Abstraction in Style

MIN LU, Shenzhen University, China
YUANFENG HE, Shenzhen University, China
ANTHONY CHEN, Peking University, China
JIANHUANG HE, Shenzhen University, China
DANIEL COHEN-OR, Tel Aviv University, Israel
HUI HUANG, Shenzhen University, China

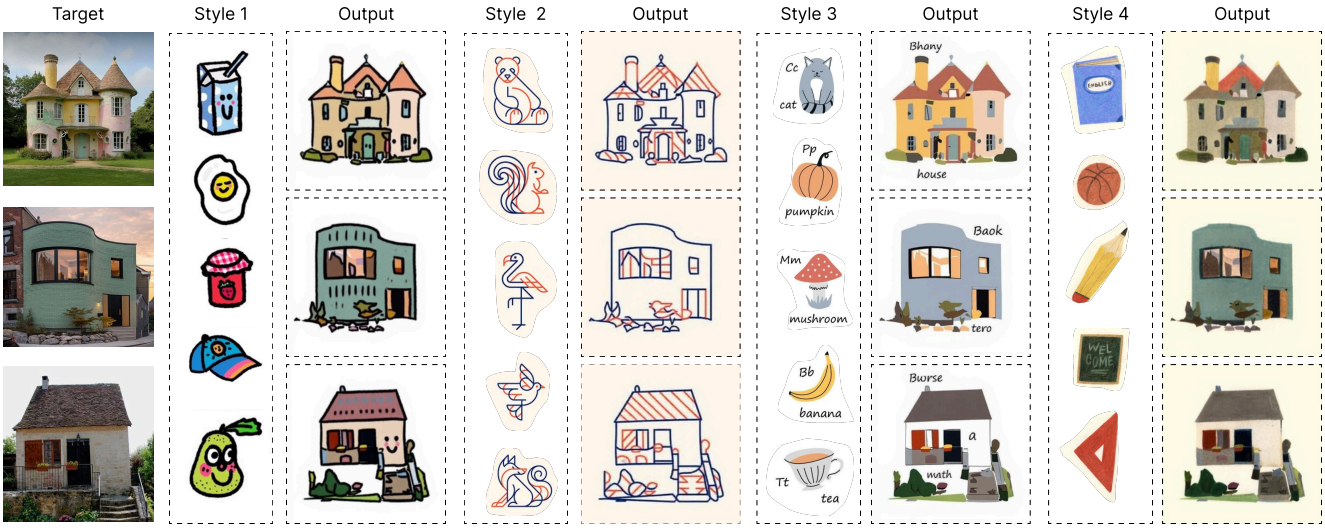

Fig. 1. Examples generated by our method: given the target images, AiS generates their abstracted versions while maintaining consistent structures.

We present Abstraction in Style, an image generative process that separates and sequences structural abstraction and visual stylization. A core premise of our work is that style and abstraction are distinct yet intertwined dimensions. Style governs visual appearance, such as stroke quality, texture, or ornamentation, while abstraction involves the reinterpretation and simplification of structure based on semantic or perceptual relevance. Unlike conventional style transfer methods that apply stylistic traits while retaining the input geometry, our approach first produces a structurally abstracted representation and then renders it in the desired style. This decoupling enables the generation of outputs that are both compositionally reimagined and stylistically coherent. By explicitly modeling abstraction alongside style, our method goes beyond conventional style transfer, supporting the generation of abstracted illustrative styles that require deeper structural reinterpretation.

1 INTRODUCTION

The generation of stylized visual content is a longstanding goal in computer graphics [8, 10, 33]. Most existing methods focus on style transfer, in which the visual traits of an artwork, such as stroke texture, color palette, or ornamentation, are applied to a target image while preserving its general underlying geometry. They transfer style in a way that adheres to the original spatial structure [37, 38], even when the target style is abstract, and therefore tend to overlook the abstraction function that may be latent in the reference style.

A core premise of our work is that style and abstraction are distinct yet intertwined dimensions. Style governs visual appearance, such as stroke quality, texture, or color, while abstraction involves reinterpretation of structure, focusing on the semantic level of the structure rather than its exact geometry and shape. Abstraction often alters the compositional essence of a structure [22, 35]: lines may become deliberately irregular rather than straight, symmetry may be intentionally broken, and proportions may be distorted to convey a particular visual character or simplification. These operations reflect a higher-level visual reasoning that standard style transfer methods are not equipped to perform.

In this work, we introduce *Abstraction in Style (AiS)*, a generative process that applies abstraction followed by stylization. Unlike prior methods that rigidly adhere to the input structure, our approach first produces an *abstraction proxy*, a structurally abstracted representation in which geometric details are reduced, compositional elements are reimagined, and the overall structure is simplified to emphasize semantic or perceptual relevance. These abstracted forms are forwarded and rendered in the desired abstract style, resulting in visual outputs that exhibit both structural transformation and stylistic consistency. While we do not attempt to model a general abstraction function, our method demonstrates that separating

Authors' addresses: Min Lu, lumin.vis@gmail.com, Shenzhen University, China;

structural abstraction from style enables more expressive and varied visual outcomes, expanding the generative capacity of stylized image synthesis.

In this work, we are particularly aiming at challenging styles like naïve illustration, which are characterized by simple, childlike drawings that often deliberately disregard proportion or perspective. While our method is not limited to this specific style, it is designed to accommodate the kind of abstraction such styles represent. As illustrated in Figure 1, these styles embed abstraction into the visual language itself, using symbolic forms, irregular outlines, and flattened geometry to convey meaning. Unlike more conventional styles that maintain a close correspondence between structure and appearance, these forms require a more interpretive, less literal mode of generation.

In our experiments, we focus primarily on subject-centric image generation, with particular emphasis on architectural structures such as buildings and houses, which serve as a rich domain for exploring structural abstraction. Examples like the one shown in Figure 1 demonstrate how architectural forms can be meaningfully simplified and stylized within this framework.

2 RELATED WORK

2.1 Visual Abstraction and Sketch Simplification

Visual abstraction has received increasing attention in recent years, particularly in the context of sketch generation [3, 21, 23, 31, 32]. Early efforts approached abstraction by modeling how artists simplify visual content. Berger et al. [3] curated a dataset of portrait sketches created by professional artists at multiple levels of abstraction and proposed a retrieval-based method that reassembles new portraits using artist-inspired strokes. Muhammad et al. [23] framed sketch abstraction as a reinforcement learning task, where an agent learns to prune redundant strokes while maintaining recognizability, using classification feedback. These methods highlight the importance of structure in abstraction and are typically developed within task-specific domains such as portrait sketching or edge-map simplification.

More recent works have shifted toward optimization-based approaches that leverage pretrained vision-language models to abstract visual content in a more flexible, data-independent manner. CLIPasso [32] introduced a method for converting object images into vectorized sketches by optimizing Bézier curves to align with both semantic and geometric similarity under a CLIP-based loss. CLIPascene [31] extended this framework to entire scenes, introducing a disentangled abstraction space along two axes: fidelity (how closely the sketch reflects the original image) and simplicity (how much visual detail is preserved). These works demonstrated that high-level semantic abstraction can be achieved without supervised sketch data, and highlighted the potential of abstraction as a controllable visual process. Related approaches, such as Neural Strokes [21], have explored sketch-like rendering from 3D data, showing abstraction can also operate across modalities. While these methods focus on abstraction, they typically produce sketches in a fixed or implicit style, without conditioning on a style example or supporting stylization as a controllable dimension within the generation process.

2.2 Stylization and Style Transfer

Style transfer aims to modify the visual appearance of an image while preserving its structure, most notably introduced by Gatys et al. through optimization over deep feature statistics [8]. Subsequent works improved efficiency and flexibility using feed-forward networks [16, 29], instance normalization [30], adaptive feature modulation [14], and patch-based or universal methods [4, 20].

More recent approaches integrate style transfer into diffusion models. StyleAligned [10] achieves consistent stylization across generations by softly sharing attention during the denoising process; a similar mechanism is explored in Cross-Image Attention [2], where attention layers are modified to propagate visual traits across reference and generated images. B-LoRA [6] adopts low-rank adaptation to implicitly disentangle style and content in diffusion blocks, enabling style recombination without full fine-tuning. InstantStyle [33] and InstantStyle-Plus [36] introduce an architecture that injects style only into selected layers, preserving both spatial layout and semantic fidelity.

While abstraction is often associated with sketch generation, it is not limited to this medium. More broadly, abstraction refers to the reinterpretation of visual structure to reflect semantic or stylistic intent, and it can emerge across a wide range of media. The work of Mehra et al. [22], which abstracts 3D shapes by simplifying geometry, and that of Yaniv et al. [35], which analyzes stylized distortions in artistic portraits, illustrate how abstraction can vary in form and modality. Unlike simplification, which merely reduces detail, abstraction may introduce exaggeration or symbolic transformation, serving as a richer and more expressive process.

3 METHOD

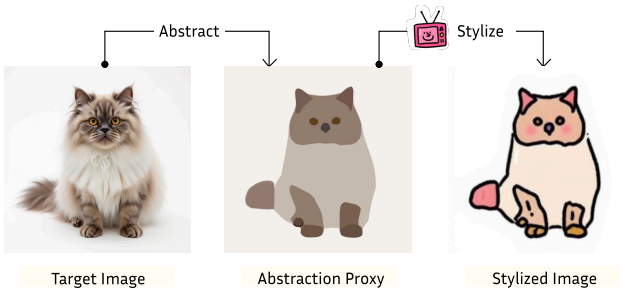

Fig. 2. Two stages of abstraction in style: First, we create a vectorized abstraction proxy that forms an editable representation of the input subject, then in the second stage, we stylize it.

As discussed in the introduction, many artistic styles do not merely alter visual appearance but also embed a form of latent abstraction. These styles often involve structural transformations that go beyond surface-level traits like texture, color, or ornamentation. When such abstraction is entangled with style, applying standard style transfer directly to a realistic image can fail to capture the deeper structural intent conveyed by the reference. To address this, our method separates structural abstraction from visual stylization

and treats them as two distinct stages (see Figure 2). Specifically, we decompose the stylization process into sequential steps: first, we apply an abstraction stage that simplifies the input structure while preserving its semantic content; then, we stylize this abstracted form using the desired target style exemplified by a few images. The output of the first stage is what we call an *abstraction proxy*, an intermediate structural representation that captures the conceptual essence of the input and serves as the basis for subsequent stylization. In the following we elaborate on the two stages.

3.1 Abstraction

The goal of the abstraction stage is to produce a simplified structural representation of the input that captures its essential form while omitting fine-grained details. To achieve this, we adopt a vectorization-based approach. Given an input image, we construct a vectorized representation that emphasizes semantically meaningful parts and eliminates low-level visual complexity. This representation serves as a clean, structured version of the input that retains recognizability but allows for interpretive variation.

We opt for vectorization because it offers a tangible and editable format: the proxy is composed of discrete, interpretable vector primitives that can be directly modified or manipulated. This makes it well-suited not only for downstream stylization, but also for optional user interaction or further abstraction tuning. Figure 3 shows two examples of abstraction proxies generated by our method, demonstrating the balance between structure preservation and detail reduction.

Fig. 3. Examples of abstraction proxies: for each pair, the target image on the left, proxy on the right.

Progressive Image Abstraction. To support the abstraction at multiple levels, we leverage the *feature-average effect* [9, 34] observed in Score Distillation Sampling (SDS) generation mechanism [24]. Unlike conventional pixel-based image simplification methods such as superpixel [1] or bilateral filter [28], SDS-based simplification utilizes the pre-trained knowledge of Denoising Diffusion Probabilistic Models (DDPMs) [11] learned from large-scale dataset. In the diffusion models, the predicted noise consists of two parts via the Classfier-Free Guidance (CFG) [12], defined as follows:

$$\epsilon_\phi^\omega(\mathbf{z}_t, y, t) = (1+\omega)\epsilon_\phi(\mathbf{z}_t, y, t) - \omega\epsilon_\phi(\mathbf{z}_t, t), \tag{1}$$

where $\epsilon_\phi(\mathbf{z}_t, y, t)$ denotes the noise predicted under textual condition y, while $\epsilon_\phi(\mathbf{z}_t, t)$ represents the predicted noise of an unconditional input. To suppress the individual details and only preserve the dominant structure, we set the conditional text prompt to empty (i.e. ' '). Consequently, the SDS loss is updated in random directions, yielding a feature-averaged abstract image. As shown in the upper row of Figure 4, the image becomes increasingly abstract as the SDS-based generation progresses. The four images (left to right) are sampled at iteration steps 0, 30, 60, and 90 respectively.

Fig. 4. Levels of abstraction: (top) from left to right, simplifying the target image via increasing iterations of SDS-based generation; (bottom) vectorizing each image at a certain abstraction level to get the corresponding abstraction proxy.

Vectorized Representation Construction. Images across levels of abstraction are vectorized to get the abstraction proxies at multiple levels (Figure 4(bottom)). To generate the abstraction proxy at a certain level, the corresponding image is first processed by the SAM model [17] for semantic segmentation. An initial spatial filtering step removes masks falling outside the central focal region. For the remaining masks, contours are extracted and geometrically simplified using the Douglas-Peucker method [5]. These contours are then parameterized as vector primitives $\mathcal{P} = \{P_1, P_2, \ldots, P_n\}$, each of which is a closed path with N cubic Bézier curves. These primitives are optimized towards the image-based loss $\mathcal{L}$ via differentiable rendering [19]. We adopt the *layer-wise structural loss* $\mathcal{L}_{\text{structure}}$ from existing work [34] and add the pixel-level Mean Squared Error between the rasterized proxy image I_{proxy} and the input target image I_{target} to maintain the visual fidelity:

$$\mathcal{L} = \underbrace{w_1\mathcal{L}_{\text{mse}} + w_2\mathcal{L}_{\text{overlap}}}_{\text{Layered structural loss [34]}} + \underbrace{w_3\|I_{\text{proxy}} - I_{\text{target}}\|^2}_{\text{Visual fidelity loss}}, \tag{2}$$

where the losses are combined with weighting coefficients $w_1 = 1$, $w_2 = 1$, and $w_3 = 1e-2$.

Style-aligned Color Assignment. Once the vectorized proxy is formed, we can easily enhance it, for example, by modifying its colors. With the visual fidelity loss during the vectorization, the vectorized proxy is, by default, colored to fit the original target image. There can be flexible coloring strategies. Figure 5 presents a possible approaches that transfer colors from the style reference to the proxy by semantic correlation between vector primitives. The semantic color assignment first identifies the feature correspondence between the proxy and stylized image using the DIFT [27] on the pixel level. Then these correspondences are aggregated through pixel voting to establish shape-to-shape relationships. Finally, each proxy shape

inherits the color from its corresponding highest-vote shape in the stylized image.

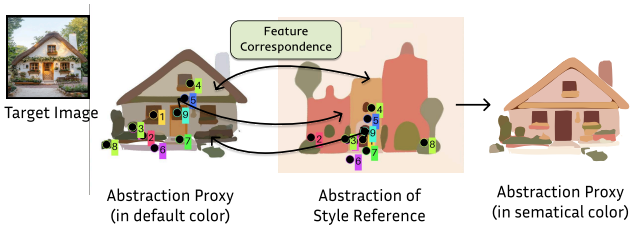

Fig. 5. Recoloring vector primitives in proxy according to their feature correspondence with the abstraction of style reference.

3.2 Stylization

In our work, we present an In-Context technique for few-shot visual stylization. Our strategy is to frame stylization as an in-context learning problem, where a small set of reference examples is used to guide the transformation from the abstraction proxy to style. This setup allows the model to internalize both visual and structural aspects of the target style through lightweight adaptation. One-shot methods like StyleAligned [10] offer fast adaptation but often miss subtle, style-defining traits, especially in abstract or expressive styles. Moreover, incorporating structural conditions, through direct feature adding, like ControlNet [37], often leads to produce results that strictly adheres to the input structure and lacks flexibility where structural reinterpretation is required. In contrast, our few-shot in-context approach leverages a small number of reference examples to more accurately capture the unique characteristics of the target style, including fine structural and visual nuances.

Abstract→Style Pair Curation. We assemble a set of reference examples in the desired style, each consisting of a pair of abstracted and stylized images (i.e., *abstract→style*). In each pair, the abstracted image is obtained by applying aforementioned abstraction (Sec. 3.1) to each stylized image cropped from the user input. These pairs guide the model in learning how style should be applied to abstracted inputs. To present these examples effectively, we organize them in a 2×2 grid (see Figure 6): the top row contains a reference pair of abstraction I^s_{abs} and its corresponding stylized image I_s, while the bottom row contains a new abstraction proxy I^t_{abs} alongside the target image I_t itself. This layout allows the model to infer the appropriate transformation from abstraction to style based on the given reference.

To support minimal user input design examples, like five examples as shown in Figure 1, we broaden our training data by synthesizing extra abstract-style grid samples with text-to-image diffusion model. Specifically, we adopt the prompt engineering technique from IC-LoRA [13] and finetune FLUX.1-Dev [18] on ten abstract→style grids from the original examples, as exemplified on the left of Figure 7. Then the model is fine-tuned to generate a richer variety of abstract→style image pairs by texts, covering diverse subjects, such as furniture, architecture, daily objects, and more. Figure 7

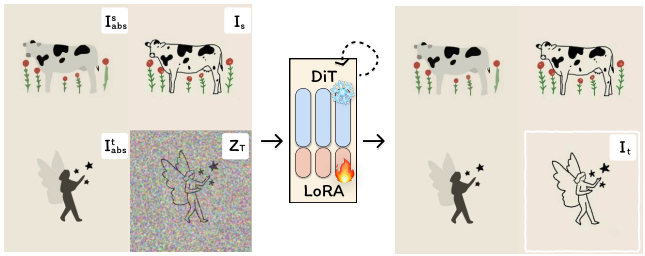

Fig. 6. Illustration of stylization process: the reference pair (abstraction and its stylized counterpart) is arranged in a 2×2 grid alongside the target abstraction and a noisy input. A diffusion transformer model is optimized to denoise the input and predict stylized output.

Fig. 7. Abstract→style pair curation: (left) a grid example of the original design, on which the FLUX.1-Dev model is finetuned; (right) a generated coherent stylized example by text prompt.

(right) shows a text-generated example exhibiting stylistic consistency with the target reference style shown on the left. More results are provided in the supplementary material.

Few-shot Tuning. Building on our four-panel *abstract→style* storyboards arranged in a 2×2 grid, we finetune an inpainting diffusion model (i.e., FLUX.1-Fill-dev [18]) to reconstruct the target stylized image I_t in the bottom-right panel. The model takes as input the visual context from the other three panels $I_c = (I^s_{\text{abs}}, I_s, I^t_{\text{abs}})$ and a text instruction T. During training, we first encode the target image into its style latent z_0, then noise it into a noisy version z_t. The context panels are encoded into image tokens c_I and the text instruction into text tokens c_T. The diffusion model learns to denoise the target noisy latent by conditioning on both the visual context and the text condition, effectively reconstructing the target image while capturing artistic style and preserving natural structure.

4 RESULTS

4.1 Implementation Details

For the abstraction part, we implemented the vectorization method using PyTorch with the Adam optimizer. We use the SAM model with checkpoint sam_vit_h_4b8939. The minimum area (in pixels) that a mask region to be vectorized is set to 100. The contour of a mask is approximated by polyline using the Douglas-Peuker algorithm [5], within a tolerance threshold ϵ (ϵ = 5.0 in our work). For the data curation part, we fine-tune FLUX.1-dev using LoRA with a rank of 16 for 5000 steps. For the in-context tuning part, we fine-tune FLUX.1-Fill-dev using LoRA with a rank set to 16 for

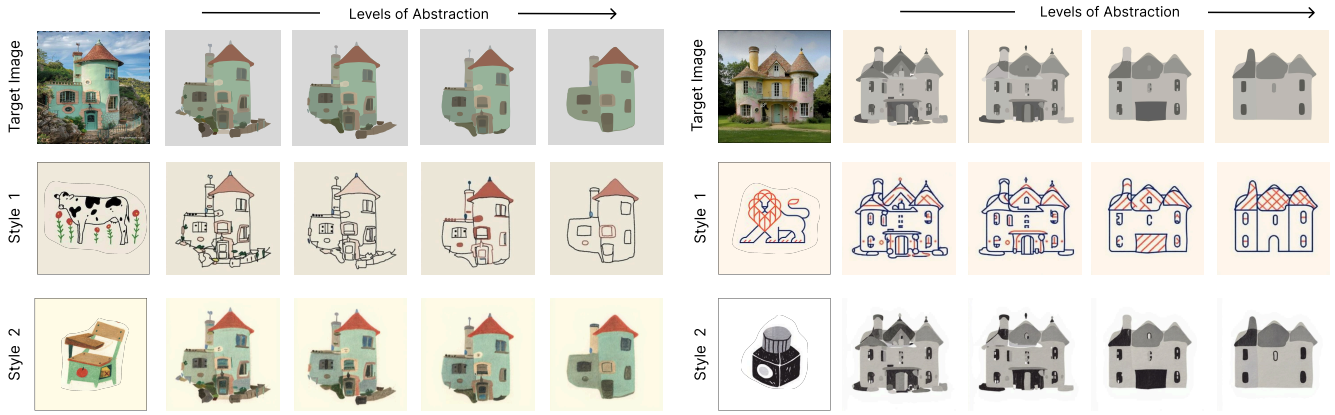

Fig. 8. Examples generated at different levels of structural abstraction: For each case, (top) four abstraction proxies are sampled at increasing simplification steps, and (bottom) their corresponding stylized images with progressively reduced detail.

1000 steps. Each training takes around two hours at 1024×1024 resolution on a single A100 (80GB) GPU.

4.2 Examples

Our method enables image generation across diverse artistic styles. Figure 14 and Figure 15 present results where the same content appears in multiple styles. As can be seen, our method transfers the style to the target image and adapts the target image's structure organically, preserving key features while reinterpretating them stylistically. Below, we demonstrate the versatility of our method through several applications enabled by integrating our approach with complementary techniques. Note that in those examples, the style references are represented by a single example, the full design references are in the supplementary material.

Progressive Abstraction in Style. Our SDS-based abstraction method allows progressive abstraction in styles, by generating abstraction proxies at multiple abstraction levels. Figure 8 illustrates this progression through four proxies sampled at different steps of SDS generation. For each proxy, we generate corresponding stylized images (shown below), demonstrating a complete transformation from fine-grained details to fundamental structures.

Sequential Image Stylization. Figure 9 shows an example of using our technique to perform style transfer on a sequence of scenes. Here we show two distinct styles, stick style (third row) and stamp-like style (fourth row). As can be seen, the characters maintain stylistic coherence throughout all scenes, including faces rendering. Due to the vectorized abstraction proxy, we can easily extend the sequence with a new scene where 'The bread is crying with tears'.

Clipart Editing and Fusion. The vector-based abstraction proxy facilitates easy editing and manipulation. Figure 10 illustrates an interactive application of our method, where users first collect image clips in wild styles, ranging from photorealistic to illustrative styles. With the abstraction proxies generated for those images by our method, users compose these elements through editing. Then the unified abstraction is stylized. Figure 10 gives two stylized examples. As can be seen, the originally disparate clips merge harmoniously in the final output.

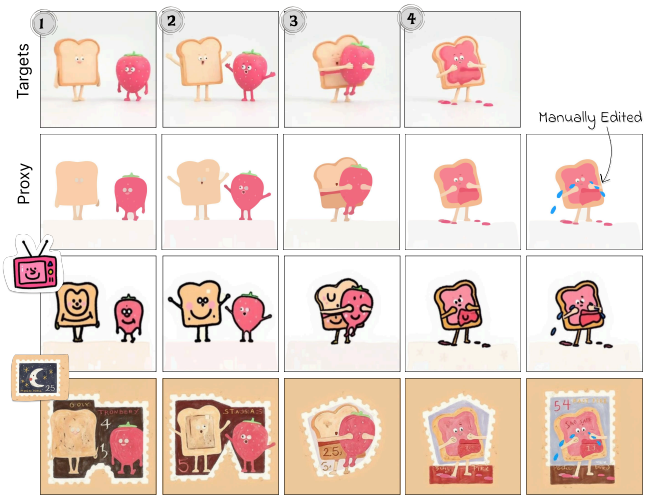

Fig. 9. Stylization over sequence of scenes: our method supports consistent stylistic generation of scenes in a sequence, based on which the scene can be easily edited, e.g., adding a new scene at the end.

5 EVALUATION

We conduct comprehensive evaluations to demonstrate the effectiveness of our approach, including both qualitative comparisons and quantitative measurements.

5.1 Ablation Study

Ablation on Abstraction Proxy.

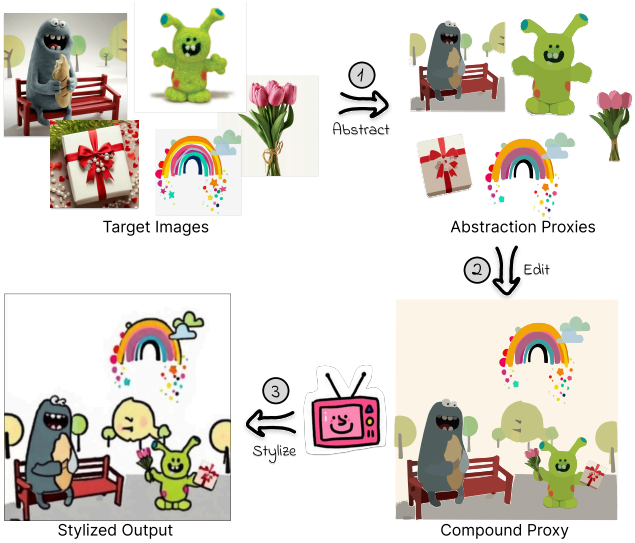

Fig. 10. Clipart fusion and stylization: Our method enables interactive fusion and editing of vector-based abstraction proxies from input images to generate visually cohesive, style-consistent compositions.

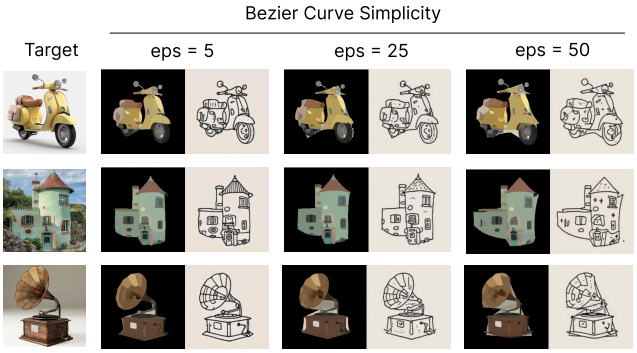

Fig. 11. Comparison of stylization results using Bézier curves at three simplification levels: finer curves (lower ϵ) preserve sharper geometric details and improve alignment with the target image, as seen in the roof-wall junction (second row). Higher simplification (larger ϵ) introduces cracked contours.

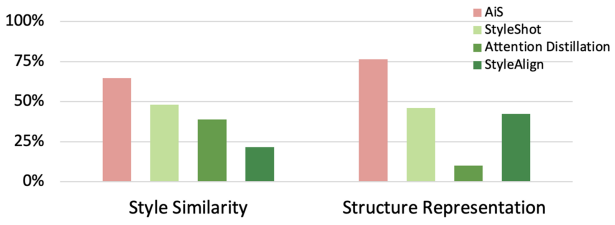

Fig. 12. User study results comparing different methods across two metrics: style similarity, and structure representation.

Ablation on the Bézier Curve Simplicity. We investigate the impact of optimizing Bézier curve simplicity in the abstraction proxy on the quality of stylization. During the abstraction stage, segmented mask contours are represented as polygons with varying levels of simplification using the Douglas-Peucker algorithm, controlled by the parameter ϵ, defining the simplification tolerance: greater values produce coarser approximations by allowing a greater deviation from the original curve, while smaller ϵ preserves finer geometric details. We tested three values of ϵ (5, 25, and 50), where larger values result in higher simplification. For each simplicity level, we paired up the simplified curves and stylized image to fine-tune the model, so that it can be used to generate stylized outputs.

As shown in Figure 11, finer-grained Bézier curves (lower ϵ) preserve more contour details, while higher simplification (larger ϵ) increases aliasing artifacts in the generated images. For all examples of this work, ϵ is set to 5.

5.2 Qualitative Comparison

Figure 13 presents a qualitative comparison between our method and previous methods, including GPT-4o [15], StyleAlign [10], StyleShot [3], and Attention Distillation [38]. As can be seen, our method more faithfully captures the essence of the style reference, exhibiting clearer and more consistent style similarity while preserving the structural integrity of the target images. In contrast to other methods, our results demonstrate superior detail retention and stylistic accuracy, closely matching the original reference.

5.3 Quantitative Analysis

Benchmark. To explore the application of diverse styles, we gathered style reference images from 23 different designers on *Pinterest* and generated 20 target images using the pretrained text-to-image model FLUX.1-dev. These target images, spanning categories such as animals, buildings, food, and everyday objects, were stylized with the 23 reference styles, resulting in a total of 460 testing samples.

Metrics. We assess the quality of stylization by evaluating the similarity between style representations extracted using Contrastive Style Descriptors (CSD) [26] and CLIP [25], B-Lora [], and UnzipLoRA []. Trained on diverse style datasets, CSD provides robust representative features for image styles, making it well-suited to examine a model's capacity for stylization. For every stylized image produced by each model, we measure its similarity to a cropped patch from the original style reference image.

Analysis. Table 1 shows the style similarity scores on our benchmark. Our method achieves consistently higher scores (indicating better style similarity) compared to baseline methods. Such quantitative results align with our qualitative observations, demonstrating that our approach better preserves the distinctive characteristics of target styles.

User Study. In addition to quantitative evaluation, we conducted a user study comparing our method with the baseline methods. We randomly sampled three tests from each style in a round robin schedule of method comparison, resulting in 69 pairs to compare (23 styles x 3 samples). For each pair, users were asked to judging (1) style

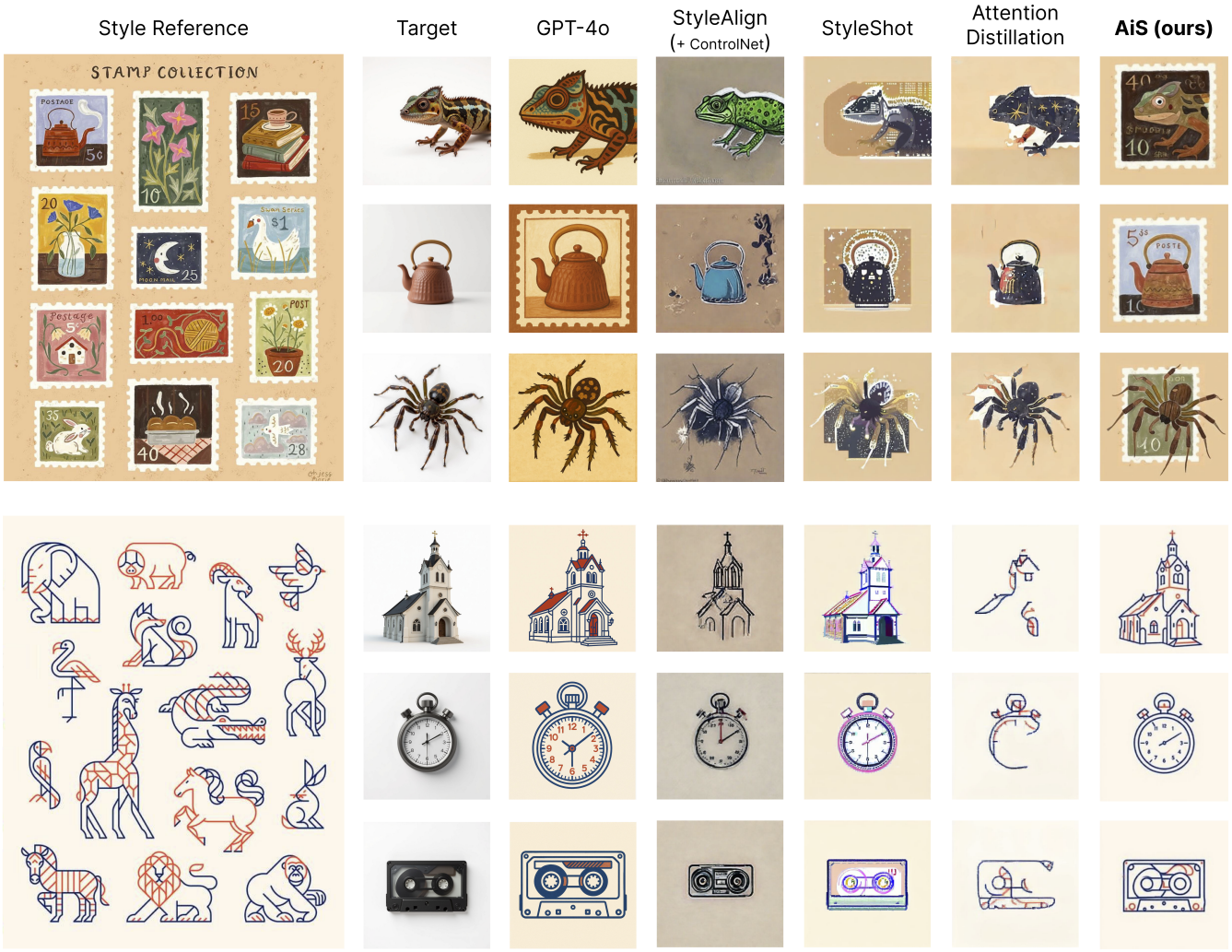

Fig. 13. Qualitative comparison with baseline methods: existing methods struggle with abstract styles and over-rigidly preserve input structure, producing unsatisfactory results. Our method excels at capturing nuanced artistic styles while maintaining natural structural variations.

Table 1. Quantitative comparison with baselines: we report average style similarity scores with different feature extractors. AD refers to Attention Distillation [38].

Metrics	StyleAlign [10]	AD [38]	StyleShot [7]	**AiS (Ours)**
CSD	0.11	0.35	0.27	**0.45** ↑
CLIP	0.47	0.51	0.48	**0.55** ↑

similarity to the design reference and (2) structure representation to represent the underlying geometries.

We gathered 2346 responses from 34 participants. The rating results, summarized in Table 1, show the percentage of judgments favoring our method. As evident, most participants preferred our approach by a significant margin. Details about the user study are provided in the supplementary materials.

6 CONCLUSION

In this work, we introduced Abstraction in Style (AiS), a generative framework that decomposes stylization into two sequential stages: structural abstraction followed by visual rendering. This separation enables the synthesis of stylized images that are not only faithful to the visual traits of the reference style but also reflect its underlying abstraction logic. In practice, we realized this concept by first constructing an editable abstraction proxy through semantic filtering and vectorization, and then stylizing it using a few-shot in-context learning approach based on an image inpainting diffusion model.

While our implementation demonstrates the effectiveness of the two-stage design, it currently relies on a relatively simple and naive

vectorization process in the abstraction stage. Nonetheless, it serves as a concrete and effective instantiation of the broader AiS paradigm. The concept of separating abstraction from style is general, and we envision alternative realizations that incorporate richer structural representations and more expressive abstraction mechanisms to extend the framework's versatility.

In the future, we aim to develop more advanced abstraction techniques that go beyond simplification and allow for controlled semantic distortion. Such methods would enable the system to selectively warp or exaggerate salient elements of the subject in ways that align with both the structure and expressive character of the target style.

REFERENCES

[1] Radhakrishna Achanta, Appu Shaji, Kevin Smith, Aurelien Lucchi, Pascal Fua, and Sabine Süsstrunk. 2012. SLIC superpixels compared to state-of-the-art superpixel methods. *IEEE transactions on pattern analysis and machine intelligence* 34, 11 (2012), 2274–2282.

[2] Yuval Alaluf, Daniel Garibi, Or Patashnik, Hadar Averbuch-Elor, and Daniel Cohen-Or. 2023. Cross-image attention for zero-shot appearance transfer. *ACM Transactions on Graphics (TOG)* 42, 4 (2023), 1–11. https://doi.org/10.1145/3592432

[3] Itay Berger, Ariel Shamir, Miriam Mahler, and Eyal Carter. 2013. Style and abstraction in portrait sketching. *ACM Transactions on Graphics (TOG)* 32, 4 (2013), 1–12. https://doi.org/10.1145/2461912.2461964

[4] Dongdong Chen, Jing Liao, Lu Yuan, Nenghai Yu, and Gang Hua. 2016. Fast patch-based style transfer of arbitrary style. *arXiv preprint arXiv:1612.04337* (2016).

[5] David H Douglas and Thomas K Peucker. 1973. Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. *Cartographica: the international journal for geographic information and geovisualization* 10, 2 (1973), 112–122.

[6] Yarden Frenkel, Yael Vinker, Ariel Shamir, and Daniel Cohen-Or. 2024. Implicit style-content separation using B-LoRA. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

[7] Junyao Gao, Yanchen Liu, Yanan Sun, Yinhao Tang, Yanhong Zeng, Kai Chen, and Cairong Zhao. 2024. Styleshot: A snapshot on any style. *arXiv preprint arXiv:2407.01414* (2024).

[8] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. 2016. Image style transfer using convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2414–2423. https://doi.org/10.1109/CVPR.2016.265

[9] Amir Hertz, Kfir Aberman, and Daniel Cohen-Or. 2023. Delta denoising score. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2328–2337.

[10] Amir Hertz, Andrey Voynov, Shlomi Fruchter, and Daniel Cohen-Or. 2024. Style aligned image generation via shared attention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 4775–4785.

[11] Jonathan Ho, Ajay Jain, and Pieter Abbeel. 2020. Denoising diffusion probabilistic models. *Advances in neural information processing systems* 33 (2020), 6840–6851.

[12] Jonathan Ho and Tim Salimans. 2022. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598* (2022).

[13] Lianghua Huang, Wei Wang, Zhi-Fan Wu, Yupeng Shi, Huanzhang Dou, Chen Liang, Yutong Feng, Yu Liu, and Jingren Zhou. 2024. In-Context LoRA for Diffusion Transformers. *arXiv preprint arxiv:2410.23775* (2024).

[14] Xun Huang and Serge Belongie. 2017. Arbitrary style transfer in real-time with adaptive instance normalization. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. 1501–1510. https://doi.org/10.1109/ICCV.2017.167

[15] Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. 2024. Gpt-4o system card. *arXiv preprint arXiv:2410.21276* (2024).

[16] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. 2016. Perceptual losses for real-time style transfer and super-resolution. In *European Conference on Computer Vision (ECCV)*. 694–711. https://doi.org/10.1007/978-3-319-46475-6_43

[17] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. 2023. Segment anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 4015–4026.

[18] Black Forest Labs. 2024. FLUX. https://github.com/black-forest-labs/flux.

[19] Tzu-Mao Li, Michal Lukáč, Michaël Gharbi, and Jonathan Ragan-Kelley. 2020. Differentiable vector graphics rasterization for editing and learning. *ACM Trans. Graph.* 39, 6, Article 193 (nov 2020), 15 pages. https://doi.org/10.1145/3414685.3417871

[20] Yijun Li, Chen Fang, Jimei Yang, Zhaowen Wang, Xin Lu, and Ming-Hsuan Yang. 2017. Universal style transfer via feature transforms. In *Advances in Neural Information Processing Systems (NeurIPS)*. 386–396.

[21] Difan Liu, Matthew Fisher, Aaron Hertzmann, and Evangelos Kalogerakis. 2021. Neural strokes: Stylized line drawing of 3D shapes. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. 11546–11555. https://doi.org/10.1109/ICCV48922.2021.01136

[22] Ravish Mehra, Qingnan Zhou, Jeremy Long, Alla Sheffer, Amy Gooch, and Niloy J Mitra. 2009. Abstraction of man-made shapes. In *ACM SIGGRAPH Asia 2009 papers*. 1–10.

[23] Umar Riaz Muhammad, Yongxin Yang, Yi-Zhe Song, Tao Xiang, and Timothy M Hospedales. 2018. Learning deep sketch abstraction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 8015–8024. https://doi.org/10.1109/CVPR.2018.00836

[24] Ben Poole, Ajay Jain, Jonathan T Barron, and Ben Mildenhall. 2022. Dreamfusion: Text-to-3d using 2d diffusion. *arXiv preprint arXiv:2209.14988* (2022).

[25] 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. arXiv:2103.00020 [cs.CV] https://arxiv.org/abs/2103.00020

[26] Gowthami Somepalli, Anubhav Gupta, Kamal Gupta, Shramay Palta, Micah Goldblum, Jonas Geiping, Abhinav Shrivastava, and Tom Goldstein. 2024. Measuring Style Similarity in Diffusion Models. arXiv:2404.01292 [cs.CV] https://arxiv.org/abs/2404.01292

[27] Luming Tang, Menglin Jia, Qianqian Wang, Cheng Perng Phoo, and Bharath Hariharan. 2023. Emergent Correspondence from Image Diffusion. In *Thirty-seventh Conference on Neural Information Processing Systems*. https://openreview.net/forum?id=ypOiXjdfnU

[28] Carlo Tomasi and Roberto Manduchi. 1998. Bilateral filtering for gray and color images. In *Sixth international conference on computer vision (IEEE Cat. No. 98CH36271)*. IEEE, 839–846.

[29] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. 2016. Texture networks: Feed-forward synthesis of textures and stylized images. In *International Conference on Machine Learning (ICML)*. 1349–1357.

[30] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. 2017. Improved texture networks: Maximizing quality and diversity in feed-forward stylization and texture synthesis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 4105–4113. https://doi.org/10.1109/CVPR.2017.437

[31] Yael Vinker, Yuval Alaluf, Daniel Cohen-Or, and Ariel Shamir. 2023. Clipascene: Scene sketching with different types and levels of abstraction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 4146–4156.

[32] Yael Vinker, Ehsan Pajouheshgar, Jessica Y Bo, Roman Christian Bachmann, Amit Haim Bermano, Daniel Cohen-Or, Amir Zamir, and Ariel Shamir. 2022. CLIPasso: Semantically-Aware Object Sketching. *ACM Transactions on Graphics (TOG)* 41, 4 (2022), 1–13. https://doi.org/10.1145/3528223.3530068

[33] Haofan Wang, Peng Xing, Renyuan Huang, Hao Ai, Qixun Wang, and Xu Bai. 2024. Instantstyle-plus: Style transfer with content-preserving in text-to-image generation. *arXiv preprint arXiv:2407.00788* (2024).

[34] Zhenyu Wang, Jianxi Huang, Zhida Sun, Yuanhao Gong, Daniel Cohen-Or, and Min Lu. 2024. Layered Image Vectorization via Semantic Simplification. *arXiv preprint arXiv:2406.05404* (2024).

[35] Jordan Yaniv, Yael Newman, and Ariel Shamir. 2019. The face of art: landmark detection and geometric style in portraits. *ACM Transactions on graphics (TOG)* 38, 4 (2019), 1–15.

[36] Kai Zhang, Yijun Li, Sifei Liu, and Jun-Yan Zhu. 2024. InstantStyle-Plus: Style transfer with content-preserving control in diffusion models. *arXiv preprint arXiv:2403.19627* (2024).

[37] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. 2023. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF international conference on computer vision*. 3836–3847.

[38] Yang Zhou, Xu Gao, Zichong Chen, and Hui Huang. 2025. Attention Distillation: A Unified Approach to Visual Characteristics Transfer. In *CVPR*.

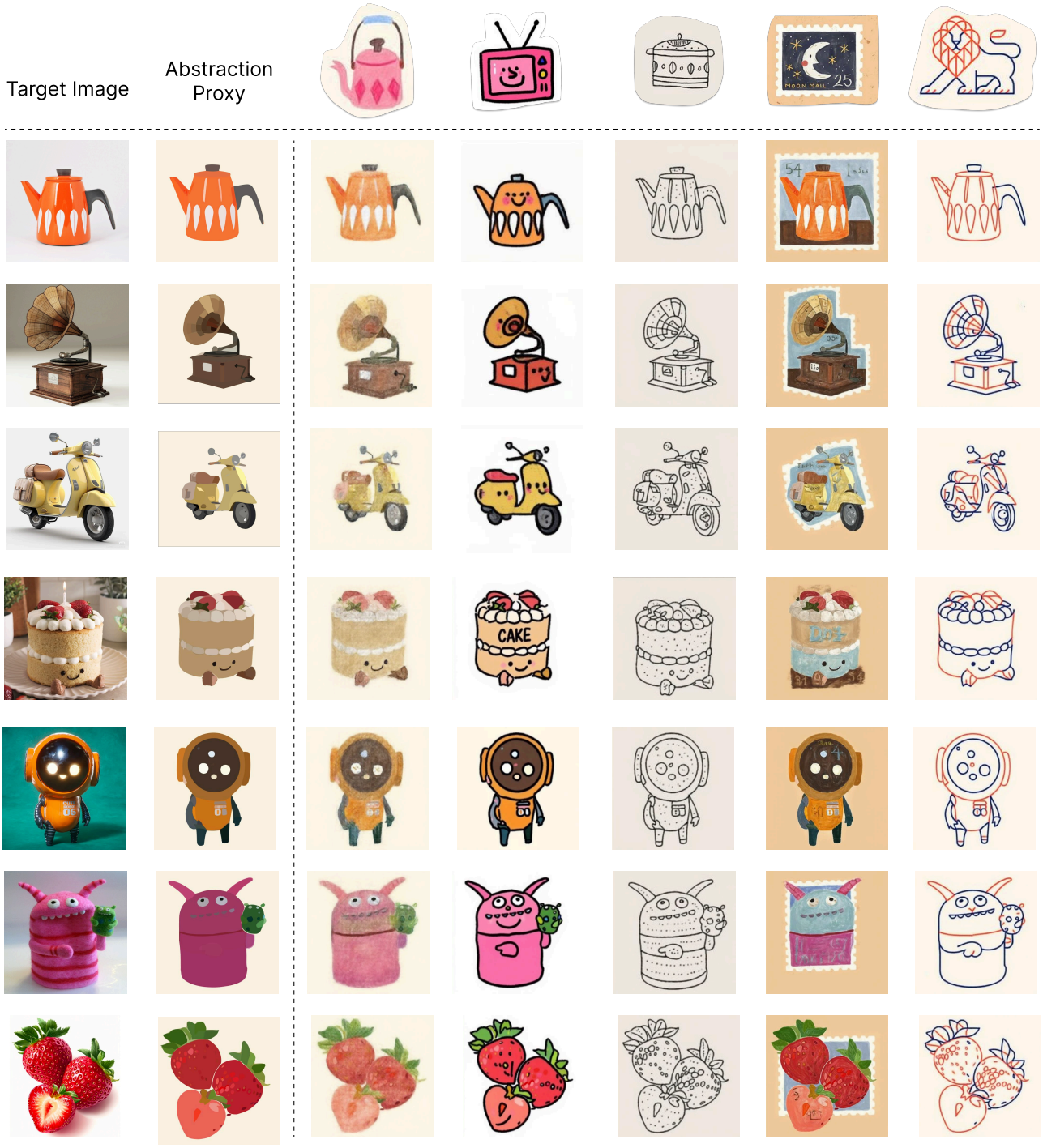

Fig. 14. A gallery of examples generated over a diverse range of styles, including line-style, watercolor-like rendering, etc.

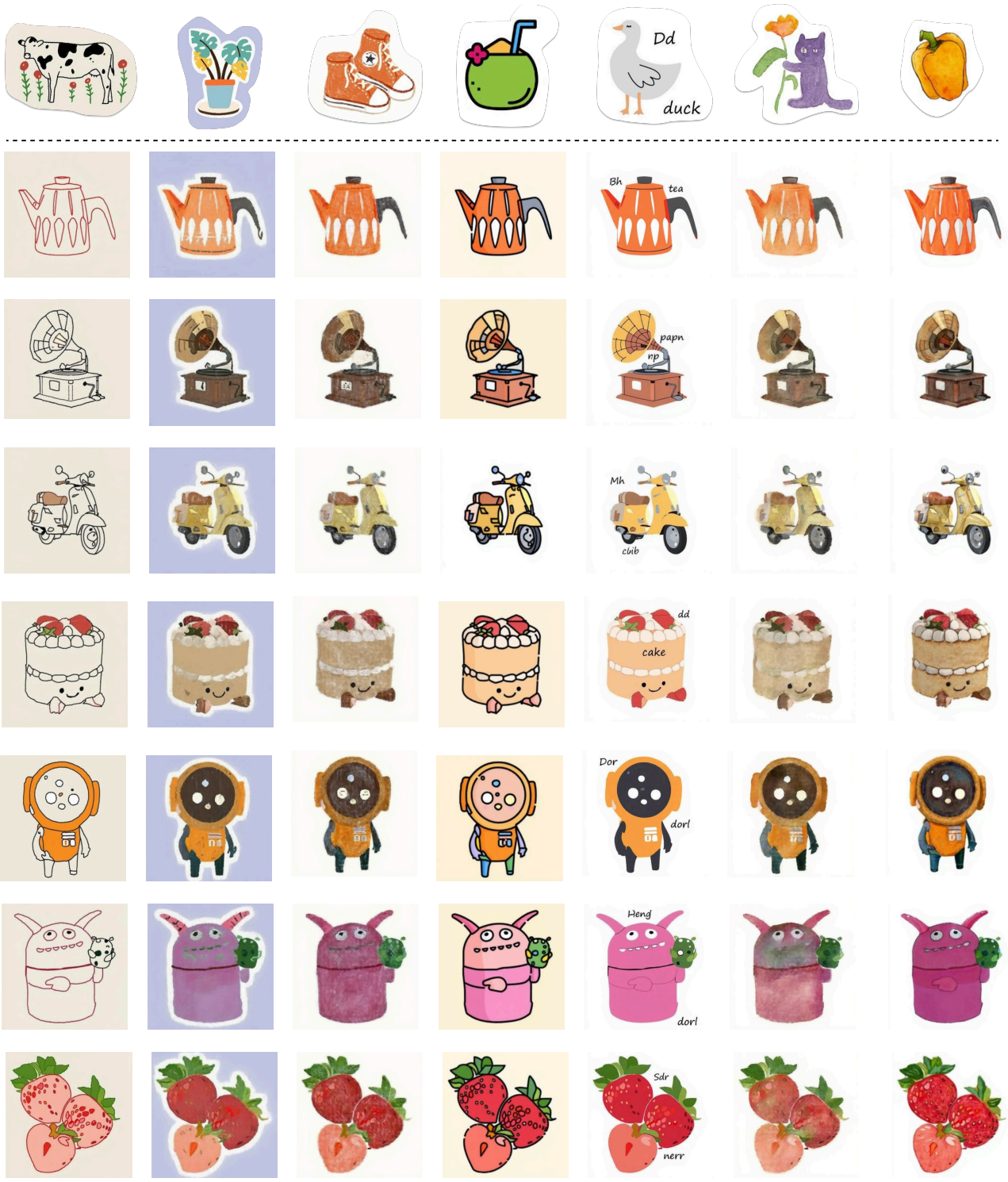

Fig. 15. A gallery of examples generated over a diverse range of styles, including line-style, watercolor-like rendering, etc. Note that this figure shares the same target in each row with Figure 14.