CSE 575 Project Prototype

Ashish Raj Shekhar
Arizona State University
1223100216

Dhruv Jain
Arizona State University
1222337324

Hoang Vu Tran
Arizona State University
1222215475

Ji Seong Han
Arizona State University
1234800231

Abstract

Text-to-image generation is emerging as a pivotal area in the pursuit of Artificial General Intelligence (AGI), driven by recent breakthroughs in models such as CLIP, Stable Diffusion, and DALL·E. Among the growing subfields, SVG generation presents a unique challenge due to its requirement for precise structural and aesthetic detail. In this work, we focus on fine-tuning open-source large language models—including Gemma, LLaMA, and Mistral—on the starvector/text2svg-stack dataset, with the goal of generating high-fidelity and semantically accurate SVG representations from textual input. To assess model performance, we utilize the CLIP score as a benchmark for semantic alignment between text and generated image. Our approach advances the capabilities of language-vision models in structured vector graphic generation and contributes to the broader goal of bridging the gap between text understanding and visual synthesis.

1 Introduction

1.1 Problem Statement

Generating SVGs from natural language is challenging due to the need for structural precision and semantic accuracy. Existing approaches often yield syntactically invalid or semantically misaligned outputs. To address this, we fine-tune LLMs to generate valid and meaningful SVGs using a curated dataset and CLIP-based evaluation.

1.2 Motivation

The generalization and fine-tuning capabilities of LLMs have revolutionized the way we develop models for any task in the field of machine learning or deep learning. We can leverage the capabilities of pre-trained models and add additional layers that learn from our training data with methods like PEFT and Lora. We expect to achieve SVG code that can combine the objects mentioned in the text description with accurate placements that can be used for scientific visualization and digital illustrations.We aim to create high-quality, and semantically accurate SVG images that are comparable to human generated images. The model will also be used for the Drawing with LLMs competition being hosted on Kaggle where the aim is to generate SVG code from text abstract, fashion, and landscape domain.

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

2 Related Work

There has been increasing interest in the research community in evaluating the capabilities of large language models (LLMs) for various tasks such as classification, segmentation, clustering, synthesis, and decomposition. One particular area of growing focus is image generation, especially for graphics and art-related applications. The emergence of models like Stable Diffusion [1], DALL·E [2], and Imagen [3] has made text-to-image generation accessible to graphic designers, artists, and the general public.

2.1 Scalable Vector Graphics

Scalable Vector Graphics (SVGs) are widely used to represent logos and graphical components due to their resolution independence, small file size, and ease of manipulation across platforms. Earlier approaches have employed RNNs and MLPs trained on pre-defined SVG commands and rules to generate vector graphics [4].

Recent works such as DiffSVG [5], VectorFusion [6], DiffSketcher [7], and Chat2SVG [8] use differentiable rasterization techniques to achieve state-of-the-art (SOTA) performance on CLIP-based benchmarks and vector graphic quality metrics such as VPSD[9]. These models, however, suffer from several limitations including incoherent element placement, inconsistent background integration, context mismatch, limited editability, and syntactic errors in the generated SVG code.

Our method aims to address these challenges by leveraging the generalization capabilities of open-source pretrained LLMs and fine-tuning them on a curated text-to-SVG dataset. The resulting SVG output is expected to be both syntactically and semantically accurate while satisfying the constraints and styles specified in the user's prompt or instruction.

3 Methodology and Progress

3.1 Approach

The training has been done using the starvector/text2svg-stack, which contains about 2.1 million rows of caption and SVG code and is about 3.34 GBs in size. There are captions generated by BLIP, llava, and cogvlm. For our task, as we want to create a model that can enhance details from a short prompt, we are using the Caption generated by BLIP2.

For this task, we have started fine-tuning these models: T5-Small, T-5 Base, Mistral, Gemma 1b, Gemma 4b, Llama 7b, Qwen2.5 Coder 3b, Qwen2.5 Coder 7b, and Qwen2.5 Coder 14b. We used PEFT and QLora methods for fine-tuning, considering our hardware limitations.

Our purpose is to find the best models that can correctly create an SVG code from the given caption, then calculate its similarity compared to the ground truth given in the dataset.

3.2 Methodology

To fine-tuning a lists of large models for the task of SVG code generation, we first decided on a pipeline of these steps:

3.2.1 Data processing

For this task, we assume that the longer SVGs are more complex and informative than their shorter counterparts. From the starvector/text2svg-stack dataset, we selected the top 1,000 samples with the longest SVGs string. For some models, we then normalized each SVG string to ensure consistency across the board. After shuffling the data, we split it into 80% training and 20% validation subsets.

3.2.2 Prompt Engineering

We have conducted extensive research to extract precise outputs from LLMs, since recent works have proved that certain prompting structures and formats activate more parameters of the model, resulting in coherent and context-aware outputs that align with the users' expectations. The earliest works include COT, Few-shot COT, and EE, being the most popular ones. [10] For our task, we are experimenting with a guided structure that focuses on defining the problem statement, the expected outcome, and the constraints and limitations that our output should adhere to.

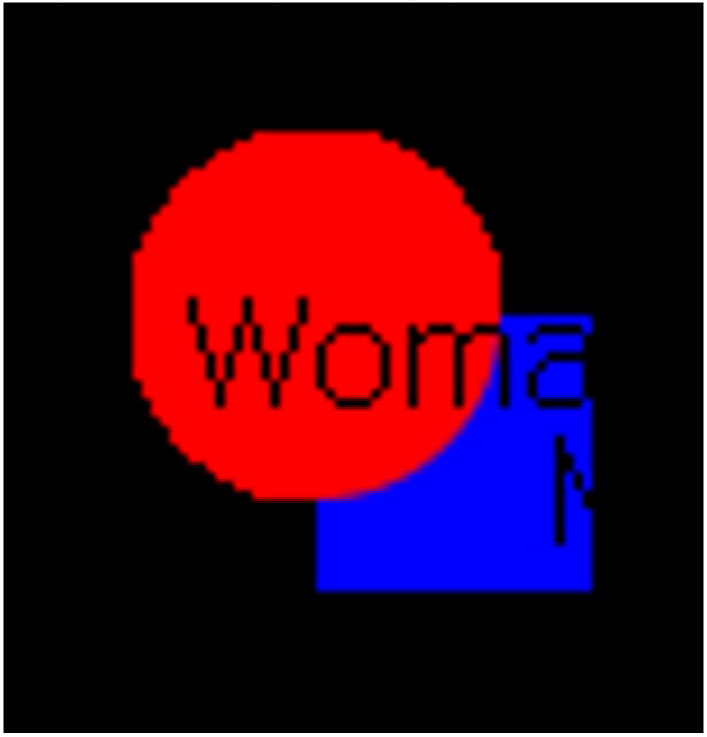
3.2.3 Tokenizing

In order to create token that can be fed into the models, we construct inputs following the conversational format. Each sample contains a system prompt that gives the model the information it needs to create an SVG code with the basic context around the SVG descriptions. The system prompt was followed by a user prompt that describes the SVG description caption given by the original dataset.

3.2.4 Fine-tuning

Using the inputs that we acquired using the previous steps, we trained several models and adapting them with full fine-tuning or parameter-efficient techniques like QLoRA and PEFT. The time to train different models, their batch size, and the memory usage vary greatly depending on the size of each model.

Figure 1: SVG for a logo for a company called Man Woman



3.3 Loss across different models

The losses while training the aforementioned models is displayed on this table:

Table 1: Training Loss across different models

Model	Model Parameter Size	Training Loss
T5 Base	220M	$\sim$2.062
T5 Small	60M	$\sim$0.970
Mistral	7B	$\sim$0.616
Gemma3	1B	$\sim$0.784
Gemma3	4B	$\sim$1.027
Llama2	7B	$\sim$0.295
Qwen2.5 Coder	3B	$\sim$0.814
Qwen2.5 Coder	7B	$\sim$1.456
Qwen2.5 Coder	14B	$\sim$3.957

4 Plans Ahead

4.1 Plans

So far, we have finalized the dataset and decided on a pipeline for pre-processing. We have also tested several models including T5, LLaMA, and Mistral.

Next, we will finalize the model architecture to be trained during the next week and start training it on SOL over small parts of the dataset and limited epochs. We will save checkpoints and evaluate them to find the best model. We plan to evaluate the model on clip scores.

For the final presentation we will present both a live demo where we enter random text prompts and view the corresponding SVG output in real time along with a summary of methods used and training parameters.

4.2 Challenges and adjustments

While we stayed on track with our original proposal, we encountered several challenges. The biggest challenge was hardware limitations, causing restricted access to GPUs, memory, and processing constraints thus slowing our progress. The dataset required some major pre-processing as well due to its massive size.

Each model required us to write a new code, tune it for the right batch size, modify the input-output template all requiring extensive trial and error iterations. Prompting strategies and background research also took longer than expected.

Despite these setbacks, we have made major progress and should be able to deliver a functional demo with promising results by the end.

References

[1] Rombach, R., Blattmann, A., Lorenz, D., Esser, P., & Ommer, B. (2022) High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR).*

[2] Ramesh, A., Pavlov, M., Goh, G., Gray, S., Voss, C., Radford, A., Chen, M., & Sutskever, I. (2021) Zero-shot text-to-image generation. In *International Conference on Machine Learning (ICML).*

[3] Saharia, C., Chan, W., Saxena, S., Li, L., Whang, J., Denton, E., Salimans, T., Ho, J., & Fleet, D. (2022) Photorealistic text-to-image diffusion models with deep language understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR).*

[4] Ha, D., & Eck, D. (2017) A neural representation of sketch drawings. *arXiv preprint arXiv:1704.03477.*

[5] Li, T., Yumer, E., Ritchie, D., & Yu, F. (2019) Differentiable vector graphics rasterization for editing and learning. In *ACM SIGGRAPH Asia*.

[6] Zhang, Y., Liu, X., Bau, D., & Zhu, J.-Y. (2022) VectorFusion: Text-to-SVG by abstracting pixel-based diffusion models. *arXiv preprint arXiv:2211.13169*.

[7] Wang, Z., Zhao, Y., Xu, Y., & Lu, J. (2023) DiffSketcher: Text-guided vector sketch synthesis through latent diffusion models. *arXiv preprint arXiv:2303.10685*.

[8] Chen, Z., Xu, Y., Li, B., Chen, Y., & Yu, F. (2023) Chat2SVG: Bridging LLMs and differentiable rasterizers for interactive vector graphic design. *arXiv preprint arXiv:2307.06955*.

[9] Xing, X., Yu, Q., Wang, C., Zhou, H., Zhang, J., & Xu, D. (2024) SVGDreamer++: Advancing Editability and Diversity in Text-Guided SVG Generation. *arXiv preprint arXiv:2411.17832*.

[10] Schulhoff, S., Ilie, M., Balepur, N., Kahadze, K., Liu, A., Si, C., Li, Y., Gupta, A., Han, H., Schulhoff, S., Dulepet, P.S., Vidyadhara, S., Ki, D., Agrawal, S., Pham, C., Kroiz, G., Li, F., Tao, H., Srivastava, A., Da Costa, H., Gupta, S., Rogers, M.L., Goncearenco, I., Sarli, G., Galynker, I., Peskoff, D., Carpuat, M., White, J., Anadkat, S., Hoyle, A., & Resnik, P. (2024) The Prompt Report: A Systematic Survey of Prompt Engineering Techniques. *arXiv preprint arXiv:2403.05697*.