A Style is Worth One Code: Unlocking Code-to-Style Image Generation with Discrete Style Space

Supplementary Material

7. Experimental Details

In this chapter, we introduce the settings of the baselines we compared against.

When validating style-code-conditioned image generation, we compared our method with MidJourney [2]. Since MidJourney's code is not open-sourced, we manually collected all test data from the MidJourney website.

When validating image generation conditioned on style images, we compared our method with state-of-the-art style image-based approaches, including StyleStudio [19], CSGO [47], USO [46], Flux-Kontext [17, 18], and InstantStyle [43]. For StyleStudio [19], we set the classifier-free guidance (CFG) [14] to 5.0 and the number of denoising steps to 50. For the style scale controlling style intensity, we used the default value of 1.0 provided by the official implementation. For CSGO [47], we set CFG to 10.0 and the denoising steps to 50. During generation, since no image condition is required, we followed the official recommendation and set the ControlNet [50] conditioning scale to 0.01 and the style scale to 1.0. For USO [46], we set CFG to 4.0 and the denoising steps to 25. For Flux-Kontext [17, 18], we set CFG to 2.5 and, following the procedure in its paper, used the style image as condition while prefixing the prompt with "Using this style, ...". For InstantStyle [44], we set CFG to 5.0, the denoising steps to 30, and the style scale to 1.0. For all the above methods, we loaded the official pre-trained model weights.

8. Training Dataset

Our dataset comprises 400,000 image pairs generated following StyleMaster [49] and 20,000 pairs from a paid, proprietary source. Each pair in the dataset contains two images with the same style but different content. The number of styles is approximately equal to the number of pairs.

For data curation, we use both manual and automatic annotation. For manual annotation, annotators remove pairs lacking style consistency or content fidelity. For automatic annotation, the CSD metric is evaluated based on style consistency and pairs with lower scores are removed. We label intra-pair comparison as $y_i = 1$ and inter-pair comparison as $y_i = 0$.

9. Analysis of High-frequency Index

To further investigate the learned style codebook, we analyze the utilization frequency of its indices to uncover potential biases or emergent patterns in how the model leverages its discrete representation space. Specifically, we construct a diverse dataset of stylistically varied images and encode each image using the trained codebook to extract the sequence of discrete indices that best reconstruct its style. We then aggregate the occurrence counts of each codebook index across the entire dataset, yielding a distribution that reflects the relative usage of each latent token.

As illustrated in Fig. A1, where the horizontal axis denotes the codebook index and the vertical axis represents the normalized selection frequency, we observe a pronounced long-tail distribution: a small subset of indices (approximately 5–10% of the total) is selected with significantly higher frequency than the vast majority of others. This imbalance suggests that the model disproportionately relies on a limited number of tokens to encode stylistic information, rather than distributing semantic content evenly across the codebook.

Motivated by this observation, we conduct a detailed analysis in Sec. 3.3 of the main paper to determine whether these high-frequency indices correspond to interpretable stylistic attributes—such as brushstroke texture, color palette, or compositional layout. Through controlled ablations, we replace these dominant indices with random or shuffled values and observe minimal degradation in reconstruction quality or perceptual style fidelity. Furthermore, when we isolate the outputs generated using only the top-k high-frequency indices, the resulting images retain strong stylistic coherence despite the absence of low-frequency tokens. These findings strongly indicate that the high-frequency indices do not encode specific, semantically meaningful stylistic features. Instead, they function as placeholder tokens, essentially serving as generic, high-probability surrogates that the model defaults to for efficiency, likely due to overfitting or insufficient codebook capacity during training. This behavior mirrors the phenomenon observed in language models, where certain "filler" tokens dominate usage in the absence of fine-grained semantic differentiation. Our analysis thus reveals a critical limitation in the codebook's expressiveness and motivates future work toward more balanced, semantically disentangled discrete representations.

10. Implications and Insights

In this paper, we not only propose a novel approach to code-to-style generation but also offer several insightful observations worth further discussion.

First, we introduce a discrete style codebook to extract

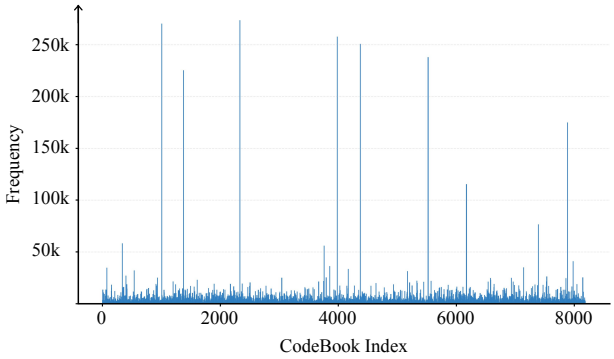

Figure A1. **Frequency distribution of style codebook indices.** A batch of images is encoded using the style codebook, and the selection frequency of all indices is calculated.

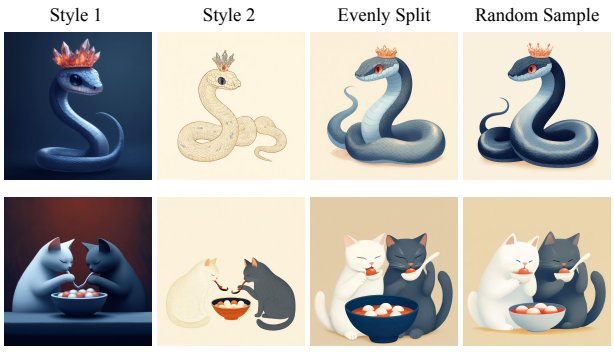

Figure A2. The impact of token selection strategy on the generation results during style interpolation.

and represent stylistic information from images in a discrete latent space. Building upon this representation, we demonstrate that smooth style interpolation can be achieved by blending different style codes. As shown in Fig. A2, we observe that the specific token selection strategy during interpolation has minimal impact on the generated results, regardless of whether it is achieved through random sampling or splitting tokens evenly between the two styles (e.g., first half from style A and second half from style B). This indirectly highlights a key distinction between our learned style representation and traditional image representations: the stylistic information is invariant to the order of its tokens. This property aligns naturally with human intuition about style, which is typically perceived as a holistic attribute rather than a sequence-dependent structure. Style interpolation is just one emergent application enabled by our learned style representation; the structural properties of this discrete space may facilitate the development of additional, more sophisticated, and practically valuable functionalities. Furthermore, this discrete feature extraction paradigm holds promise for extension to other modalities, such as audio.

Second, we advocate for injecting style information from the textual modality rather than the visual. We argue that human perception of style is inherently semantic rather than purely chromatic; thus, conditioning on text enables the model to capture stylistic attributes that better align with human intuition.

Finally, code-to-style image generation is a task of substantial practical potential, yet it has remained unexplored in the academic literature. We present the first comprehensive framework to address this problem and publicly release both the model weights and source code, which we hope will inspire further research in this emerging direction.

11. Limitations

The CoTyle framework represents a significant advancement in the domain of code-to-style image generation. However, several limitations should be acknowledged and explored in future research.

Firstly, while the experimental results indicate that the stylistic diversity of images generated by CoTyle is commendable, it remains somewhat lower than that observed in Midjourney. This outcome is likely linked to the training regimen of the autoregressive style generator, which relies on a discrete style codebook. This codebook, constructed from a limited dataset of curated style-paired images, may not fully encapsulate the extensive variability inherent in human artistic expressions. Expanding the diversity of training data with a broader range, or implementing more advanced generative architectures like mixture-of-experts models, may enhance the representation of complex, multi-modal style distributions.

Secondly, the resolution and creativity of the generated styles are largely dictated by the quality of the style codebook. Despite utilizing contrastive loss to differentiate styles, the necessity to quantize these into a finite codebook inevitably entails some loss of detailed stylistic subtleties. As a result, certain intricate or hybrid nuances found in reference images may be attenuated or omitted, potentially affecting the expressiveness of the generated outputs.

Finally, although the numerical style code offers a robust framework for reproducibility, the exploration of aesthetically or artistically engaging styles presently involves a stochastic search across a vast code space. Introducing more intuitive mechanisms for navigating or modifying this latent style space could significantly enhance both user agency and practical applicability.

Although CoTyle still has some areas for improvement, as the first open-source work on the task of code-to-style generation, it introduces an overall framework for addressing this task, holding the potential to inspire subsequent research in the field of code-to-style generation.

12. More Results

12.1. Style Interpolation

We provide additional style interpolation results in Fig. A4. Given two images, we encode them into two sets of indices using the style codebook and achieve style interpolation by randomly blending the two sets of indices according to a specified mixing ratio. In Fig. A4, the leftmost and rightmost images represent the style references, respectively, while the intermediate images show the results of style interpolation generated with a given prompt, using progressively varying blending ratios between the two styles.

12.2. Code-to-Style Generation

We present additional code-to-style generation results in Fig. A5, Fig. A6, and Fig. A7 to provide a more comprehensive evaluation of our method's capability in translating structured code inputs into visually coherent stylistic outputs. In each figure, the generation process is carefully controlled: all images within a given row are produced using the same random seed, ensuring that variations across columns arise solely from differences in the input text prompt, rather than stochastic sampling. Conversely, all images within a column are generated under the same textual instruction, allowing direct comparison of stylistic consistency and fidelity across different code conditions. This design enables a clear disentanglement of the influence of code structure versus textual guidance on the final output. The corresponding text prompts for each column are listed in Table A2. These results further demonstrate the robustness of our approach in mapping diverse code-based styles to rich, semantically aligned visual outputs.

12.3. Style Injection

As shown in Tab. A1, we supplement the quantitative results of the study on the effect of style injection via the textual branch, which further supports the viewpoint that style injection through the textual branch is more effective.

Table A1. Comparison of injecting style condition to DiT through visual branch and textual branch.

Methods	Aesthetics ↑	CLIP-T ↑	Consistency ↑
Visual Branch	0.7175	**0.3255**	0.5306
Textual Branch	**0.7178**	0.3230	**0.5791**

13. User Study

We collect 35 volunteers' responses for a total of 144 samples for user study. We invited human volunteers to provide ratings along four metrics: diversity, consistency, aesthetics, and overall preference.For consistency and aesthetics, participants were presented with image sets generated by the same style code and asked to score them on a 3-point scale.For diversity, they were shown multiple image sets produced with different style codes and instructed to rate them on the same 3-point scale.For overall preference, images generated by Cotyle and Midjourney were split into two separate groups, and participants were asked to select their preferred group; the selection frequency served as the overall preference score.Finally, the scores of all metrics were normalized to the range $[0, 100]$, with the results illustrated in Fig. A3 (a).

Additionally, we conduct an automatic evaluation using GPT in a similar manner, as illustrated in Fig. A3 (b). Both the human and automatic results demonstrate that CoTyle yields superior aesthetics and style consistency, albeit with slightly lower diversity.

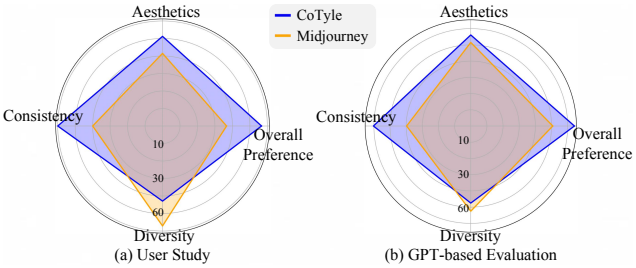

Figure A3. More Evaluations.

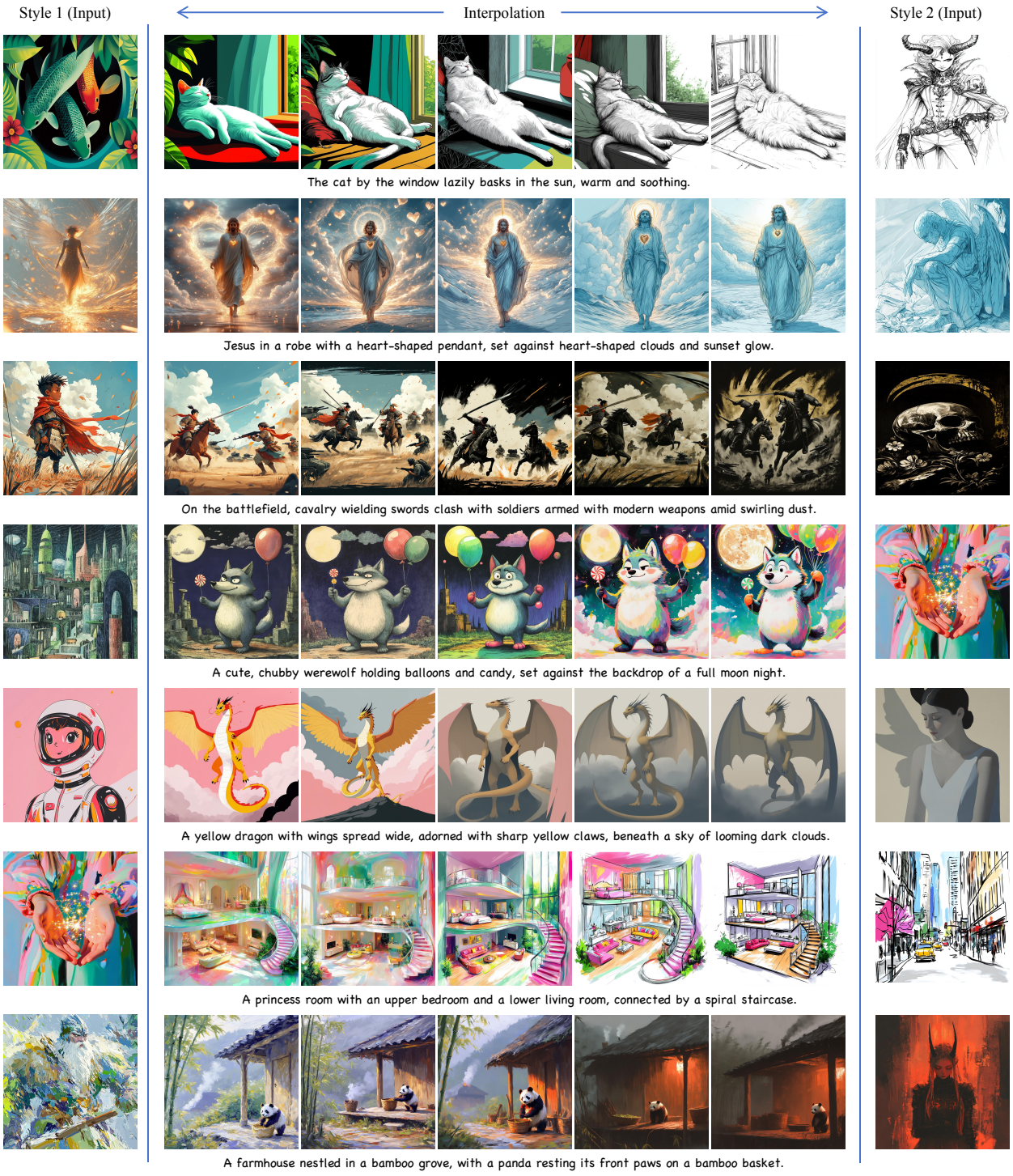

Figure A4. More visual results of style interpolation.

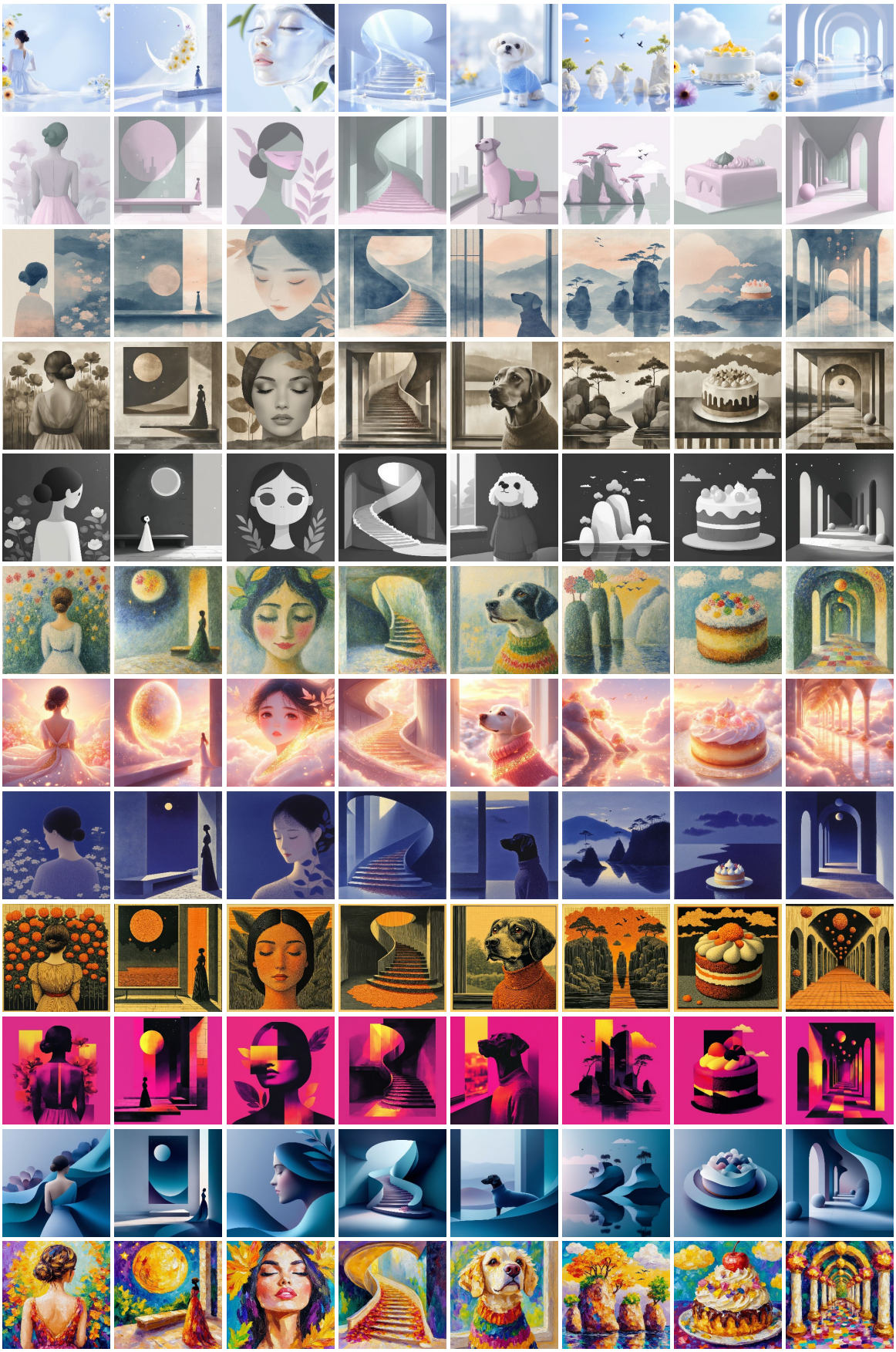

Figure A5. More visual results of CoTyle.

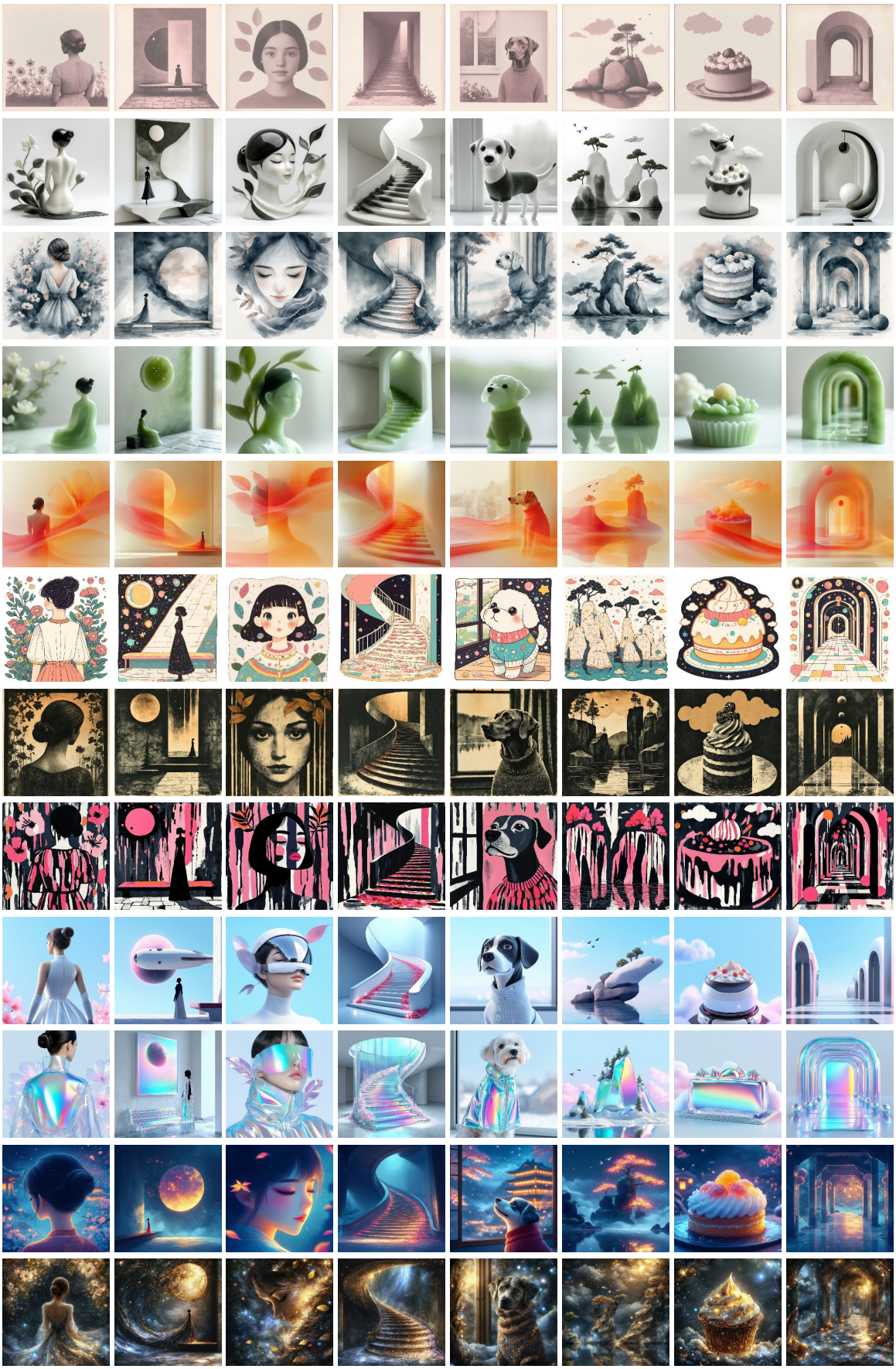

Figure A6. More visual results of CoTyle.

Figure A7. More visual results of CoTyle.

Table A2. Prompts used in visual figures.

id	prompt
1	“A woman in a dress, with her hair tied in a bun, faces away from the viewer. The background is composed of floral patterns.”
2	“The image shows an artwork featuring a silhouette of a woman in a long dress, standing next to a bench, with an abstract painting in the background containing a large circular object that appears to be the moon.”
3	“The image shows a portrait of a woman, with her face obscured by leaves and a few leaves decorating the background.”
4	“The image shows a spiral staircase with light streaming down from the top, illuminating the staircase’s surface. The staircase is located inside a large building, with the walls and ceiling in the background.”
5	“The image shows a dog wearing a sweater, standing by a window, with the scenery outside the window blurred in the background.”
6	“The image shows a natural landscape with several uniquely shaped rocks and trees growing in the foreground. In the background is a sky dotted with clouds and two flying birds. The rocks and trees are clearly reflected in the water.”
7	“The image shows a cake covered in buttercream and candies, with clouds in the background.”
8	“The image shows a corridor made of arches and spheres, with a floor made of blocks and a background.”