4.26	132k
Unconstrained Qwen3.5-27B with Single-Turn Generation	0.260	4.71	1.3k
Unconstrained Qwen3.5-27B with Unstructured Iterative Refinement	0.257	4.65	2.9k

Image-to-SVG	**DINO**↑	**LPIPS**↓	**# Tokens**
Best SOTA	0.994	0.035	0.3k
Worst SOTA	0.860	0.403	322k
Unconstrained Qwen3.5-27B with Unstructured Iterative Refinement	0.782	0.311	12k
Unconstrained Qwen3.5-27B with Single-Turn Generation	0.753	0.370	1.6k

Key Paper Takeaways

- Iterative Refinement does not consistently improve visual fidelity
  - Models struggle to implement useful changes from visual feedback
- In-Distribution Task > Out-of-Distribution Task
  - Outputting homogenous SVG code or JSON is easier than mixing
- Interface > Model Size
  - Successful generation can vary more by pipeline than by parameter count
- Constrained Decoding improves compilation and hurts quality
  - Constraining outputs pushes the model toward compliant but unlikely/bad token choices
- Compilation Rate and Visual Quality are distinct objectives

Next Steps

- **Finalize writing** for abstract, intro, system overview, and results
- Submission-Quality **diagrams** of our iterative refinement system
- Generate **samples from GPT** for paper-ready example outputs
- Submit!