VectorEdits: A Dataset and Benchmark for Instruction-Based Editing of Vector Graphics

Josef Kuchař[1], Marek Kadlčík[1], Michal Spiegel[1,2], Michal Štefánik[1,3]
[1]TransformersClub @ Faculty of Informatics, Masaryk University
[2]Kempelen Institute of Intelligent Technologies
[3]R&D Centre for Large Language Models, National Institute of Informatics, Japan

Abstract

We introduce a large-scale dataset for instruction-guided vector image editing, consisting of over 270,000 pairs of SVG images paired with natural language edit instructions. Our dataset enables training and evaluation of models that modify vector graphics based on textual commands. We describe the data collection process, including image pairing via CLIP similarity and instruction generation with vision-language models. Initial experiments with state-of-the-art large language models reveal that current methods struggle to produce accurate and valid edits, underscoring the challenge of this task. To foster research in natural language-driven vector graphic generation and editing, we make our resources created within this work publicly available.

Keywords: SVG, vector graphics, dataset, benchmark, editing, multimodal learning, image transformation, CLIP similarity, vision-language models, generative AI, image editing evaluation, vector representation learning, instruction-based image editing, visual reasoning

1. Introduction

Vector graphics play a crucial role in modern digital content creation, enabling scalable, resolution-independent visual elements across web, print, and user interface design. Unlike raster images, vector graphics are composed of geometric primitives such as paths, curves, and shapes, making them highly editable and efficient for a wide range of design tasks.

In this work, we focus on the task of instruction-guided vector image editing—modifying a vector graphic based on a natural language instruction. This is a challenging problem that requires a model to combine multiple advanced capabilities: visual understanding to interpret the source image, spatial reasoning to identify and localize elements described in the instruction, and code generation to produce valid and semantically meaningful SVG edits.

Beyond its technical complexity, this task has meaningful practical implications. Progress in instruction-based SVG generation and editing could significantly lower the barrier to entry for digital creativity, enabling the free creation, customization, and sharing of vector-based digital art. Such systems could support both novice users and professionals in rapidly prototyping, adapting, and remixing designs through simple language commands.

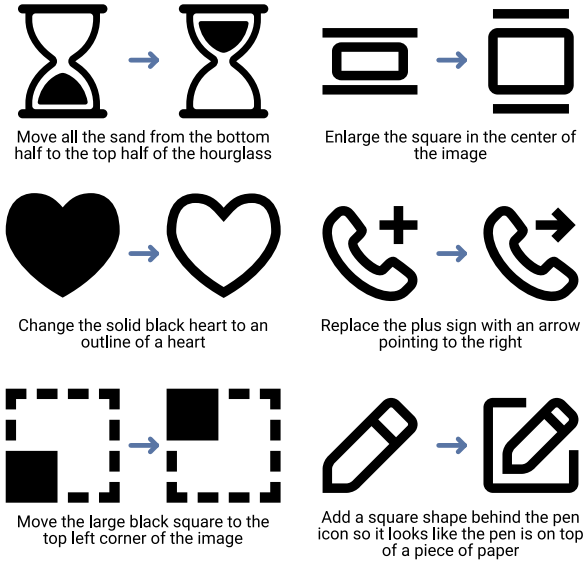

Figure 1: Examples of instruction-guided vector image editing pairs from our dataset.

2. Background

Several recent efforts have explored vision-language tasks in the vector domain. Datasets like SVGBench (Rodriguez et al., 2023) and VG-Bench (Zou et al., 2024) provide paired SVG and caption data to support tasks such as image generation from text and vice versa. While these resources enable research into SVG generation and representation learning, they are not designed for editing tasks where a transformation between two vector images must be described and executed.

Instruction-based image editing has seen rapid progress in the raster domain, with models like InstructPix2Pix (Brooks et al., 2022) enabling photo-realistic editing based on natural language prompts.

Proceedings of the Fifteenth Language Resources and Evaluation Conference (LREC 2026), pages 11119–11124
11–16 May 2026.

These approaches leverage large-scale datasets and diffusion models to modify pixel-based images.

Our work presents a new, challenging dimension of image editing by introducing a large-scale dataset for instruction-guided vector image editing, where the goal is to transform one SVG into another using a natural language instruction. Unlike prior work focused solely on generation or simple attribute edits, our dataset is structured around realistic transformations drawn from curated vector collections, supporting both fine-grained evaluation and more sophisticated model development.

The task most closely related to instruction-based editing is text-to-SVG generation, which produces a vector graphic directly from a textual description. This problem has gained attention in recent years as a counterpart to text-to-image generation for raster data. Methods such as CLIPDraw (Frans et al., 2022) and DeepSVG (Carlier et al., 2020) demonstrate that text supervision via large-scale vision–language models can guide the optimization of differentiable vector primitives, enabling the synthesis of semantically meaningful vector scenes. Building upon these ideas, SVGDreamer (Xing et al., 2024) and IconShop (Wu et al., 2023) introduce diffusion and transformer-based architectures capable of generating structured, editable SVGs from textual prompts. Despite these advances, text-to-SVG generation remains limited in its ability to modify existing graphics, as it typically synthesizes new content from scratch rather than performing targeted edits that preserve the original structure or style.

While SVGEditBench (Nishina and Matsui, 2024) explores instruction-based SVG editing, it focuses on six narrowly defined transformation types. In contrast, our dataset covers a much broader and more diverse set of edits, offering a more generalized benchmark for studying instruction-conditioned vector editing at scale.

Developing resources for instruction-based vector editing opens new possibilities for evaluating how well models comprehend, transform, and preserve the compositional structure of graphical content.

3. Dataset

Our dataset consists of 271,306 pairs of vector images, each accompanied by a natural language instruction that describes the transformation from the original image to its edited version. Each pair contains a source vector graphic, a target vector graphic, and a corresponding instruction detailing the editing operation required to achieve the transformation. This structure is designed to support training and evaluation of models for instruction-guided vector image editing. Example pairs can be seen in Figure 1.

To facilitate robust evaluation, we split the dataset into three subsets:

- Training set: 269,106 pairs
- Validation set: 200 pairs
- Test set: 2,000 pairs

To minimize data leakage, the splits were constructed by selecting entire individual collections. This approach reduces the likelihood that related vector styles or motifs appear across different subsets, supporting a more reliable evaluation of generalization to unseen styles and image structures.

This dataset is derived from SVG Repo (svg), a large repository of free and open-source vector graphics that includes icons as well as various other high-quality vectors such as illustrations and design elements. Pairs of vector images were created by sampling within individual collections, which group graphics sharing a consistent visual style. Although images within a collection are not variations of the same design, their stylistic similarity provides a coherent context for generating natural language instructions that describe transformations between pairs.

SVG Repo serves as a proxy distributor of open-licensed vector content, indexing works from various public domain, open source, and user-submitted sources. Most content falls under the SVG Repo License, which permits sharing and adaptation without attribution, although credit is appreciated. In cases where specific licenses apply (e.g., MIT, GPL, CC BY), the corresponding terms are respected. All content was used in accordance with these licensing conditions.

In the following sections, we describe each phase in the dataset creation.

3.1. Image Pair Sampling

For an image editing task to be meaningful, the source and the target image should not only maintain overall style, but also share semantic components.

To sample sensible image pairs, we compared all images within each collection by computing the cosine similarity between their CLIP (Radford et al., 2021) embeddings, extracted from the rendered bitmaps. By computing pairwise similarity scores between images, we selected pairs with a similarity above a defined threshold, ensuring that the transformations are meaningful yet diverse. This process was applied only within the same collection to maintain stylistic consistency while capturing a range of visual changes suitable for instruction-based editing tasks.

In addition to CLIP, we also tested other similarity and clustering methods, including DINOv2 (Oquab et al., 2023) visual embeddings, TF-IDF on image tags and generated captions, and pixel MSE between rasterized versions of the SVGs.

For each method, thresholds were tuned by manually going from most to least similar pairs in 30 separate collections to identify first pair that does not represent a meaningful editing transformations according to human judgment. The variability of threshold across collections for each method is illustrated in Figure 2. CLIP was ultimately chosen due to its lowest threshold deviation, making it the most reliable and consistent approach.

Figure 3 documents how the manually annotated set of last relevant and first non-relevanft pairs overlap in terms of chosen similarity measure. We select the threshold so that there are no false positives marked by the similarity on manually annotated set, i.e., above the most similar non-relevant pair.

After the similarity measure and threshold were established, we identified a total of 271,306 image pairs.

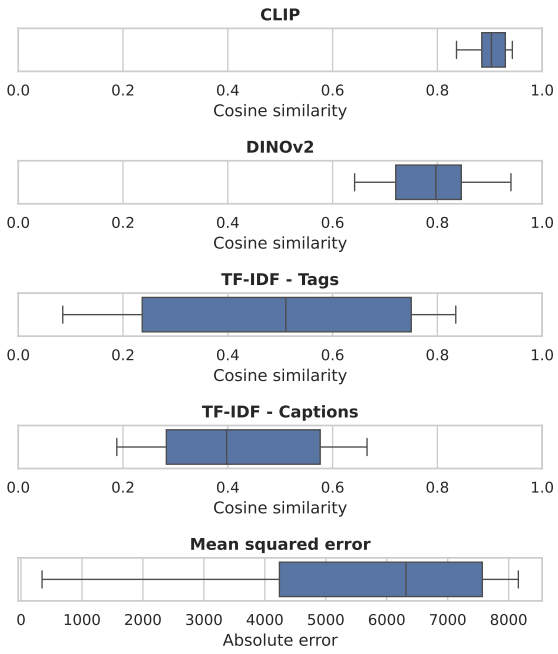

Figure 2: Distribution of manually selected similarity thresholds across 30 samples for five different pairing methods. Lower variance indicates more consistent and reliable pair selection.

3.2. Instruction Generation

With the image pairs selected, the next step was to generate natural language instructions that describe the visual transformation from one image to the other.

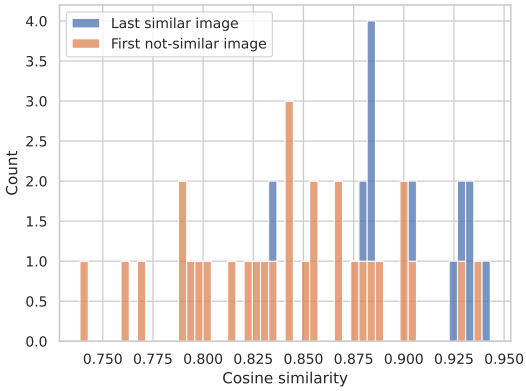

Figure 3: Histogram of CLIP similarity scores for the last image marked as similar and the first marked as not similar in a set of manually reviewed examples.

Given the large scale of the dataset, edit instructions for individual pairs were generated automatically using vision-language models. We evaluated three different models for this task by manually ranking the quality of their generated instructions on a sample of 100 pairs.

Based on this evaluation, the models were ordered from best (1) to worst (3) in terms of instruction accuracy and relevance. The results of the ranking are shown in Table 1.

Model	Mean rank
OpenAI GPT 4.1	1.28
Llama 4 Maverick	1.54
Qwen2.5-VL 70B	1.51

Table 1: Human ranking of labeling performance across three vision-language models. Lower mean rank indicates higher quality edit instructions, based on a manual evaluation of 100 image pairs.

Although Qwen2.5-VL 70B was not ranked as the best-performing model in our manual evaluation, it was ultimately selected for generating instructions across the full dataset. This decision was driven by practical considerations: Qwen2.5-VL 70B can be hosted locally, significantly reducing the cost and dependency associated with using API-based models like GPT-4.1 while supporting reproducibility and further development.

4. Evaluating models

Since instruction-guided vector image editing is a novel task with no specialized models available, we evaluate general pretrained large language models (LLMs) to establish a baseline performance on our dataset. In our setup, the initial SVG image and the corresponding edit instruction are provided as

Model	CLIP (↑)	DINOv2 (↑)	MSE (↓)	Invalid count (↓)
Baseline – no edit	0.9634	0.9011	10488	0
Baseline – white image	0.7188	0.2637	17262	0
GPT-4o mini	0.9040	0.8058	8526	14
Gemini 2.0 Flash	0.9089	0.8135	11810	32
Llama 4 Maverick	0.9094	0.8331	9627	13
Gemma 3 27B	0.9105	0.8268	11881	66
DeepSeek V3 0324	0.9203	0.8444	11194	278

Table 2: Comparison of CLIP, DINOv2 similarity, MSE distance, and invalid SVG counts across models. *Baseline – no edit* uses the original unedited image as the output, while *Baseline – white image* uses a blank white image. ↑ – higher is better; ↓ – lower is better

input prompts to the model, which is then tasked with generating the edited SVG output.

4.1. Metrics

To evaluate model performance, we first rasterize both the generated and reference SVGs to 512×512 pixel images to enable visual comparison. We then use several metrics including Mean Squared Error (MSE), DINOv2 similarity, and CLIP score to compare the generated images against ground truth edited images. Following the findings of the Starvector (Rodriguez et al., 2023), we emphasize that semantic similarity metrics like DINOv2 and CLIP better align with human judgments of output quality than purely pixel or geometry-based measures such as MSE. Additionally, we track the count of invalid SVG outputs, which are generated files that do not conform to SVG syntax or semantics, as a critical measure of model reliability. Thus, these metrics provide a more meaningful assessment of how well models understand and apply the editing instructions.

4.2. Results

We evaluate model performance against two baselines to contextualize results. The first, *Baseline – no edit*, uses the original, unedited SVG image as the model output, treating it as if the edit had been applied. The second, *Baseline – white image*, uses a completely white image as the predicted edit. These baselines help illustrate the difficulty of the task and the strength of naive strategies.

All tested models performed significantly below expectations, failing to surpass the baseline strategy of leaving the original image unedited. Despite being prompted with both the initial SVG and a natural language instruction, models frequently produced outputs that were semantically incorrect or unrelated to the intended transformation. Most generated SVGs were syntactically valid and rendered successfully, as shown in Table 2, but they often failed to capture the intended change. This reveals a gap between structural validity and semantic understanding: models can produce correct SVG syntax yet misinterpret or only partially apply the desired edit.

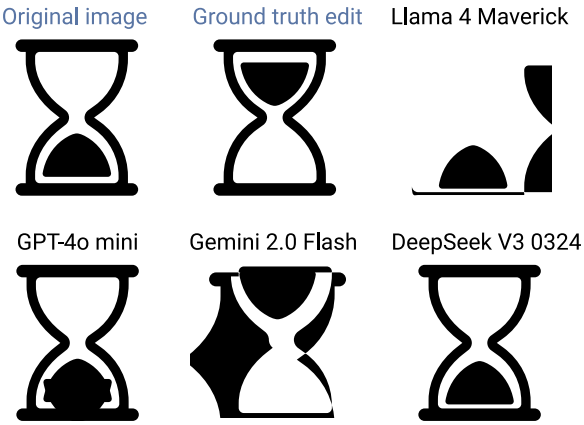

Figure 4: Failed edit attempts by several models on the hourglass example. The instruction was: **Move all the sand from the bottom half to the top half of the hourglass.**

In contrast, the *Baseline – no edit*, where no changes were made to the input image, achieved higher similarity scores (CLIP, DINOv2) relative to the generated outputs. These results, summarized in Table 2, highlight the difficulty of the task and the current limitations of existing models in understanding and executing fine-grained vector editing instructions. A representative example of failed attempts is illustrated in Figure 4, where several models struggle to correctly execute a simple sand movement instruction on an hourglass image.

5. Conclusions

We introduced a large-scale dataset for instruction-guided vector image editing, filling a gap in existing research focused primarily on raster images. Our dataset enables the development and evaluation of models that interpret natural language instructions to perform structured edits on vector graphics, highlighting the compositional and symbolic nature of vector data.

Initial experiments with state-of-the-art models show that this task remains highly challenging, with current approaches struggling to generalize and often failing to outperform a simple no-edit baseline. These results underscore the need for methods that better capture the hierarchical and parametric structure of vector representations.

As future work, we plan to fine-tune models on this dataset to improve their understanding of SVG structure and their ability to execute precise, instruction-based transformations. We hope that our dataset will encourage and support further research into language-driven vector manipulation and the development of models better suited to structured visual domains.

Limitations

Although our dataset provides a valuable resource for instruction-guided vector editing, it has some limitations. The clustering approach based on similarity thresholds, although effective, may introduce errors and may occasionally pair images with less meaningful or ambiguous transformations, despite the fact that our methodology sets the thresholds such that the risk of these errors is minimized. Additionally, the automatic generation of edit instructions using vision-language models may occasionally introduce noise and inconsistencies, as these models are not perfect in understanding all visual differences. Another limitation is that some instructions may contain excessive detail, effectively allowing a model to generate the "edited" SVG from scratch without relying on the source image. Future work could focus on refining clustering methods and improving instruction quality through human-in-the-loop verification or more advanced labeling techniques.

Data and Code Availability

The VectorEdits dataset introduced in this work was created by the authors and is made publicly available for research purposes. The dataset can be accessed on Hugging Face at https://huggingface.co/datasets/mikronai/VectorEdits. Furthermore, the code for the dataset creation and model benchmarking is available on GitHub at https://github.com/JosefKuchar/vector-edits.

Bibliographical References

SVG Repo - free svg vectors and icons — svgrepo.com. https://www.svgrepo.com/. [Accessed 19-05-2025].

Tim Brooks, Aleksander Holynski, and Alexei A. Efros. 2022. Instructpix2pix: Learning to follow image editing instructions.

Alexandre Carlier, Martin Danelljan, Alexandre Alahi, and Radu Timofte. 2020. Deepsvg: A hierarchical generative network for vector graphics animation. In *Advances in Neural Information Processing Systems*, volume 33, pages 16351–16361. Curran Associates, Inc.

Kevin Frans, Lisa Soros, and Olaf Witkowski. 2022. Clipdraw: Exploring text-to-drawing synthesis through language-image encoders. In *Advances in Neural Information Processing Systems*, volume 35, pages 5207–5218. Curran Associates, Inc.

Kunato Nishina and Yusuke Matsui. 2024. Svgeditbench: A benchmark dataset for quantitative assessment of llm's svg editing capabilities.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mahmoud Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. 2023. Dinov2: Learning robust visual features without supervision.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. Learning transferable visual models from natural language supervision.

Juan A. Rodriguez, Abhay Puri, Shubham Agarwal, Issam H. Laradji, Pau Rodriguez, Sai Rajeswar, David Vazquez, Christopher Pal, and Marco Pedersoli. 2023. Starvector: Generating scalable vector graphics code from images and text.

Ronghuan Wu, Wanchao Su, Kede Ma, and Jing Liao. 2023. Iconshop: Text-guided vector icon synthesis with autoregressive transformers.

Ximing Xing, Haitao Zhou, Chuang Wang, Jing Zhang, Dong Xu, and Qian Yu. 2024. Svgdreamer: Text guided svg generation with diffusion model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4546–4555.

Bocheng Zou, Mu Cai, Jianrui Zhang, and Yong Jae Lee. 2024. VGBench: Evaluating large language models on vector graphics understanding and

generation. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 3647–3659, Miami, Florida, USA. Association for Computational Linguistics.