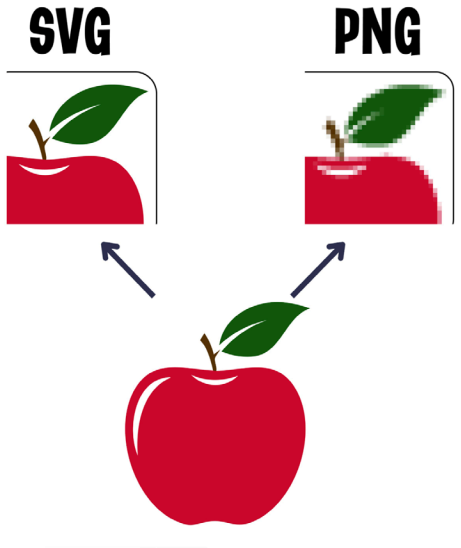
Week 4
Structured Vector Graphics Editing With VLMs

Matthew Perlman

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

Background

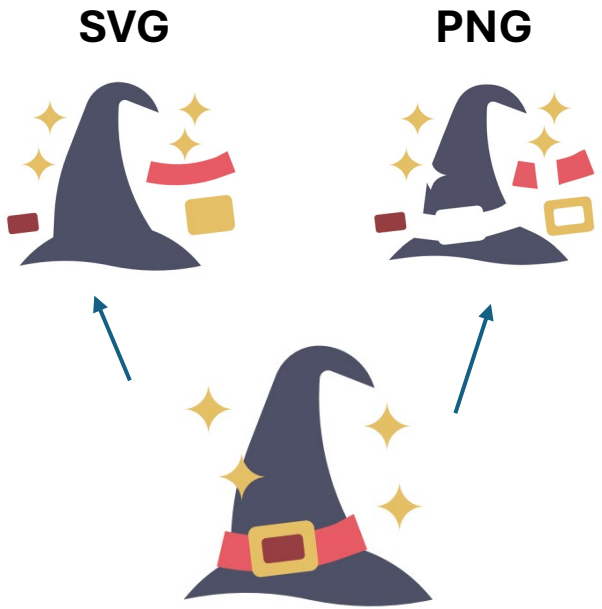
Why do we want to use SVG images instead of raster images?



Lossless Upscaling



Reduced Storage



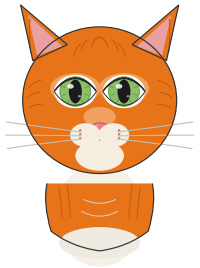
Human Editable

Existing Work

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



GPT-5.5



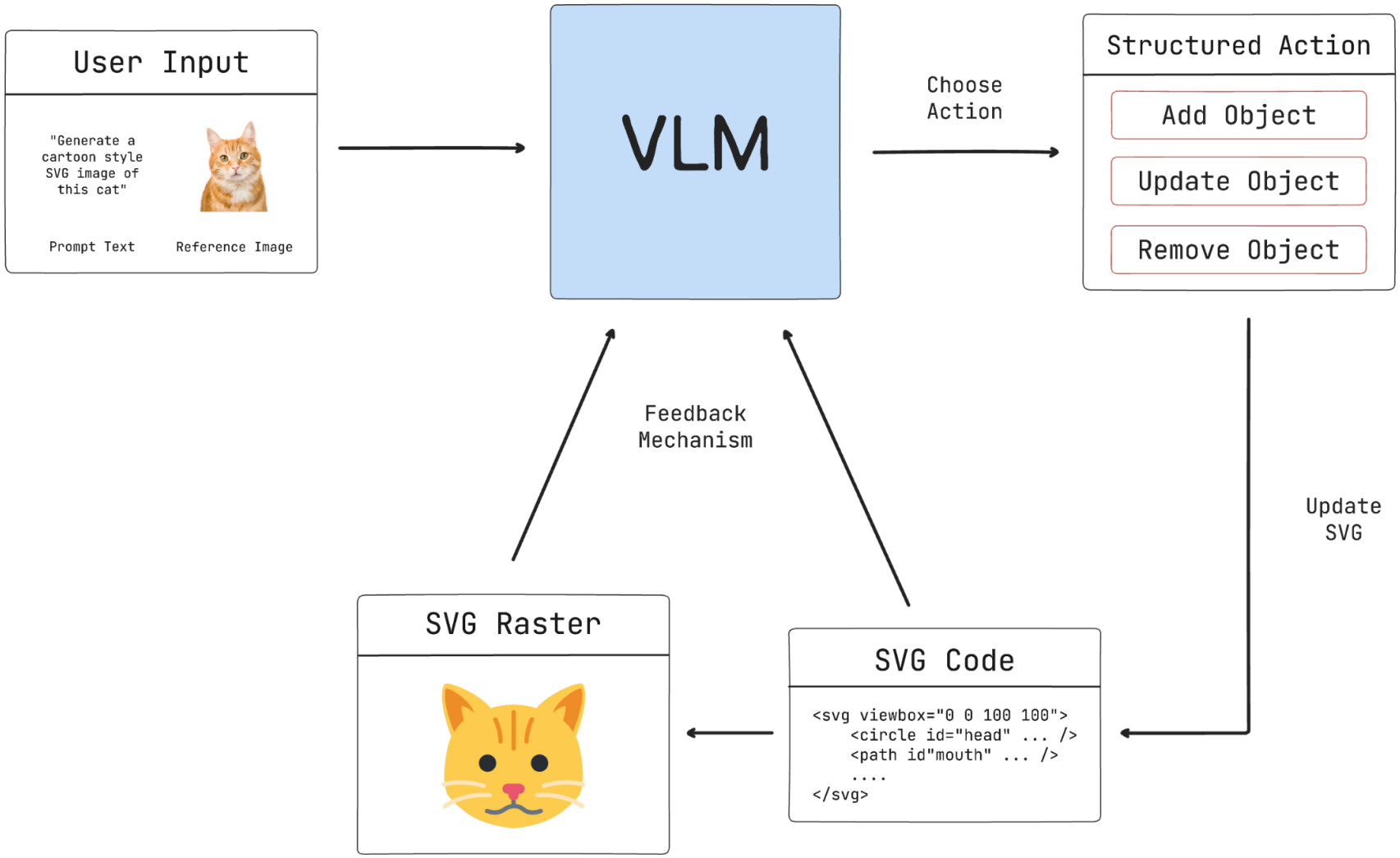
Sonnet-4.6

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

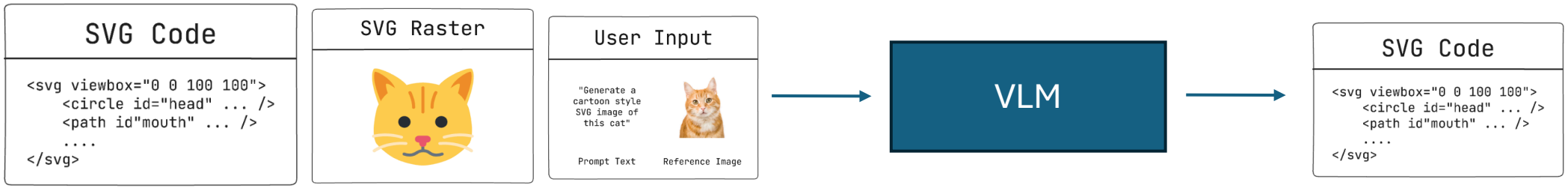
- Smaller models **lack reasoning skill** needed to produce full SVG code
- VLMs **never see** the result, they guess how code looks
- They produce code autoregressively, making choices **irreversible**

Our Solution – Iterative Refinement

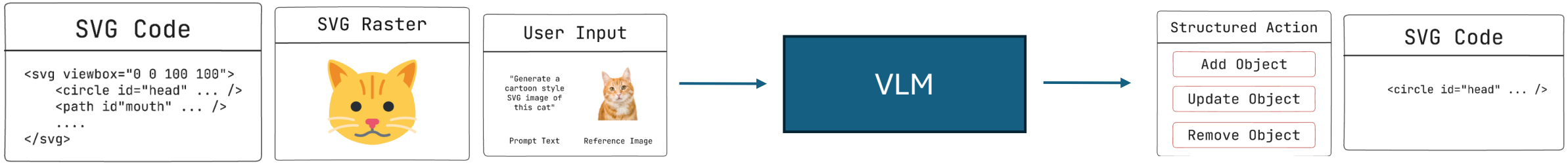


Pipeline Variants

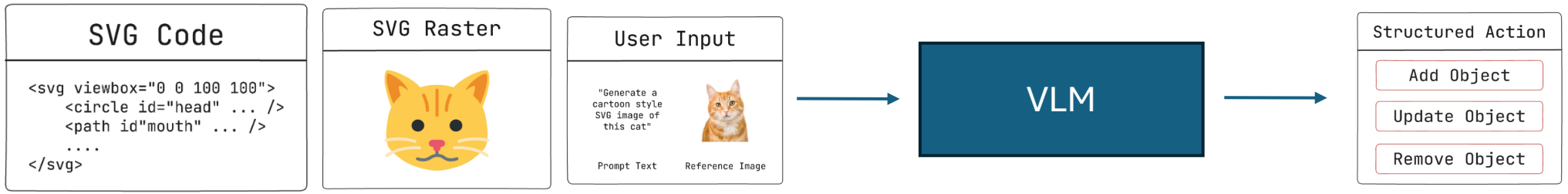
Unstructured Actions



Semi-Structured Actions



Structured Actions



Progress This Week

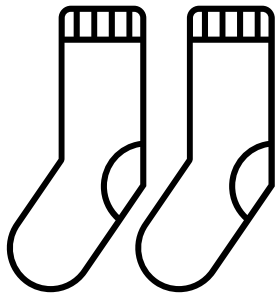
- Implemented **standard metrics** for generative SVG models
  - Successful Compilation Rate
  - Average Component Count
  - CLIP Caption Alignment
  - CLIP Reference Image Alignment
  - DINO Reference Image Alignment

- Setup the **MMSVG-2M Dataset** composed by OmniSVG

- **Evaluated iterative refinement** variants on a subset of MMSVG-2M

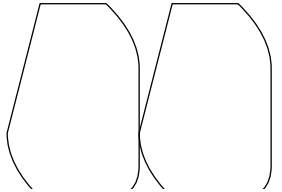
Evaluating Iterative Refinement

USER INPUT

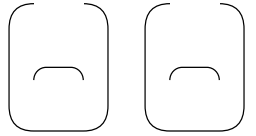


"The image contains two socks that are entirely outlined in black, with the top portion of each sock featuring white stripes. These white stripes are located near the upper edges of the socks, where they would normally fit around the calf. The socks are positioned parallel to each other, with a slight gap between them. The toe areas of both socks are indicated by gently curved lines bulging outward, suggesting the shape of toes. The entire illustration is set against a plain white background, emphasizing the stark contrast between the black outlines and the white background. There are no additional colors, textures, or decorative elements present in the image."

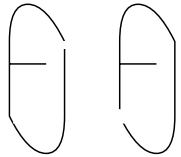
Single Turn (No Iterative Refinement)



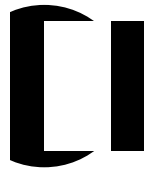
Unstructured Refinement



Semi-Structured Refinement



Structured Refinement



Evaluating Iterative Refinement

Model: Qwen2.5-VL-72B-Instruct, Dataset: MMSVG-2M subset (100 samples)

	Compilation Rate	Components	CLIP-Text	CLIP-Image	DINO
Single Turn Generation	40%	3.83	0.113	0.262	0.244
Iterative Refinement: Structured Actions	16%	1.67	0.041	0.105	0.010
Iterative Refinement: Semi-Structured Actions	81%	6.69	0.237	0.534	0.520
Iterative Refinement: Unstructured Actions	90%	9.42	0.271	0.618	0.599

Next Steps

- **Setup vLLM** library for faster local inference
- Generate SVGs based using **MMSVG-Bench** set with **multiple models**
  - (300 Prompts) x (Qwen 3B, 7B, 14B, 72B)
  - (30 Prompts) x (GPT-5, Sonnet-4)
- Refine system prompts for semi/fully structured Actions
- Add / revise **metrics** of image-prompt alignment
  - SSIM
  - LPIPS
- Finalize qualitative + quantitative **paper results**