LineArt: A Knowledge-guided Training-free High-quality Appearance Transfer for Design Drawing with Diffusion Model

Supplementary Materials

Overview of the Supplementary Materials

- Sec 1: Background research covering relevant work.
- Sec 2: Presents three core insights about lines, guiding single/double lines, and soft edges design for structural and texture consistency.
- Sec 3: Describes experimental optimizations for the appearance transfer stage, including patch size selection and base layer shaping for better highlight perception.
- Sec 4: About data sources, construction process, and selection criteria for the ProLines dataset.
- Sec 5: Covers eight metrics, SOTA experimental configurations, user study 5.3, detailed explanation of ablation experiments 5.4, and additional qualitative experimental results 5.5.
- Sec 6: Discusses limitations and outlines future improvement directions.
- Sec 7: To improve the clarity and completeness of our work, we provide additional analyses at the end of the supplementary materials based on the reviewers' valuable suggestions.

1. Related Work

Sketch-to-Image (S2I) Generation. S2I generation aims to transform sketches into photorealistic images. Early work focused on sparse, abstract sketches, treating S2I as a domain transfer task [5, 13]. Traditional GAN-based models improved generation through contextual loss [25], multistage generation [16], or by mapping sketches to latent spaces of pre-trained GANs [21, 23, 31, 38]. Recently, diffusion-based methods have gained popularity. For example, [39] maps sketches into latent spaces of pre-trained diffusion models, while SDEdit [26] adds noise to sketches and denoises them based on textual prompts. SGDM [36] ensures that noisy images align with sketches. Multi-conditional frameworks like ControlNet [42] and T2I-Adapter [27] use additional inputs (e.g., depth maps, color palettes) to enhance control over generated images. CoGS [18] employs supervised learning with image-sketch pairs to generate high-quality images from rough sketches. In contrast, unsupervised methods like [3] do not require paired training data, offering greater flexibility for handling sketches of different abstraction levels and real photos. Our work focuses on translation tasks that maintain close structural consistency with the source image, ensuring key structural elements remain unchanged while selectively modifying other aspects.

Control and Guidance in Diffusion Models. Diffusion models have garnered significant attention in the generative modeling domain. Researchers have been exploring image-guided stylization using diffusion models to achieve personalized results [8, 20, 40]. Early methods in the mid-1990s used brush strokes for stylistic effects. Then, Neural Style Transfer [14] and [35] laid the foundation for modern stylization, although initially limited to a single style. [15] introduced a style prediction network to support multiple styles. In 2022, [10] demonstrated the superior performance of transformers in stylization. Since 2023, more diffusion-based stylization methods have appeared [19, 29, 32, 44]. These methods leverage the extensive prior knowledge to interpret and manipulate both structural and artistic elements in images.

2. Our Three Core Insights about Lines

2.1. Insights 1: single lines

We hold the perspective that the pivotal line dictating an object's three-dimensional form and surface segmentation should be perceived as a singular, continuous, and closed loop. This view is rooted in the common practice of sketching, where contour lines are primarily employed to delineate the object from its background. Subsequently, the intricate interplay of these lines, through their intersections and spatial relationships, articulates the boundaries of each facet of the object.

Building on this perspective, we conducted experiments to evaluate the role of these pivotal lines in maintaining the integrity of surface segmentation and appearance transfer, as shown in Figure 5 of the main text. Specifically, we applied morphological filtering to intentionally disrupt the regional segmentation of line drawings, effectively breaking the closed and continuous nature of these lines. The results revealed a significant impact on the generated images: colors tend to be consistent in the same area, while distinct regions displayed noticeably different colors. This observation highlighted the essential role of continuous and closed lines in preserving the spatial coherence of color and surface boundaries during generation.

2.2. Insight 2: double lines

Recognizing that human attention during the drawing process is unevenly distributed, leading to variations in stroke thickness and emphasis, we propose that the texture and details in line drawings should be represented as prominent

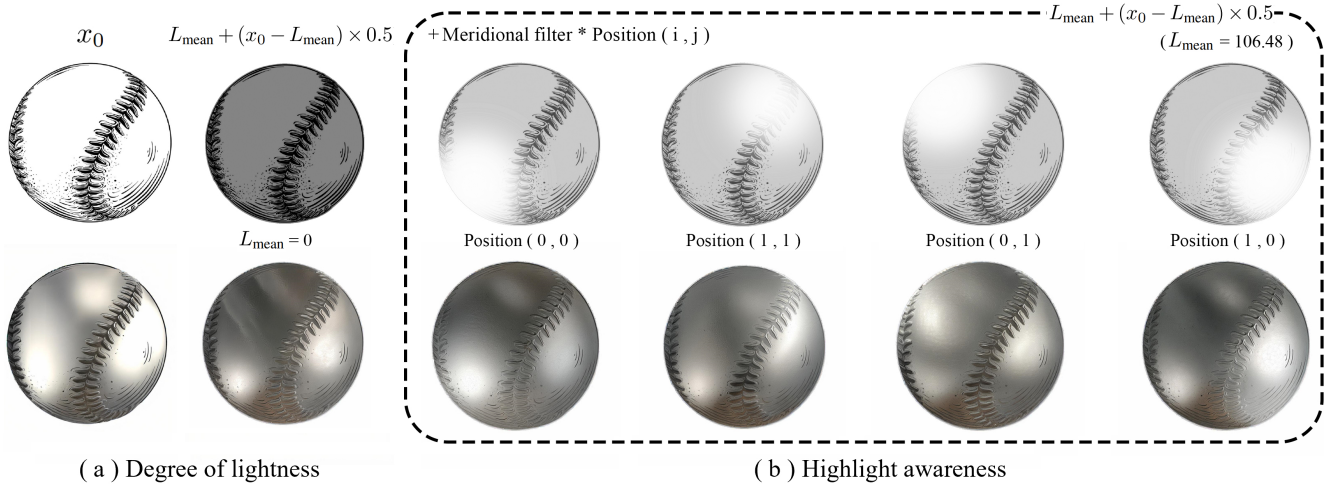


Figure 1. **Base layer shaping for overall brightness and highlight perception.** (a) Adjusting the grayscale distribution of the initial input can effectively control the overall brightness and darkness of the generated results. (b) By applying various radial filtering effects on base layer shaping, highlight awareness can be achieved, enabling nuanced control over the placement and intensity of highlights in the output.

double-line structures. These double lines are envisioned as continuous and narrow regions, rather than simple single strokes. Compared to single lines, double lines attract greater generative attention from the model. When these double lines form closed loops, they naturally define distinct regions that can exhibit clear color differentiation, enabling the creation of perceptual boundaries between adjacent areas. This method not only enhances the model's ability to handle complex textures but also provides finer control over the thickness and density of patterned regions, ensuring a more precise and visually compelling result, as shown in main Figure 12(a): Double lines.

2.3. Insight 3: soft edges

By performing frequency analysis on the real material image, we observe that lines serve a dual purpose in visual representation: not only do they imply spatial gradient relationships, but their detailed depiction also conveys material characteristics. This unique property of lines makes them invaluable for capturing subtle transitions and texture variations. To leverage this, we propose using a combination of points and lines to regulate spatial transitions. These elements act as high-frequency soft constraints, effectively bridging the structural and textural information. By incorporating these constraints, we aim to guide the subsequent appearance generation process, ensuring smoother transitions and more faithful material representation in the final output.

3. Analysis for Appearance Transfer

3.1. Patch Size

In the original text, we described removing the background from the appearance image and segmenting the remaining textured regions into multiple patches to optimize texture features. Specifically, the process involves removing the background of the reference appearance image, retaining only the valid pixel area, dividing it into smaller patches, and reassembling them into a new texture reference image. This method effectively eliminates the influence of redundant background information on the generated results. By carefully selecting the patch size, it also preserves the light and shadow textures of the original appearance image, thereby enhancing the material embedding features generated by the image encoder.

As shown in main Figure 12(c): Patch size, comparative experiments demonstrate that this method significantly improves the controllability of texture and color. However, the patch-based method may disrupt the spatial consistency of the texture. For example, using excessively small patch sizes can result in outputs that lack continuity and glossiness, as the network struggles to reconstruct the original texture relationships, leading to suboptimal outcomes. To address this limitation, future work could explore replacing the patch-based method with more robust texture synthesis algorithms that maintain spatial coherence. While this direction holds promise, it is beyond the scope of this paper.

3.2. Highlight Awareness in Base Layer Shaping

The shaping of the base layer plays a pivotal role in guiding perception, influencing not only overall brightness and darkness adjustments but also highlights. As illustrated in Figure 1, modifying the grayscale distribution of the initial input or applying radial filtering to different positions of the image effectively controls how light interacts with the surface. This method creates smooth transitions and enhances visual depth. Moreover, it preserves the fidelity of line structures while emphasizing the dynamic interplay be-

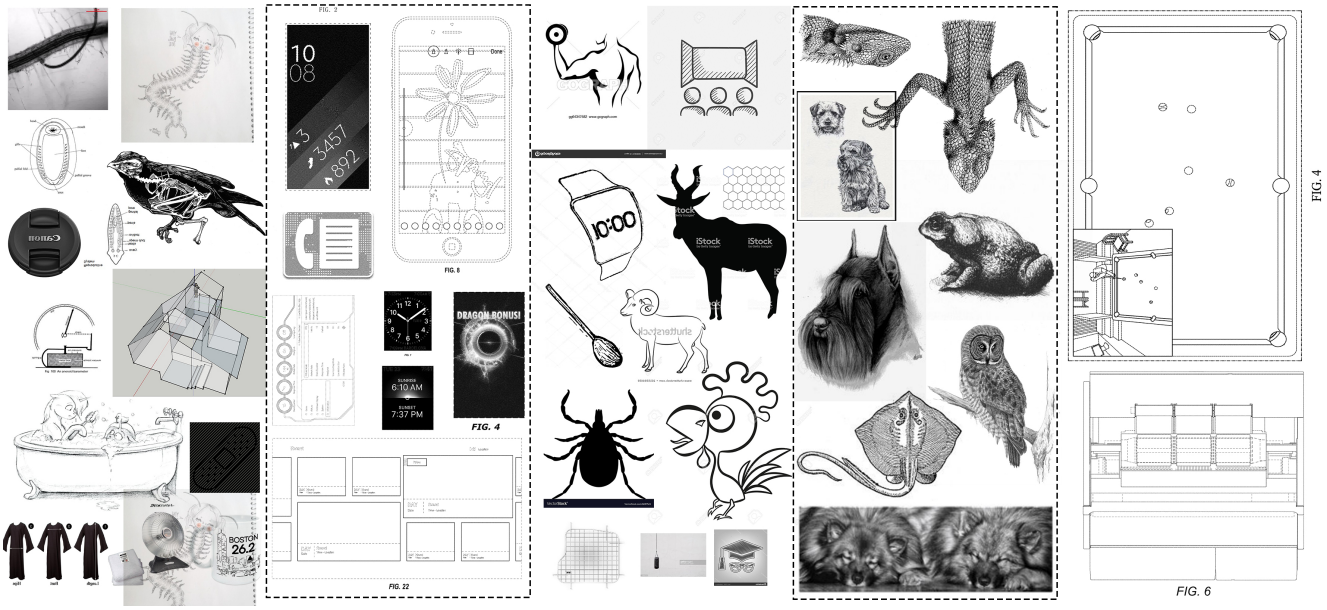


Figure 2. **Five types of deleted data displays.** (1). Items not meeting task definitions, such as anatomical sketches, X-rays, 3D models, watermarked images, or off-center objects. (2). Non-physical items like web pages, interface designs, and fonts. (3). Oversimplified, non-3D images with poor visibility due to viewing angles, including deformed/user sketches with errors or redundant strokes. (4). Sketches with extensive light/shadow textures. (5). Non-single objects like interior scene images.

tween textures and highlights, ensuring the generated output aligns seamlessly with artistic intent and material properties. Such meticulous attention to the base layer is crucial for achieving realistic and visually compelling results in design and rendering tasks.

4. ProLines Dataset

4.1. Knowledge-guided data collection

As the first work focusing on object-centric design drawings and photo-realistic synthesis of appearance references, we systematically collected and organized currently available public line drawing datasets during the evaluation experiment phase. In addition, previous research on sketches has not clearly defined and classified line drawings. Therefore, we designed a screening solution and selected 5101 design drawings from four datasets: Bronze, DifferSketching [41], ImageNet-Sketch [37] and DeepPatent [24].

Specifically, although there are currently multiple manually drawn sketch datasets, they are usually too abstract and unverified (such as Sketchy [33], QuickDraw [22], Pseudosketches [18], TU-Berlin [11], QMUL-Shoe/Chair [34], and OpenSketches [17]), or contain too much information, such as sketches with fine texture and lighting information. In order to ensure the professionalism of the generated results, we selected four datasets as the main data sources: 1. Bronze: We collected over three thousand bronze images from four published archaeology books, and sorted out 3297 images as the *Bronze* dataset. All the line drawings in this dataset were manually created by archaeology professionals. Given the stringent and precise requirements of archaeological tasks for artifact line drawings, this dataset is particularly well-suited to meet the demands of our research. 2. DifferSketching [41]: Contains 362 expert-level line drawings verified by multiple people, which also meets our task positioning. 3. ImageNet-Sketch [37]: Built on Google Image search "sketch of __", where "__" is the standard category name, the line drawings are widely distributed in types and complexity, with a total of more than 50,000, so it also won our favor. 4. DeepPatent [24]: This is a large-scale patent dataset centered on objects, containing more than 350,000 patent images.

However, the common problem of these four datasets is that the complexity of images is not uniform. For example, the DeepPatent patent dataset also contains non-physical objects such as web pages and written layouts. There are also a large number of 3D model views and line drawings that do not meet the requirements of the dataset; ImageNet-Sketch has a large amount of duplicate data and overly complex scene sketches. Thus, it is necessary to clean the dataset.

We developed a set of semi-automatic screening method based on image complexity. As shown in Figure 10 of the main text, we use ICNet [12] as a metric for quantifying the complexity of line drawings, all images in the dataset are scored for complexity, so that the complexity of each image is quantified as a specific value between 0 and 1. Then, we write the image names and corresponding complexity scores of the four datasets into four tables and sort them by score to obtain the normal distribution diagram of the

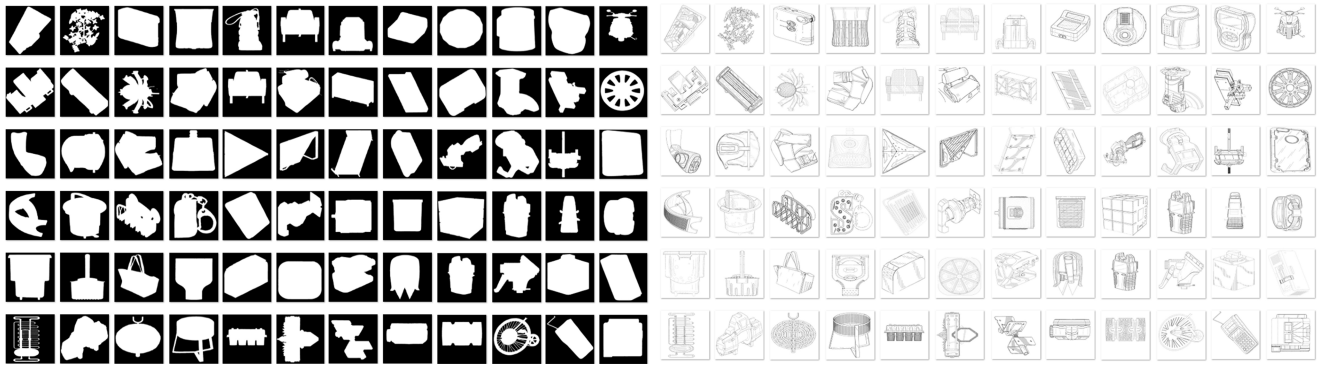


Figure 3. **The result of dataset preprocessing.** Mask and original line graph with background noise removed.

complexity of each of the four datasets. The line drawings we need should not be too simple or too complex. Therefore, we first focus on the threshold interval where the distribution of noise images is recorded, and then avoid these noise values and try to select the central threshold interval with a higher proportion of ideal line drawings. After a lot of experimental attempts, this method helped us to initially eliminate most of the inappropriate line drawings.

Subsequently, we invited 6 computer professionals with CV/CG backgrounds to manually screen these images. The data types that are filtered out are (as shown in Figure 2): 1. Does not meet our task object definition. For example, some of the line drawings are anatomical sketches, biological illustrations, X-ray images, 3D models, black-and-white product photos, background noise, images heavily obscured by watermark text, or images not centered on the object. 2. Non-physical objects such as web pages, interface designs, and fonts. 3. Images that are too simple, not three-dimensional, and difficult to see the actual meaning due to inappropriate viewing angles. For example, the object belongs to a simple drawing/deformation/user sketch, with too many errors and redundant strokes. 4. Sketches containing a lot of light and shadow textures to depict details. 5. Non-single objects such as interior decoration scene pictures.

Through the above process, we finally built a professional design drawing dataset that met our task requirements and had a considerable number of images. In the end, the screening results on each dataset were as follows:

- The threshold interval of the Bronze dataset was $0.2576 \sim 0.2903$, and 380 images were screened.
- The threshold interval of the DifferSketching dataset was $0.0461 \sim 0.2165$, and 362 images remained. Although DifferSketching is relatively simple overall, since professional design line drawing data is difficult to obtain, we still retain the dataset as a test of our work on simple but accurate line drawing objects.
- The threshold interval of the ImageNet-Sketch dataset was $0.2500 \sim 0.2650$, and 1756 images were screened.
- The threshold interval of the DeepPatent dataset was $0.2715 \sim 0.2790$, and 2603 images were screened.

4.2. Data preprocessing

After screening the ideal line drawing data for complex images, We built an automated workflow for line drawing mask extraction, background masking, and size cropping for subsequent generation and evaluation experiments. In our data preprocessing process, we first perform edge detection on the input line drawing, and use methods such as the Canny algorithm to extract image edges. The extracted edge map is input into ControlNet as a control signal, and then embedded into the finely tuned Stable Diffusion model. Then, we deployed the BiRefNet [45] to perform mask extraction and cropping of foreground objects on the generated RGB image, and then used the obtained mask to perform further background masking and size cropping on the original line drawing, as shown in Figure 3. After the mask and cropped line drawing generated by the workflow were completed, each image was manually reviewed and proofread by 3 reviewers with a computer professional background to ensure that its quality met the standards for the use of the dataset. Through this process, we successfully constructed a high-quality test dataset, providing a complete process for line drawing classification and processing that can be referenced for subsequent related research.

4.3. Appearance Image Collection Process

To conduct qualitative and quantitative analyses, we utilized the ProLines dataset as the structural basis and supplemented it with a collection of 61 appearance images. To ensure the meaningfulness of the generated results, we categorized the objects in the ProLines dataset into 9 distinct groups based on their characteristics:

1. **Sculptures (362 images)**: Derived from the full DifferSketching dataset.
2. **Bronze Artifacts (380 images)**: Entire Bronze dataset.
3. **Patent Objects (2603 images)**: Entire DeepPatent dataset.
4. **Fur Textures (612 images)**: Includes animals like cats, dogs, and sheep from ImageNet-Sketch, which exhibit prominent fur textures.

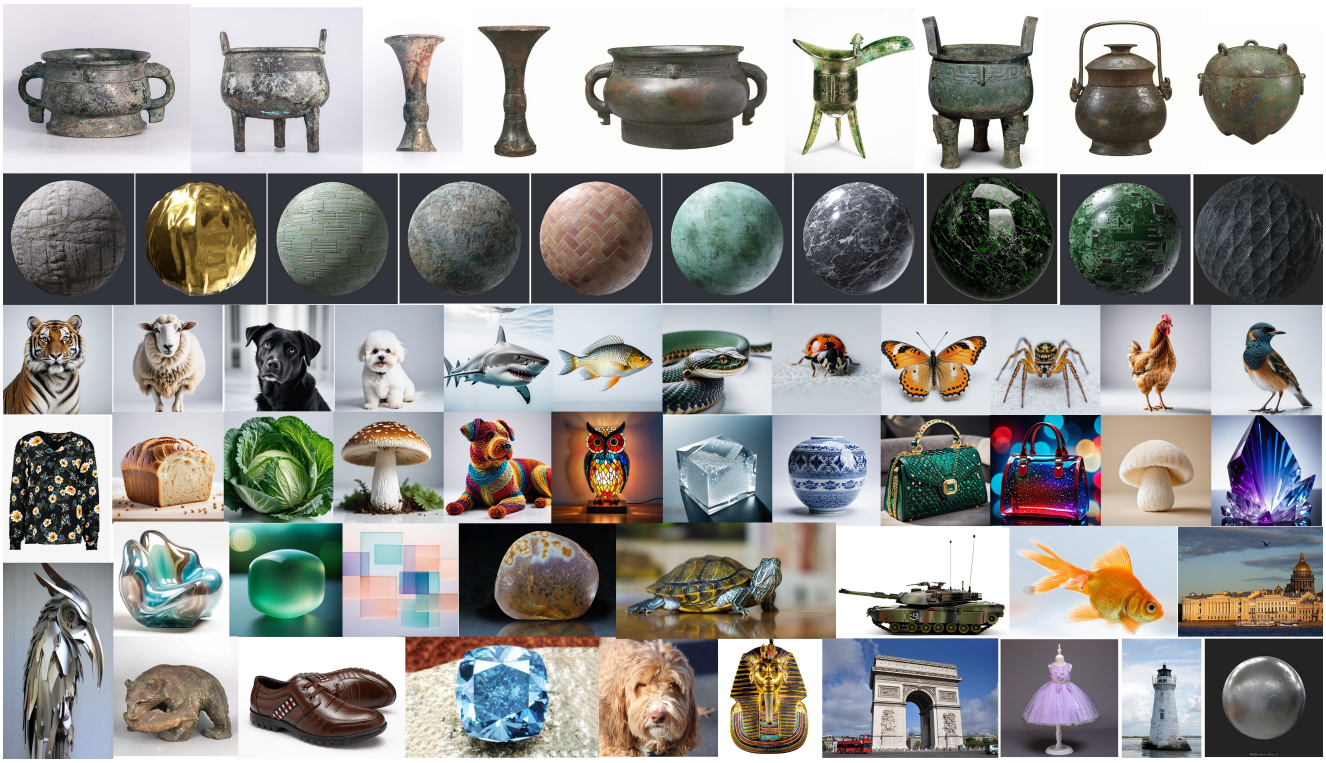


Figure 4. **61 appearance images.** The appearance reference pictures we collected.

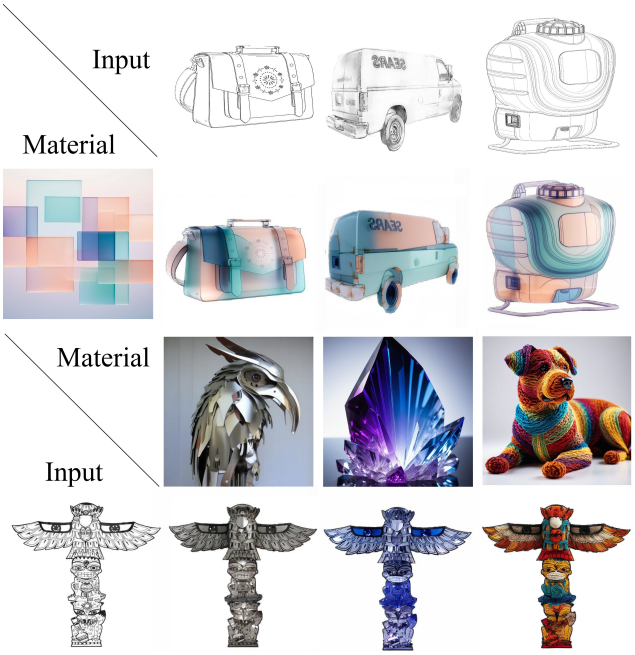


Figure 5. **Other examples of our method in assisting design.** Design tasks require matching a specified material with a suitable object, or finding a suitable appearance-rendering material for a specified design drawing. Our method can generate corresponding rendering effects based on input design drawings and material references.

5. **Scaled Textures (187 images)**: Animals such as fish, snakes, and frogs from ImageNet-Sketch, are characterized by smooth, scale-like textures.
6. **Feathered Animals (162 images)**: Birds and avian species from ImageNet-Sketch.
7. **Smooth Textures (118 images)**: Insects and spiders from ImageNet-Sketch, noted for their polished surfaces.
8. **Simple Everyday Objects (538 images)**: Non-living daily objects from ImageNet-Sketch.
9. **Food (139 images)**: Includes items such as bread, fruits, and vegetables from ImageNet-Sketch.

For each category, we collected 4 to 10 appearance images to serve as texture and material prompts during the image generation process. This ensured high-quality outputs while maintaining sufficient complexity in the experimental tasks. We adopted three primary strategies to gather the appearance images: 1. Professional Archaeological Resources: We curated 9 high-quality real bronze artifact photographs from published professional archaeology books. 2. Existing public datasets and website: We selected 27 images from VITON-HD [7] ImageNet [9] and publicly available PBR material websites. 3. Design-Inspired Renderings: We sourced real design rendering examples from design platforms such as Behance and Pinterest. Using these examples as inspiration, we crafted textual descriptions of 25 specific design scenarios and generated corresponding appearance images with DALLE-4K.

Through this process, we compiled a compact yet diverse testing set of 61 appearance images as shown in Figure 4. All subsequent qualitative and quantitative experi-

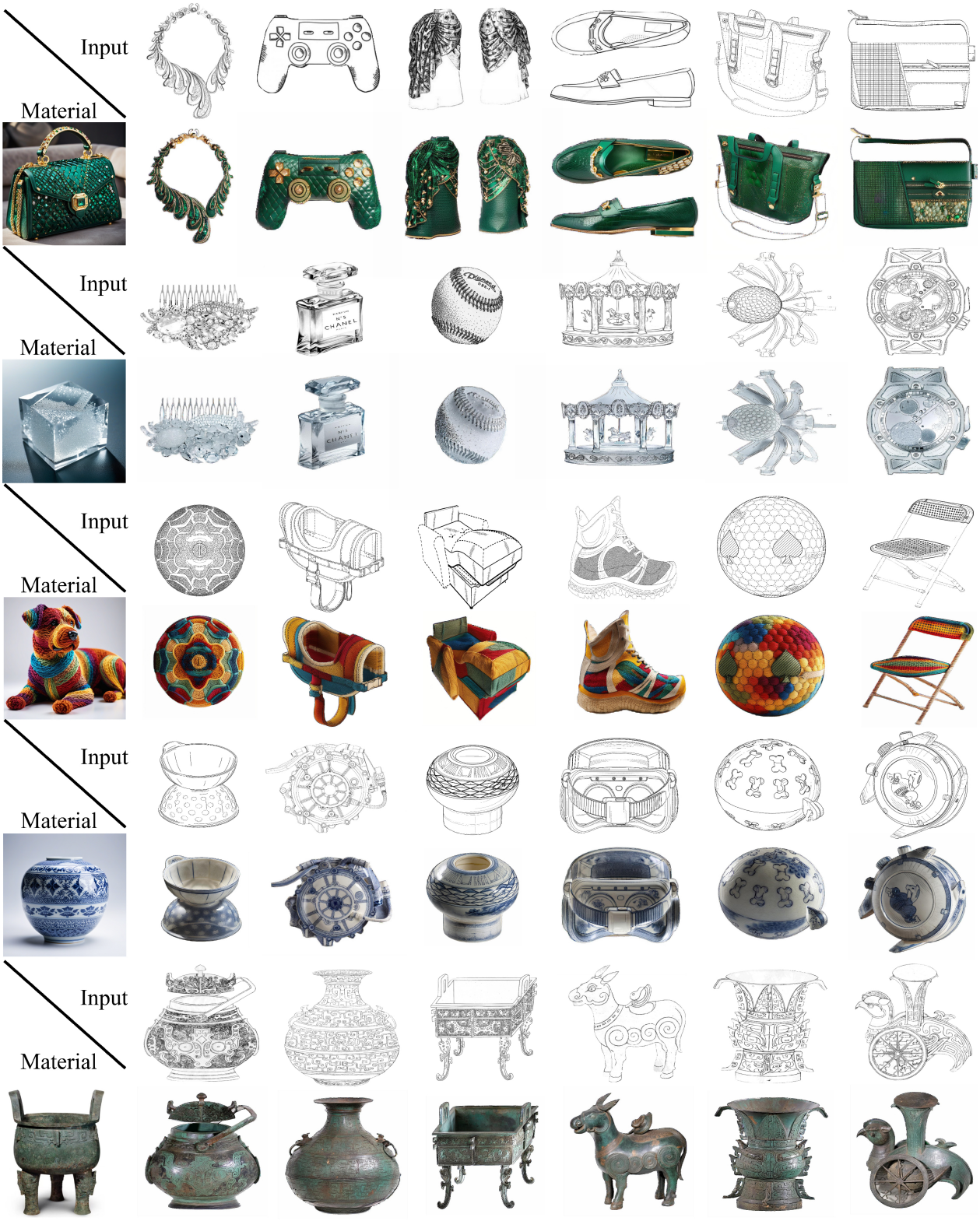


Figure 6. **More results in design tasks using our work.** Each task consists of an input line drawing, which represents the final design, and a material reference image, which specifies the target material. These tasks cover a variety of design fields, from jewelry to household items to clothing, and demonstrate the generation capabilities of our work across different design objects and multiple materials.

Figure 7. **Examples of professional sketch generation results of animals by our method.** We strictly screened the complexity and selection of line drawings, but in experiments, we found that our method also performs well for professional sketches with complexity beyond the expected level.

Figure 8. **Example of user abstract hand-drawn generation results by our method.** As for abstract user hand-painted pictures, we show the effect of material transfer.

ments were conducted using the ProLines dataset combined with this appearance image test set, ensuring a rigorous and comprehensive evaluation of our method.

5. Experiments Details

5.1. Metrics

We evaluate the generated results from three dimensions: edge fidelity, appearance transfer perception, and overall perceptual quality. In terms of edge-fidelity, we use two metrics: Structural Similarity Index (SSIM) and Chamfer Distance (CD). SSIM is used to measure structural similarity, which quantitatively evaluates the performance of the generated image in preserving the details of the original line. The CD further quantifies the degree of alignment between the generated line and the original contour, ensuring the accurate capture of details. To evaluate the quality of appearance transfer, we introduce three metrics: Gray-level Co-occurrence Matrix (GLCM), Peak Signal-to-noise Ratio (PSNR) [30], and Color Histogram (CH) [1] loss. GLCM can capture the texture similarity between the generated image and the reference texture, so as to deeply understand the effectiveness of appearance transfer. PSNR measures fidelity at the pixel level, while the color histogram feature quantifies the similarity of color distribution, which helps to evaluate the degree of matching of the generated result with the target appearance image in terms of color features.

In the evaluation of overall perceptual quality, we used Fréchet Inception Distance (FID), Learned Perceptual Patch Similarity (LPIPS) [43], and Contrastive Language Image Pre-training (CLIP). FID evaluates the naturalness of the generated results by measuring the statistical similarity between the generated images and the real images. LPIPS evaluates the perceptual similarity, which is closely related to human visual perception. CLIP evaluates the consistency between the visual content and semantic consistency with the reference, ensuring that the generated images are not only visually realistic, but also semantically consistent with the expected appearance.

5.2. Setting of SOTAs

We selected these SOTA works for experimental results evaluation: ZeST [6], Cross-Image Attention (CIA) [2], StyleID [8], DreamBooth [32], InstructPix2Pix [4] + IP-

Adapter (InstructPix2Pix'), AesPA-Net [20], ControlNet-Canny + IP-Adapter (Baseline). All works are implemented in PyTorch and run on the Nvidia A-40 GPU with 40 GB of RAM. Among the methods compared, ZeST, Baseline, Ours, and DreamBooth all utilize Stable Diffusion XL Inpainting [28]. Specifically, ZeST employs the corresponding version of depth-based ControlNet alongside IP-Adapter, while both Baseline and Ours leverage the corresponding version of canny-based ControlNet in combination with IP-Adapter. For DreamBooth, we use the official LoRA-DreamBooth provided by Diffusers. CIA and InstructPix2Pix' utilize Stable Diffusion version 1.5. StyleID and AesPA-Net use the official code and version.

5.3. User Study

To evaluate the performance of our method, we conducted a user study with 20 design-background participants to assess its effectiveness in preserving line structures and material appearance, as well as its overall perceptual quality. The study involved a ranking task with three distinct evaluation criteria: edge fidelity, appearance fidelity, and overall perception as shown in Figure 9(b). *Edge Fidelity*: Participants assessed how closely each generated image resembled the line structure reference, focusing on the preservation of black-and-white line drawing details. *Appearance Fidelity*: Participants evaluated the similarity of the generated images to the appearance reference image, paying particular attention to the accuracy of color and texture reproduction. *Overall Perception*: Participants considered the balance between maintaining the characteristics of the line structure reference and the material and texture features of the appearance reference, ranking images based on their overall quality and coherence.

Participants were randomly presented with a total of 8 images per test case. The first two images served as references: a black-and-white line drawing as the line structure reference and a color image as the appearance reference representing texture and material. The remaining 6 images were variations generated by different methods, which the participants were required to rank according to the specified criteria as shown in Figure. And the scoring rule is:

$$\text{Score} = \frac{\sum \text{frequency} \times \text{weight}}{\text{number}_{\text{people}}}$$

Where frequency indicates the number of times a job is selected and ranked in a certain position, the weight indicates the score corresponding to the position (the first place is recorded as 6 points while the lowest rank is recorded as 1 point), and the number indicates the number of people selected. Methods in the options appear in a random order and their names are anonymous. Results showed that our method achieved the highest scores across all three metrics (5.03/6, 5.43/6, 4.83/6) as shown in Figure 9(a).

5.4. Ablation Study

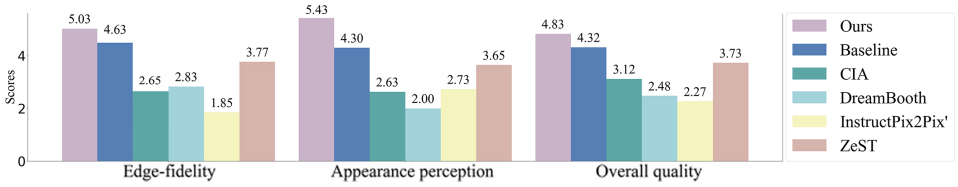
We performed an ablation study to validate the effectiveness of key design choices in our method. Specifically, we evaluated the following four aspects: (1) the role of double lines in controlling the thickness of pattern details; (2) the improvement of texture feature generation by soft edges as high-frequency constraints; (3) the impact of grayscale distribution simulation in base layer shaping on the color of the generated results; (4) the impact of patch size selection in texture synthesis. The experimental results are detailed in Figure 12 of the main text. Although the baseline can achieve the basic sketch-guided generation effect, the generated appearance attributes deviate greatly from the target reference image. In each row of the diagram, we build on the baseline method by incrementally adding our proposed key components.

As shown in Figure 12 of the main text, the introduction of double line operations can accurately control the thickness of the pattern in the line graph and enhance the model's fidelity to structured information. Using soft edges can significantly affect the fidelity and trend of the texture primitive texel. The application of base layer shaping significantly improves the similarity of the generated results to the target appearance image in terms of color and light and dark contrast. In texture synthesis, when the patch size value is set to 150, better texture feature encoding effects can usually be achieved.

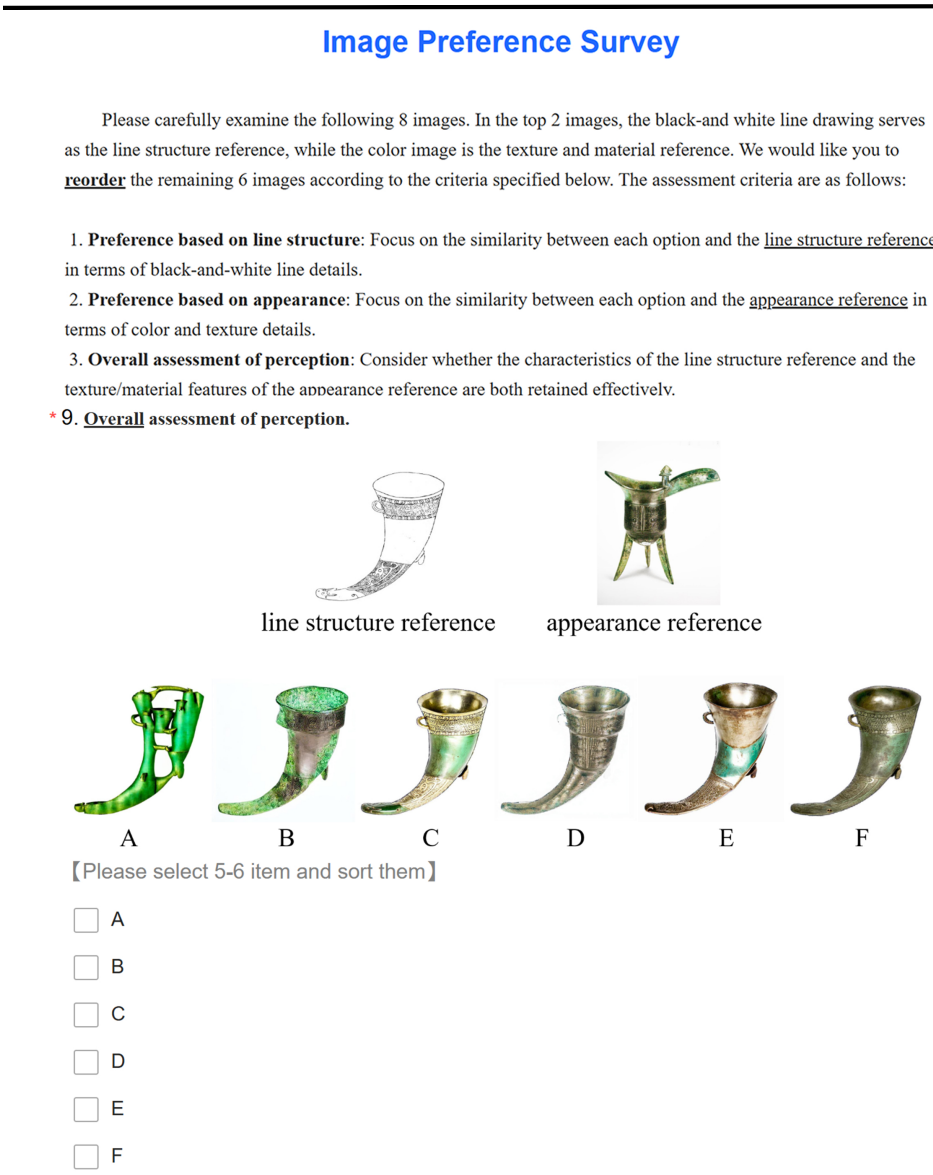
5.5. Additional Qualitative Results

We show more generated results by our method on the design tasks. As shown in Figure 6, our work facilitates rapid and efficient execution of design workflows. Specifically, it can swiftly render design drawings with specified material effects, providing designers with an intuitive way to assess whether the design of a line draft is practical within material constraints. For fixed product line drawings, our method offers the capability to incorporate various material textures into technical drawings, generating diverse material effect previews. This functionality assists designers in identifying the most suitable material combinations for their product line draft designs.

Based on the discussion above, we refined the categorization of current sketch-related tasks and proposed a detailed filtering and preprocessing workflow. Beyond this, we also conducted experiments on data types not ideally suited for this work to explore the robustness of our method. For instance, as shown in Figure 7, we performed animal fur color transfer on professionally drawn animal sketch line art, achieving visually plausible results. Additionally, as illustrated in Figure 8, our method demonstrated impressive generation performance on more abstract and deformed sketches, such as those from user studies, highlighting its adaptability to diverse input styles and levels of abstraction.



(a) Result of User Study



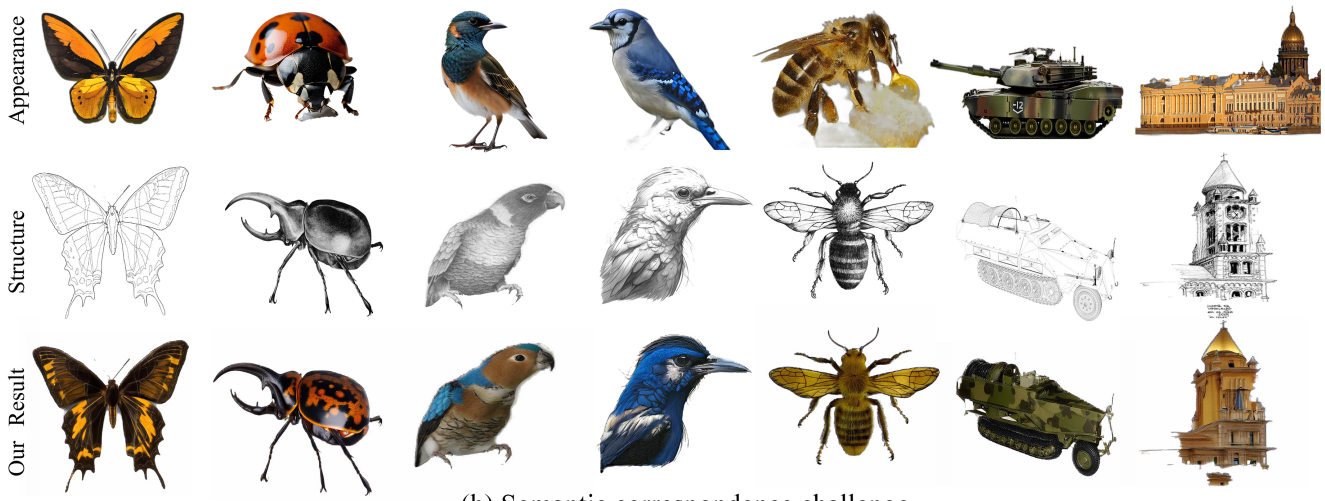
(b) User Questionnaire Overview

Figure 9. (a) Scoring results of user study questionnaire answers. (b) User study questionnaire overview.

6. Limitation

As shown in Figure 7 above, our method also performs well for sketches that contain detailed shadows, but the appearance transfer effect of line drawings that contain too many shadows or objects that have drawn a lot of texture details is still challenging. For example, in column 3 of Figure 10(a), the details of the owl's eyes are lost during the generation process, and the identity consistency is not maintained. In Figure 10(a)-5, the details of the dog's ear sketch conflict with the wool texture, resulting in the dog's long ears degenerating into part of the body in the generated result. In Figure 10(a)-6, the soft edges in our design struggle to preserve the detailed pattern of the cabinet and the clarity of the fish scale appearance image at the same time, resulting

(a) Dealing with objects that have been painted with a lot of texture detail



(b) Semantic correspondence challenge

Figure 10. **Limitation examples.** *Top (a)* shows soft edges struggle to retain both the heavy textures details in structure and the texture in appearance. For example: Owl eye details are lost, affecting identity consistency ((a)-3 column). Wool texture conflicts with the dog's ear sketch, causing the ears to merge with the body ((a)-5 column). *Bottom (b)* shows that we face challenges in the semantic correspondence task. For example, white feathers on the bird's abdomen are omitted due to lack of semantic correspondence ((b)-4)). Patching and shuffling disrupt texture features in the butterfly and camouflage car examples ((b)-5 and 6 columns). Architectural details like windows struggle to maintain intra-class consistency during generation ((b)-7 column).

in a conflict between the two.

Therefore, we would like to emphasize that we focus more on dealing with the complexity of key lines, especially when the detailed structural lines are correlated to textures. Our collection and processing of the dataset reflect our attempt to subdivide the sketch task based on image complexity, which is different from other challenges. In the processing of ideal target objects, the challenge we face is mainly the inability to control the precise semantic correspondence. For example, in the fourth column of Figure 10(b), the white feathers on the bird's abdomen are not preserved due to the lack of semantic correspondence. In Figure 10(b)-6, about the appearance transfer example of the camouflage car, the corresponding texture features are destroyed due to the patching and shuffling operations. In the last column of Figure 10(b), for the structural details in the architectural sketches such as windows, we struggle with intra-class consistency during the generation process. These challenges highlight the need for further work in handling complex details and semantic correspondence.

7. Additional Analysis

To further enrich the interpretation of our work, we have included an analysis of the reviewers' key comments at the

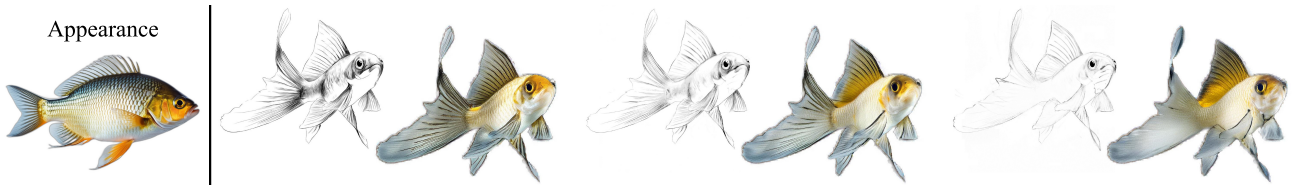


Figure 11. **Generated results by decreasing strokes.**

Table 1. **Experiments on the depth estimation module.** Replacing the Depth estimation module has no significant effect on the results.

DifferSketching IC: 0.0461-0.2165	FID↓	LPIPS↓	$CLIP_i$↑	PSNR↑	CH↓	GLCM↓	SSIM↑	CD↓
LineArt w/ Depth Anything	211.37	**0.23**	**0.83**	19.25	**0.62**	**1.24**	**0.96**	2.48
LineArt w/ MiDaS	**183.39**	**0.23**	0.82	**19.28**	0.64	2.08	**0.96**	**1.97**
DreamBooth (Best SOTA)	202.52	0.24	0.77	18.88	0.76	37.48	0.95	30.93

end of the supplementary materials. We believe that the consideration and response to these questions can improve the clarity and completeness of our study.

7.1. Experiment of Strokes Decrease

We show an example as shown in Figure 11 where the number of strokes progressively decreases. It shows that our work can still maintain stable generation results when the number of strokes is decreasing.

7.2. Depth Estimation Module

We replaced Depth Anything (2024) with Midas (2019) in LineArt, and the results (in Table 1) show that performance remains stable, indicating minimal impact from the depth estimation choice.

7.3. A Novel Perspective on LineArt: Graphics Rendering

To further explain our ideas, we draw on the key comments of anonymous reviewers and provide an analogy for LineArt from another perspective: the traditional graphics rendering pipeline.

After the analysis of the main text, we decompose the input line drawings into three levels, including continuous single lines for area division, double lines for emphasizing local details, and soft edges representing spatial transition relationships and high-frequency features of textures. This design can be compared to the geometry stage in the rendering pipeline, which uses single lines and double lines to define the shape and contour information of the model. Soft edges are similar to bump mapping, which is used to enhance surface details so that the generated appearance has more refined appearance features.

In the texture synthesis process of LineArt, the appearance modeling method based on the diffusion model is adopted, which is similar to the shading process in the rendering pipeline. Specifically, base layer shaping is regarded as the setting of the basic color of the surface of the object, which can be compared to the control effect of PBR (physically based rendering) on the texture material. Surface layer coloring is similar to texture shading in rendering. By accurately modeling the texture material information, the naturalness and authenticity of the final visual effect are ensured.

7.4. Specific Parameter Settings

Dilation and erosion are basic morphological operations, with core parameters typically having default configurations in many publicly available algorithms and tools (e.g., OpenCV).

Figure 5 of the main text shows how to control the continuity and closure of the region in the original line image by controlling the morphological filter. Taking the erosion filter in the third row as an example, the specific process is to create a 5x5 two-dimensional array with all elements set to 1 as the erosion operator and then erode the foreground object (usually the white area) of the input image. The erosion operator slides on the image to remove bright (white) objects in the image that are smaller than the structural element. By comparing the detection results obtained based on the Canny algorithm in the first and third rows of the fourth column (the upper and lower limits are 100 and 200, respectively), it can be clearly seen that the image contour structure obtained by the erosion operator is visually more concise and intuitive. In our experiment, we simply arrange the acquisition of single lines as follows: we obtain them by performing Canny edge detection on the mask image. This process can independently decide whether to apply a structure element with a shape of (5, 5) and all elements with values 1 to erode the result to ensure alignment with the double-edge result.

As for the implementation of the double edge, the structural pattern in the image is first emphasized by the opening operation (combining the erosion operator with a kernel size of 5 and the dilation operator with a size of 3). Then

the Canny edge detection (same as above) is used to obtain a double line effect that enhances the visual hierarchy and continuity.

In our experiment, to obtain soft edges, we created a Numpy array of shapes (1, 2) as the structure element of the erosion operation with all elements set to 1. The erosion operation will reduce the edge part of the G_{init} image after Canny edge detection (the two threshold parameters are 25 and 50, respectively) according to this structure element, and the result will serve as a high-frequency soft constraint for subsequent texture generation and spatial transition. We use the above basic kernels and keep them unchanged in the experiment.

References

[1] Mahmoud Afifi, Marcus A Brubaker, and Michael S Brown. Histogan: Controlling colors of gan-generated and real images via color histograms. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7941–7950, 2021. 8

[2] Yuval Alaluf, Daniel Garibi, Or Patashnik, Hadar Averbuch-Elor, and Daniel Cohen-Or. Cross-image attention for zero-shot appearance transfer. In *ACM SIGGRAPH 2024 Conference Papers*, pages 1–12, 2024. 8

[3] Dina Bashkirova, Jose Lezama, Kihyuk Sohn, Kate Saenko, and Irfan Essa. Masksketch: Unpaired structure-guided masked image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1879–1889, 2023. 1

[4] Tim Brooks, Aleksander Holynski, and Alexei A Efros. Instructpix2pix: Learning to follow image editing instructions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18392–18402, 2023. 8

[5] Shu Yu Chen, Wanchao Su, Lin Gao, Shihong Xia, and Hongbo Fu. Deepfacedrawing: Deep generation of face images from sketches. *ACM Transactions on Graphics*, 39(4): 72, 2020. 1

[6] Ta-Ying Cheng, Prafull Sharma, Andrew Markham, Niki Trigoni, and Varun Jampani. Zest: Zero-shot material transfer from a single image. In *European Conference on Computer Vision*, pages 370–386. Springer, 2025. 8

[7] Seunghwan Choi, Sunghyun Park, Minsoo Lee, and Jaegul Choo. Viton-hd: High-resolution virtual try-on via misalignment-aware normalization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 14131–14140, 2021. 5

[8] Jiwoo Chung, Sangeek Hyun, and Jae-Pil Heo. Style injection in diffusion: A training-free approach for adapting large-scale diffusion models for style transfer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8795–8805, 2024. 1, 8

[9] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009. 5

[10] Yingying Deng, Fan Tang, Weiming Dong, Chongyang Ma, Xingjia Pan, Lei Wang, and Changsheng Xu. Stytr2: Image style transfer with transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11326–11336, 2022. 1

[11] Mathias Eitz, James Hays, and Marc Alexa. How do humans sketch objects? *ACM Transactions on graphics (TOG)*, 31 (4):1–10, 2012. 3

[12] Tinglei Feng, Yingjie Zhai, Jufeng Yang, Jie Liang, Deng-Ping Fan, Jing Zhang, Ling Shao, and Dacheng Tao. Ic9600: a benchmark dataset for automatic image complexity assessment. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(7):8577–8593, 2022. 3

[13] Chengying Gao, Qi Liu, Qi Xu, Limin Wang, Jianzhuang Liu, and Changqing Zou. Sketchycoco: Image generation from freehand scene sketches. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5174–5183, 2020. 1

[14] Leon A Gatys. A neural algorithm of artistic style. *arXiv preprint arXiv:1508.06576*, 2015. 1

[15] Golnaz Ghiasi, Honglak Lee, Manjunath Kudlur, Vincent Dumoulin, and Jonathon Shlens. Exploring the structure of a real-time, arbitrary neural artistic stylization network. *arXiv preprint arXiv:1705.06830*, 2017. 1

[16] Arnab Ghosh, Richard Zhang, Puneet K Dokania, Oliver Wang, Alexei A Efros, Philip HS Torr, and Eli Shechtman. Interactive sketch & fill: Multiclass sketch-to-image translation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1171–1180, 2019. 1

[17] Yulia Gryaditskaya, Mark Sypesteyn, Jan Willem Hoftijzer, Sylvia Pont, Frédo Durand, and Adrien Bousseau. Opensketch: A richly-annotated dataset of product design sketches. *ACM Transactions on Graphics (Proc. SIGGRAPH Asia)*, 38, 2019. 3

[18] Cusuh Ham, Gemma Canet Tarres, Tu Bui, James Hays, Zhe Lin, and John Collomosse. Cogs: Controllable generation and search from sketch and style. In *European Conference on Computer Vision*, pages 632–650. Springer, 2022. 1, 3

[19] Mark Hamazaspyan and Shant Navasardyan. Diffusion-enhanced patchmatch: A framework for arbitrary style transfer with diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 797–805, 2023. 1

[20] Kibeom Hong, Seogkyu Jeon, Junsoo Lee, Namhyuk Ahn, Kunhee Kim, Pilhyeon Lee, Daesik Kim, Youngjung Uh, and Hyeran Byun. Aespa-net: Aesthetic pattern-aware style transfer networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 22758–22767, 2023. 1, 9

[21] Youngjoo Jo and Jongyoul Park. Sc-fegan: Face editing generative adversarial network with user's sketch and color. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1745–1753, 2019. 1

[22] Jonas Jongejan, Henry Rowley, Takashi Kawashima, Jongmin Kim, and Nick Fox-Gieg. The quick, draw!-ai experiment. *Mount View, CA, accessed Feb*, 17(2018):4, 2016. 3

[23] Subhadeep Koley, Ayan Kumar Bhunia, Aneeshan Sain, Pinaki Nath Chowdhury, Tao Xiang, and Yi-Zhe Song. Picture that sketch: Photorealistic image generation from abstract sketches. In *Proceedings of the ieee/cvf conference on computer vision and pattern recognition*, pages 6850–6861, 2023. 1

[24] Michal Kucer, Diane Oyen, Juan Castorena, and Jian Wu. Deeppatent: Large scale patent drawing recognition and retrieval. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2309–2318, 2022. 3

[25] Yongyi Lu, Shangzhe Wu, Yu-Wing Tai, and Chi-Keung Tang. Image generation from sketch constraint using contextual gan. In *Proceedings of the European conference on computer vision (ECCV)*, pages 205–220, 2018. 1

[26] Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. Sdedit: Guided image synthesis and editing with stochastic differential equations. *arXiv preprint arXiv:2108.01073*, 2021. 1

[27] Chong Mou, Xintao Wang, Liangbin Xie, Yanze Wu, Jian Zhang, Zhongang Qi, and Ying Shan. T2i-adapter: Learning adapters to dig out more controllable ability for text-to-image diffusion models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 4296–4304, 2024. 1

[28] Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. Sdxl: Improving latent diffusion models for high-resolution image synthesis. *arXiv preprint arXiv:2307.01952*, 2023. 9

[29] Tianhao Qi, Shancheng Fang, Yanze Wu, Hongtao Xie, Jiawei Liu, Lang Chen, Qian He, and Yongdong Zhang. Deadiff: An efficient stylization diffusion model with disentangled representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8693–8702, 2024. 1

[30] Edgar Riba, Dmytro Mishkin, Daniel Ponsa, Ethan Rublee, and Gary Bradski. Kornia: an open source differentiable computer vision library for pytorch. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 3674–3683, 2020. 8

[31] Elad Richardson, Yuval Alaluf, Or Patashnik, Yotam Nitzan, Yaniv Azar, Stav Shapiro, and Daniel Cohen-Or. Encoding in style: a stylegan encoder for image-to-image translation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2287–2296, 2021. 1

[32] Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 22500–22510, 2023. 1, 8

[33] Patsorn Sangkloy, Nathan Burnell, Cusuh Ham, and James Hays. The sketchy database: learning to retrieve badly drawn bunnies. *ACM Transactions on Graphics (TOG)*, 35(4):1–12, 2016. 3

[34] Jifei Song, Kaiyue Pang, Yi-Zhe Song, Tao Xiang, and Timothy M Hospedales. Learning to sketch with shortcut cycle consistency. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 801–810, 2018. 3

[35] Dmitry Ulyanov, Vadim Lebedev, Andrea Vedaldi, and Victor Lempitsky. Texture networks: Feed-forward synthesis of textures and stylized images. *arXiv preprint arXiv:1603.03417*, 2016. 1

[36] Andrey Voynov, Kfir Aberman, and Daniel Cohen-Or. Sketch-guided text-to-image diffusion models. In *ACM SIGGRAPH 2023 Conference Proceedings*, pages 1–11, 2023. 1

[37] Haohan Wang, Songwei Ge, Zachary Lipton, and Eric P Xing. Learning robust global representations by penalizing local predictive power. *Advances in Neural Information Processing Systems*, 32, 2019. 3

[38] Sheng-Yu Wang, David Bau, and Jun-Yan Zhu. Sketch your own gan. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14050–14060, 2021. 1

[39] Tengfei Wang, Ting Zhang, Bo Zhang, Hao Ouyang, Dong Chen, Qifeng Chen, and Fang Wen. Pretraining is all you need for image-to-image translation. *arXiv preprint arXiv:2205.12952*, 2022. 1

[40] Zhizhong Wang, Lei Zhao, and Wei Xing. Stylediffusion: Controllable disentangled style transfer via diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 7677–7689, 2023. 1

[41] Chufeng Xiao, Wanchao Su, Jing Liao, Zhouhui Lian, Yi-Zhe Song, and Hongbo Fu. Differsketching: How differently do people sketch 3d objects? *ACM Transactions on Graphics (TOG)*, 41(6):1–16, 2022. 3

[42] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3836–3847, 2023. 1

[43] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018. 8

[44] Yuxin Zhang, Nisha Huang, Fan Tang, Haibin Huang, Chongyang Ma, Weiming Dong, and Changsheng Xu. Inversion-based style transfer with diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10146–10156, 2023. 1

[45] Peng Zheng, Dehong Gao, Deng-Ping Fan, Li Liu, Jorma Laaksonen, Wanli Ouyang, and Nicu Sebe. Bilateral reference for high-resolution dichotomous image segmentation. *CAAI Artificial Intelligence Research*, 3:9150038, 2024. 4