SVGauge: Towards Human-Aligned Evaluation for SVG Generation

Leonardo Zini[1], Elia Frigieri[1], Sebastiano Aloscari[1], Marcello Generali[2], Lorenzo Dodi[2], Robert Dosen[2], Lorenzo Baraldi[1]

[1] University of Modena and Reggio Emilia
[2] Doxee S.p.A.

Abstract. Generated Scalable Vector Graphics (SVG) images demand evaluation criteria tuned to their symbolic and vectorial nature: criteria that existing metrics such as FID, LPIPS, or CLIPScore fail to satisfy. In this paper, we introduce SVGauge, the first human-aligned, reference-based metric for text-to-SVG generation. SVGauge jointly measures (*i*) visual fidelity, obtained by extracting SigLIP image embeddings and refining them with PCA and whitening for domain alignment, and (*ii*) semantic consistency, captured by comparing BLIP-2-generated captions of the SVGs against the original prompts in the combined space of SBERT and TF-IDF. Evaluation on the proposed SHE benchmark shows that SVGauge attains the highest correlation with human judgments and reproduces system-level rankings of eight zero-shot LLM-based generators more faithfully than existing metrics. Our results highlight the necessity of vector-specific evaluation and provide a practical tool for benchmarking future text-to-SVG generation models.

Keywords: Human Evaluation Metric · Text-to-SVG Generation · Scalable Vector Graphics

1 Introduction

The objective of text-to-SVG generation is to produce vectorial graphical representations conditioned on natural language prompts, accurately conveying the intended semantics while respecting the stylistic and structural properties of SVGs. Unlike raster images, SVGs offer a symbolic, abstract, and resolution-independent representation of visual concepts, often emphasizing geometric structures and minimalistic design choices. As such, the task demands not only understanding the textual description but also capturing fine-grained visual abstractions and compositional relationships between elements.

Recent advances in generative modeling have considerably improved the quality of SVG synthesis, with innovative strategies including rasterization-then-vectorization pipelines [13], differentiable vector renderers [12], and latent diffusion frameworks adapted to the vector space [8, 23, 25]. Additionally, there has been a growing interest in leveraging Large Language Models (LLMs) for autoregressive SVG token generation [18, 20, 21, 24], harnessing the human-readable nature of SVG markup for more interpretable synthesis.

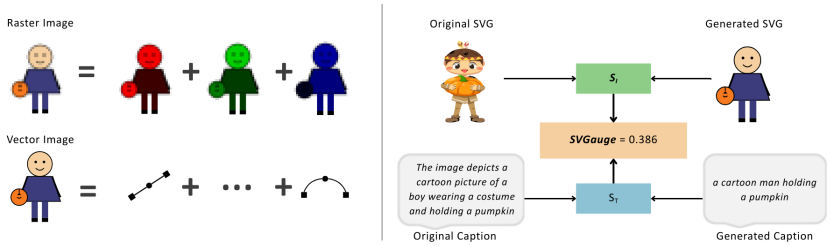

Fig. 1: Comparison between raster (*e.g.*, JPG) and vector images (*e.g.*, SVG) and an overview of SVGauge for SVG generation evaluation.

As generation quality continues to progress, the need for effective evaluation becomes more critical. Current evaluation methods largely rely on metrics designed for raster images, such as Fréchet Inception Distance (FID) [6], Learned Perceptual Image Patch Similarity (LPIPS) [27], or DINOv2-based similarities [14], or on text-image alignment scores like CLIPScore [5]. However, these approaches often fall short when applied to SVGs: they fail to capture the symbolic, geometric, and stylistic nuances intrinsic to vector graphics, and can exhibit strong sensitivity to low-level changes despite semantic equivalence.

In light of these limitations, we argue that evaluating text-to-SVG generation requires dedicated metrics that specifically account for the distribution of SVG images and align closely with human perceptual and semantic judgment. In response, we propose SVGauge, a novel metric for assessing the quality of text-to-SVG generations, designed to robustly capture both visual similarity in a vector-aware embedding space and semantic preservation via multimodal text analysis (Fig. 1). Our approach begins by rasterizing both the reference and generated SVGs to enable feature extraction via pre-trained vision backbones, followed by domain adaptation techniques including PCA and whitening. Visual similarity is computed by comparing embeddings in this adapted space, while semantic consistency is evaluated through a captioning loop that uses a multimodal LLM to generate descriptions of the SVGs, which are then compared using a combination of Sentence-BERT embeddings and TF-IDF-weighted similarity.

Extensive experiments demonstrate that SVGauge achieves superior alignment with human judgment across a wide range of SVG generation scenarios, outperforming conventional raster-based metrics. We show that our dual-axis design provides robustness and increased alignment with human evaluations, better reflecting the qualities valued in SVG synthesis. As a complementary contribution, we also develop and release the first dataset with prompt-SVG pairs annotated with human evaluations.

2 Related Works

Here, we provide a concise overview of the most relevant works related to SVG generation using Latent Diffusion Models or LLMs, and quality measurement.

Latent Diffusion Models. A common approach to SVG generation synthesizes a raster image from a text prompt using latent diffusion models, followed by vectorization via traditional tools like LIVE [13], VTracer, or Potrace. DiffVG [12] introduced a differentiable renderer enabling SVG optimization through backpropagation with image-based losses such as Score Distillation Sampling (SDS). VectorFusion [8] initializes an SVG with fixed paths and optimizes it using latent SDS gradients, while SVGDreamer [25] improves initialization using activation maps but at the cost of a complex pipeline. Recently, SVGFusion [23] proposed a unified architecture based on a Vector-Space Diffusion Transformer and a Vector-Pixel Autoencoder to improve both efficiency and quality.

Large Language Models. The structured nature of SVGs has also motivated LLM-based generation approaches. IconShop [21] trained a Transformer decoder to produce simple icons from text, while LLM4SVG [24] built a large dataset of 250,000 auto-captioned SVGs and fine-tuned compact LLMs with instruction-tuned prompts. Chat2SVG [20] developed a prompting pipeline for SVG generation, and StarVector [18] proposed a multimodal model that translates raster images into SVGs by predicting SVG tokens with a large language model.

Evaluation Methods for SVG generation. Current evaluation methods for text-to-SVG rely on metrics developed for raster images, such as DINOv2-based similarity [14], FID [6], and LPIPS [27]. Text-image alignment metrics like CLIPScore [5] have also been employed. However, these methods can be unreliable for SVGs: being built on backbones optimized for natural images, they often fail to capture the abstract, symbolic, or geometric semantics typical of SVGs. For instance, an icon with a unique stylistic interpretation might be penalized by CLIPScore even if it faithfully represents the input caption. These limitations result in reduced alignment with human perception, which ultimately hinders the development of generative models, as they are evaluated with metrics that poorly reflect true quality and fail to reward meaningful improvements.

3 Proposed approach

We argue for moving beyond the application of metrics developed for images coming from the raster domain and instead propose a metric specifically designed for the SVG domain, carefully aligned with human judgment. Our approach takes a dual perspective, accounting for both visual similarity and semantic preservation, and demonstrates state-of-the-art alignment with human evaluations.

Overview of our approach. Our approach for text-to-SVG evaluation begins with a reference pair (T_O, S_O), where T_O represents a natural language prompt describing a visual concept, and S_O is an SVG image that accurately depicts this concept. Given a generator $\mathcal{G}$ which takes the prompt T_O as input, we evaluate the quality of its generation, $S_G = \mathcal{G}(T_O)$ through a dual-axis evaluation that captures both the visual resemblance of the generated vectorial image, and the semantic consistency between the reference and generated outputs.

Rasterization and Image Encoding. Since SVGs are vectorial representations and not directly compatible with standard image similarity techniques, we

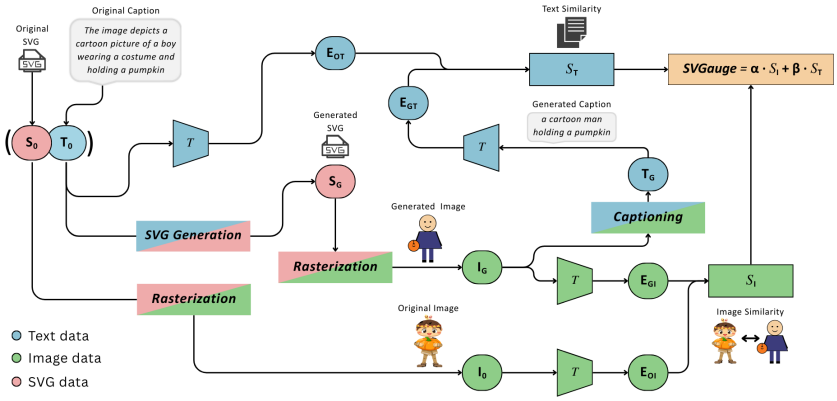

Fig. 2: Overview of the SVGauge metric for text-to-SVG generation evaluation.

first rasterize both the reference S_O and generated S_G into pixel-based images, denoted as I_O and I_G, respectively. This rasterization step enables the use of pre-trained vision backbones operating on image tensors.

Each rasterized image is passed through a visual backbone, producing high-dimensional embeddings E_{OI} and E_{GI}. To enhance spatial localization, we extract a grid of feature vectors (*e.g.*, from the final self-attention layer of a ViT backbone) and compute their average. Formally, the embedding for the reference image is defined as

$$E_{OI} = \frac{1}{H \times W} \sum_{i,j} \text{backbone}(I_O)_{[i,j]}, \tag{1}$$

where (H, W) denotes the feature grid dimensions. The embedding for the generated image is computed analogously.

To further refine these embeddings for similarity computation and adapt them to the domain of vectorial images, we apply a two-stage post-processing pipeline consisting of PCA and whitening, which have been demonstrated to improve retrieval performance in high-dimensional spaces [9]. Specifically, PCA projects each embedding onto a lower-dimensional subspace by (i) centering the data via mean subtraction, μ, computed over the distribution of vectorial images, *i.e.*,

$$X_{\text{centered}} = X - \mu,$$

where μ is the mean vector over a distribution of vector images, and (ii) projecting onto the principal components P associated with the largest eigenvalues of the covariance matrix:

$$X_{\text{PCA}} = P^{\mathsf{T}} X_{\text{centered}}.$$

This projection not only reduces noise but also emphasizes the most discriminative directions, as it ensures that the projected components are decorrelated

– *i.e.*, for any two different eigenvectors p_i and p_j, $p_i^\mathsf{T} p_j = 0$. Further, being the projection estimated on a distribution of SVG images, it also reprojects the original embeddings into a subspace which is tailored to the distribution of vectorial images.

Following PCA, we apply a whitening transformation to further normalize the embeddings. Operating in the reduced d'-dimensional subspace identified by PCA, whitening rescales each principal component to unit variance, effectively normalizing the covariance matrix to the identity. Let $\lambda_1, \ldots, \lambda_{d'}$ denote the eigenvalues associated with the selected eigenvectors. The whitening transformation is defined as

$$\hat{X} = \operatorname{diag}\left(\lambda_1^{-\frac{1}{2}}, \ldots, \lambda_{d'}^{-\frac{1}{2}}\right) X_{\mathrm{PCA}},$$

where each component is individually scaled to achieve decorrelation and variance equalization. The resulting embeddings $\hat{X}$ exhibit a balanced distribution, ensuring that no direction dominates the similarity computation and all feature dimensions contribute equally.

Visual Similarity Evaluation. Overall, the combination of averaging, PCA, and whitening yields compact and robust image representations, aligning with the principles of negative evidence and decorrelation [7, 9]. Finally, we compute the cosine similarity between the embedding of the reference and generated image, to obtain a visual similarity score S_I. The aforementioned similarity score quantifies how similar the generated image I_G is to the reference I_O in the learned embedding space [27].

Semantic Evaluation via Captions. While visual similarity provides a measure of surface-level resemblance, it may not fully capture whether the generated image semantically reflects the input prompt. An image can be structurally similar to the reference yet fail to convey the intended meaning.

To address this, we introduce a semantic evaluation loop. A multimodal LLM, such as BLIP-2 [11], is employed to generate a caption T_G for the generated image I_G, capturing its semantic content. We then compare T_G to the original prompt T_O by encoding both with Sentence-BERT (SBERT) [17], obtaining dense sentence embeddings E_{GT} and E_{OT}. Semantic similarity is initially measured via cosine similarity between these embeddings.

However, SBERT may overestimate similarity for short or generic sentences. To mitigate this, we integrate a TF-IDF weighting mechanism that emphasizes rare, informative terms. By doing so, matches on distinctive words are rewarded, while overlaps on common or generic terms are downweighted. The final semantic similarity score is computed as

$$S_T \;=\; \mathrm{CosineSim}(E_{OT},\, E_{GT}) \;\cdot\; \Big(0.8 \;+\; 0.2\,\mathrm{CosineSim}\big(V_{OT},\, V_{GT}\big)\Big), \qquad (2)$$

where E_{OT}, E_{GT} are the SBERT-space embeddings of the reference and generated texts, respectively, and V_{OT}, V_{GT} are their corresponding TF-IDF vectors.

The TF-IDF factor is normalized within the range $[0.8, 1]$ to act as a gentle rescaling term that adjusts similarity based on the content informativeness of the caption match.

Combined Evaluation Metric. Our final evaluation metric combines both visual and semantic similarities to form a unified score, defined as

$$\text{SVGauge} = \alpha \cdot S_I + \beta \cdot S_T.$$

Here, α and β are scalar weights that allow us to tune the importance of each component based on the use case. For instance, applications that prioritize aesthetic consistency may favor $\alpha > \beta$, while semantic-critical applications (*i.e.*, educational content generation) may prefer $\beta > \alpha$.

Why Both Similarities Matter. Existing metrics, primarily designed for raster images, struggle to capture the intrinsic characteristics of vector graphics. Image similarity in the projected space effectively captures local visual cues specific to vectorized representations, while textual similarity provides a global measure of semantic alignment.

For example, a text-to-SVG model may generate an SVG that differs structurally from the reference (*e.g.*, a triangle with rounded edges instead of sharp corners) but preserves the intended concept. In this case, the semantic similarity S_T would remain high, while the image similarity S_I would drop. Conversely, a model might replicate the visual structure but misrepresent the intended meaning (*e.g.*, generating a star when a flower was described), resulting in high S_I but low S_T. By combining both scores, our metric remains robust to such discrepancies and achieves closer alignment with human judgment of SVG quality and semantic fidelity.

4 Experimental Evaluation

In this section, we assess the quality and human-alignment level of the proposed metric for text-to-SVG generation. Using a newly constructed dataset consisting of SVG-prompt pairs and human ratings, we explore various methodological combinations and benchmark our metric against established metrics (LPIPS [27], DINO similarity [14], FID [7], and CLIPScore [5]). Further, we also apply our metric in system-level correlation experiments, where we assess eight LLM-based generators under zero-shot conditions.

4.1 The SHE Dataset

To assess the alignment of the proposed metric with the human judgment, we create and release the SVG Human-Evaluation dataset (SHE)[3] – a collection consisting of 333 SVG-prompt pairs, each associated with around eight generations coming from different models, evaluated with human scores. The collection

[3] The dataset will be publicly released upon paper acceptance.

Table 1: Dataset generation and user study statistics. Here, "% Generated" indicates the percentage of times that the generator returned SVG code, "% Correct syntax" the percentage of times the generated SVG code was syntactically correct, and "% Whites" the percentage of times the generated SVG code resulted in a fully white image. We also report the average human score obtained over images generated by each model.

	% Generated	% Correct Syntax	% Whites	Human Score
DeepSeek-R1-8B [3]	93.4	100.0	21.9	1.21
Llama 3.1 8B [2]	98.8	100.0	6.1	1.83
Gemma2-9B [19]	92.8	100.0	64.6	1.90
Ministral 8B	93.4	100.0	1.6	1.80
Mistral 24B	100.0	96.4	0.3	2.49
DeepSeek-R1-70B [3]	98.5	98.8	5.3	2.23
Llama 3.1 70B [2]	99.7	100.0	3.6	2.46
ChatGPT-4o [1]	100.0	100.0	0.0	3.37

of the original SVG images is initially obtained through web scraping and then manually assessed to ensure diversity in image complexity – ranging from simple black icons to elaborate illustrations. The prompts accompanying the SVG images are instead automatically generated through a state-of-the-art Multimodal LLM, Idefics3 [10], and subsequently validated through human annotation.

To collect generated SVG images for the subsequent collection of human judgment, we select eight state-of-the-art LLMs, covering both small-scale models (Gemma2-9B [19], Llama 3.1 8B [2], DeepSeek-R1-8B [3], and Ministral 8B) and larger models (ChatGPT-4o [1], Llama 3.1 70B [2], Mistral 24B, and DeepSeek-R1-70B [3]). Generated outputs that failed to produce syntactically correct SVG code were excluded from the dataset, while syntactically valid yet visually blank outputs were retained to increase diversity and fairness of evaluation in system-level evaluations. In Table 1, we report statistics on the generated results across the selected models. As can be observed, smaller LLMs typically struggle even with basic SVG image generation, irrespective of output quality, while larger models tend to provide syntactically correct SVG generations.

We then conducted a human evaluation study to collect human judgment annotations. The study involved 40 participants with diverse backgrounds, ranging from AI researchers to non-experts. Participants were asked to evaluate the correlation between input prompts and the corresponding generated SVG images on a scale from 1 (completely unrelated) to 5 (completely related). The study resulted in a total of 2,461 annotations, with an average of 57 evaluations per user. Visually blank generations were always evaluated with the lowest score. Finally, the dataset is split into training and test subsets containing 2,000 and 461 samples, respectively. Table 1 also reports the average human rating obtained by each generator.

4.2 Quantitative results

Implementation details. To assess the role of the visual backbone for computing the visual similarity score, we consider different image encoders, namely

Table 2: Average Human Correlation with different captioners and image encoders.

	CLS token		
	Florence2 [22]	**Idefics3 [10]**	**Blip2 [11]**
CLIP [16]	33.0	26.5	34.1
DINOv2 [14]	36.2	29.7	37.1
SigLIP [26]	**37.5**	**30.5**	**38.8**
	Mean Feature Grid		
MAE [4]	0.8	-5.2	-1.7
CLIP [16]	32.5	23.2	33.9
DINOv2 [14]	35.2	28.3	36.1
SigLIP [26]	**42.0**	**35.4**	**42.9**

Table 3: Effects of transformation in the features space of each image encoder, when using Blip2 [11] as captioner.

	GeM Pooling			PCA	
	$p=1$	$p=2$	$p=4$	w/o whit.	w/ whit.
MAE [4]	-1.7	30.4	31.6	32.1	32.1
CLIP [16]	33.9	34.4	34.9	39.7	39.6
DINOv2 [14]	36.1	36.1	36.2	38.7	38.5
SigLIP [26]	**42.9**	**36.3**	**36.2**	**44.1**	**44.3**

DINOv2-base [14], *MAE-vit-base* [4], *SigLIP-base-patch16-224* [26], and *CLIP-vit-base-patch32* [16]. SVG images are rasterized with a fixed white background, using the input resolutions employed by each encoder. Unless otherwise stated, we always consider the grid of activations coming from the last self-attention layer of the architecture, ignoring the `[CLS]` token. For PCA computation, we retain the first 128 largest eigenvectors. Additionally, to assess the role of the captioner when encoding the generated image, we consider three captioning models, namely *Florence2-base* [22], *Idefics3* [10], and *Blip2* [11].

To measure the correlation between the ratings of SVGauge and human evaluations, we employ the Spearman, Kendall, and Pearson correlation coefficients.

Ablation studies. We first evaluate the impact of the selected image encoder and the choice of captioning model. In Table 2, we present aggregated human correlation values, when employing feature vectors extracted from different visual encoders, and when verifying the captioner employed to generate the description of the generated image. In particular, we test by employing the `[CLS]` token of the last self-attention layer, or by taking the average of the grid of features at the last self attention layer (excluding the `[CLS]`) of CLIP, DINOv2 and SigLIP.

To represent the overall performance of each experimental combination, we aggregate the human correlation values across different values of α and β (at evenly spaced values between 0 and 1.0) and by taking the average of the Spearman, Kendall, and Pearson correlation values. This yields a single aggregated score, which is formally computed as

$$\frac{1}{C \times P} \sum_{\alpha} \sum_{\beta} S_{\alpha,\beta} + K_{\alpha,\beta} + P_{\alpha,\beta},$$

where C is 3 (the number of correlation coefficients) and P is 11 (number of α,β combinations).

As shown in Table 2, Blip2 [11] on average emerges as the most effective captioning model across all image encoders, while SigLIP [26] proves to be the most effective image encoder. Interestingly, employing the average of the feature grid always provides better results than the `[CLS]` token, underlining the effectiveness of considering spatial-aware features.

Table 4: Correlation with human judgment for SVGauge with SigLIP [26] and Blip2 [11] for different coefficients α and β. Correlation used are Spearman ($S\rho$), Kendall ($K\tau$) and Pearson (Pr). The higher the better for all correlation scores.

Coefficients		**Training set**			**Test set**		
α	β	**S**ρ	**K**τ	**P**r	**S**ρ	**K**τ	**P**r
1.0	0.0	42.6	33.1	52.2	38.2	29.3	46.3
0.9	0.1	45.4	35.2	53.6	41.8	31.9	48.3
0.8	0.2	47.5	36.7	55.0	44.8	34.2	50.4
0.7	0.3	48.8	37.8	56.1	47.3	36.2	52.2
0.6	0.4	**49.2**	**38.0**	56.8	**48.3**	**37.0**	53.7
0.5	0.5	48.8	37.7	**57.0**	**48.4**	**37.1**	54.6
0.4	0.6	47.3	36.4	56.5	47.2	36.1	**54.7**
0.3	0.7	45.0	34.6	55.1	45.1	34.4	53.8
0.2	0.8	42.3	32.3	52.8	42.3	32.3	52.0
0.1	0.9	39.1	29.8	49.7	39.4	30.1	49.4
0.0	1.0	35.8	27.1	46.0	36.5	27.7	46.3
Overall Mean		44.7	34.4	53.7	43.6	33.3	51.0

We then conduct further experiments to assess the impact of the different transformations applied to the visual feature vectors. Table 3 reports the results obtained when applying a Generalized Mean Pooling [15] (with $p = 1, 2, 4$) and PCA, with and without whitening. PCA is applied by transforming the average of the feature grid into a lower-dimensional subspace. When PCA is combined with whitening, data is centered by subtracting the mean, and each component is scaled to achieve unit variance, resulting in decorrelated features and alignment with the distribution of SVG images.

As can be noticed from Table 3, the best performance is achieved when combining PCA and whitening with SigLIP [26] as visual feature extractor, and Blip2 [11] as captioner. Table 4 further illustrates the correlation with human evaluation across different values of α and β on both training and test sets, underscoring the robust generalization capability of our proposed approach. Based on the results obtained from the training set, we select $\alpha = 0.6$ and $\beta = 0.4$ as the default values of our final metric configuration.

Comparison with the State-of-the-Art. Finally, we compare our proposed metric against established alternatives for text-to-SVG generation – namely FID [7], LPIPS [27], DINO similarity [14], and CLIPScore [5], by performing experiments both at the system level and instance level. Table 5 shows that, as per human evaluation scores, ChatGPT-4o [1] is the best-performing model, followed by Llama 3.1 70B [2] and Mistral 24B. Notably, our metric produces a model ranking that closely mirrors human judgments, assigning the highest score to ChatGPT-4o [1], followed by Mistral 24B and Llama 3.1 70B [2]. In contrast, existing older metrics such as FID [7] and LPIPS [27] either poorly correlate with human ratings or produce rankings that are not aligned with observed semantic relevance. Moreover, those metrics struggle to correctly rank generator models, which may be attributed to the absence of SVG-like data in their training distributions. It is also worth noting that lower FID scores indicate better performance, yet the average FID values observed in our experiments are

Table 5: System-level comparison between eight different generator models and different metrics.

	FID	LPIPS	DINO Sim.	CLIP Score†	Human	SVGauge ↑
DeepSeek-R1-8B [3]	656.43	0.61	0.32	0.65	1.28	**0.11**
Ministral 8B	497.95	0.56	0.40	0.70	1.86	**0.17**
Llama 3.1 8B [2]	541.41	0.59	0.38	0.66	1.66	**0.18**
Gemma2-9B [19]	800.39	0.59	0.29	0.62	1.66	**0.15**
Mistral 24B	485.95	0.56	0.40	0.71	2.31	**0.23**
DeepSeek-R1-70B [3]	502.27	0.58	0.42	0.72	2.23	**0.18**
Llama 3.1 70B [2]	458.15	0.54	0.44	0.71	2.38	**0.22**
ChatGPT-4o [1]	878.37	0.52	0.78	0.50	3.42	**0.25**

† Reference free

Table 6: Comparison of system-level correlations. Correlation between the mean of human judgements for each generator model with other metrics.

	FID	LPIPS	DINO Sim.	CLIP Score†	SVGauge
$\mathbf{S}\rho$	-7.14	-76.2	76.2	80.9	**91.0**
$\mathbf{K}\tau$	-7.14	-57.1	57.1	64.2	**83.6**
$\mathbf{P}r$	27.7	-71.6	77.9	86.3	**93.1**

† Reference free

Table 7: Instance-level correlation between CLIPScore [5] and SVGauge both in reference-free and reference based versions.

	$\mathbf{S}\rho$	$\mathbf{K}\tau$	$\mathbf{P}r$
LPIPS	-14.0	-10.7	-16.5
Dino Sim.	34.6	26.5	39.3
SVGauge Ref. based	**48.4**	**37.1**	**53.7**
CLIP Score†	**37.8**	**28.7**	40.6
SVGauge Ref. free	36.5	27.7	**46.3**

significantly high, suggesting a poor fit of this metric to the SVG domain (*e.g.*, icons, illustration, etc.).

This trend is also quantitatively confirmed in the system-level correlations reported in Table 6, where our metric achieves the highest correlation with human judgments across Spearman, Kendall, and Pearson scores. These results highlight the limitations of state-of-the-art automatic metrics in capturing the semantic alignment between SVG generated images and text prompts, and highlight the effectiveness of our proposed approach in producing evaluations that are more consistent with human perception.

In Table 7, we report instance-level correlations in terms of Spearman, Kendall and Pearson correlation coefficients for four metrics (LPIPS [27], DINO similarity [14], CLIPScore [5], and SVGauge) in both reference-based and reference-free settings. For the purpose of this experiment and fairness of evaluation, we also show a reference-free version of SVGauge, where we remove the visual similarity portion of the metric (*i.e.*, setting $\alpha = 0$). Reference-based SVGauge achieves the strongest agreement with human judgments. The reference-free variant of SVGauge trails CLIPScore [5] in ranking metrics but surpasses it in Pearson correlation. Overall, our reference-free approach offers a competitive alternative to CLIPScore [5]. These results further underline that current spread-wise evaluation methods remain unreliable when applied to SVG-based content.

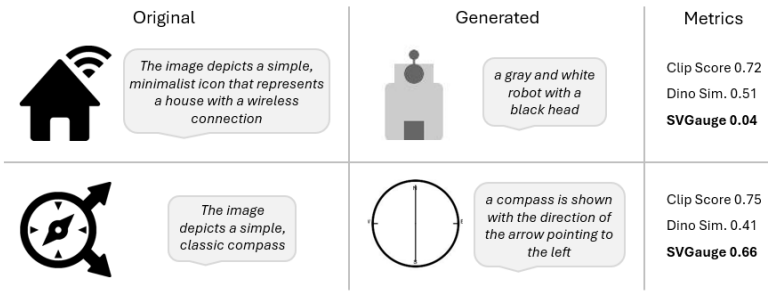

Fig. 3: Sample qualitative results.

4.3 Qualitative results

Finally, in Figure 3 we showcase a qualitative comparison between SVGauge, CLIPScore [5] and DINO similarity [14]. In the first row, although the generated SVG is visually and semantically unrelated to the reference image, CLIPScore [5] and DINO similarity [14] still give relatively high scores. In contrast, SVGauge correctly assigns a low score, better aligned with human perception. On the other hand, in the second row, when the generated SVG accurately preserves the core semantics required by the prompt, despite the lower quality image, SVGauge correctly assigns a moderate score. These examples highlight that SVGauge features an improved sensitivity to semantic consistency, and demonstrates a significant ability to distinguish between meaningful and misleading generations in comparison with other existing metrics.

5 Conclusion

We proposed the first evaluation framework specifically designed for text-to-SVG generation. By capturing both visual similarity and semantic alignment, our proposed metric SVGauge can provide quantitative evaluations with significant alignment with human perception. As a complementary contribution, we also developed the first dataset with prompt-SVG pairs annotated with human evaluations. We believe that our approach can serve as a foundation for future benchmarks and model evaluations in structured generative domains.

6 Acknowledgments

We acknowledge the CINECA award under the ISCRA initiative, for the availability of high-performance computing resources. This work has been conducted with the support of the PRIN 2022-PNRR project "MUCES" (CUP E53D23016 290001), the PRIN 2022 project "MUSMA"' (CUP E53D23008310001) and the European project MINERVA, funded by European High-Performance Computing Joint Undertaking (JU) under grant agreement No 101182737.

References

1. Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F.L., Almeida, D., Altenschmidt, J., Altman, S., Anadkat, S., et al.: GPT-4 Technical Report. In: arXiv preprint (2023)
2. Grattafiori, A., Dubey, A., Jauhri, A., Pandey, A., Kadian, A., Al-Dahle, A., Letman, A., Mathur, A., Schelten, A., Vaughan, A., et al.: The llama 3 herd of models. In: arXiv preprint (2024)
3. Guo, D., Yang, D., Zhang, H., Song, J., Zhang, R., Xu, R., Zhu, Q., Ma, S., Wang, P., Bi, X., et al.: Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. In: arXiv preprint (2025)
4. He, K., Chen, X., Xie, S., Li, Y., Dollár, P., Girshick, R.: Masked autoencoders are scalable vision learners. In: CVPR (2022)
5. Hessel, J., Holtzman, A., Forbes, M., Bras, R.L., Choi, Y.: CLIPScore: a reference-free evaluation metric for image captioning. In: EMNLP (2021)
6. Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., Hochreiter, S.: Gans trained by a two time-scale update rule converge to a local nash equilibrium. In: NeurIPS (2017)
7. Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., Hochreiter, S.: Gans trained by a two time-scale update rule converge to a local nash equilibrium. In: NeurIPS (2017)
8. Jain, A., Xie, A., Abbeel, P.: Vectorfusion: Text-to-svg by abstracting pixel-based diffusion models. In: CVPR (2023)
9. Jégou, H., Chum, O.: Negative evidences and co-occurrences in image retrieval: the benefit of PCA and whitening. In: ECCV (2012)
10. Laurençon, H., Marafioti, A., Sanh, V., Tronchon, L.: Building and better understanding vision-language models: insights and future directions. In: arXiv preprint (2024)
11. Li, J., Li, D., Savarese, S., Hoi, S.: Blip-2: bootstrapping language-image pre-training with frozen image encoders and large language models. In: ICML (2023)
12. Li, T.M., Lukáč, M., Gharbi, M., Ragan-Kelley, J.: Differentiable vector graphics rasterization for editing and learning. TOG (2020)
13. Ma, X., Zhou, Y., Xu, X., Sun, B., Filev, V., Orlov, N., Fu, Y., Shi, H.: Towards layer-wise image vectorization. In: CVPR (2022)
14. Oquab, M., Darcet, T., Moutakanni, T., Vo, H., Szafraniec, M., Khalidov, V., Fernandez, P., Haziza, D., Massa, F., El-Nouby, A., et al.: Dinov2: Learning robust visual features without supervision. TMLR (2024)
15. Radenović, F., Tolias, G., Chum, O.: Fine-tuning cnn image retrieval with no human annotation. IEEE TPAMI (2019)
16. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. In: ICML (2021)
17. Reimers, N., Gurevych, I.: Sentence-bert: Sentence embeddings using siamese bert-networks. In: arXiv preprint (2019)
18. Rodriguez, J.A., Puri, A., Agarwal, S., Laradji, I.H., Rodriguez, P., Rajeswar, S., Vazquez, D., Pal, C., Pedersoli, M.: Starvector: Generating scalable vector graphics code from images and text. AAAI (2024)
19. Team, G.: Gemma. Kaggle (2024)
20. Wu, R., Su, W., Liao, J.: Chat2SVG: Vector Graphics Generation with Large Language Models and Image Diffusion Models. In: arXiv preprint (2024)

21. Wu, R., Su, W., Ma, K., Liao, J.: IconShop: Text-Guided Vector Icon Synthesis with Autoregressive Transformers. TOG (2023)
22. Xiao, B., Wu, H., Xu, W., Dai, X., Hu, H., Lu, Y., Zeng, M., Liu, C., Yuan, L.: Florence-2: Advancing a unified representation for a variety of vision tasks. In: CVPR (2023)
23. Xing, X., Hu, J., Zhang, J., Xu, D., Yu, Q.: Svgfusion: Scalable text-to-svg generation via vector space diffusion. In: arXiv preprint (2025)
24. Xing, X., Hu, j., Zhang, L., Guotao, J., Xu, D., Yu, Q.: Empowering llms to understand and generate complex vector graphics. In: arXiv preprint (2024)
25. Xing, X., Zhou, H., Wang, C., Zhang, J., Xu, D., Yu, Q.: Svgdreamer: Text guided svg generation with diffusion model. In: CVPR (2024)
26. Zhai, X., Mustafa, B., Kolesnikov, A., Beyer, L.: Sigmoid loss for language image pre-training. In: ICCV (2023)
27. Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric. In: CVPR (2018)