StyleKeeper: Prevent Content Leakage using Negative Visual Query Guidance

Supplementary Material

1. Related work

1.1. Controlling style with training

Dreambooth [9, 21, 39, 40] and Adapter [23, 42, 60] variants fine-tune a pre-trained diffusion model (DM) with a few images with a same style to completely change the model. Textual inversion variants [10, 13] learn a customized text embedding to use the same model with a minimal extra component. Meanwhile, ControlNet [60] stands out as one of the most effective frameworks to guide structural contents. Similarly, adapter-based methods [52, 59] attach an auxiliary image encoder which receives style reference. Several works[31, 35, 44, 45, 55] train the model to solve image analogy where instructions demonstrate the desired manipulation.

Although these methods often succeed in controlling style, they require a time-consuming additional training procedure or limit the applicable styles within their training set, which hinders practical usage.

1.2. Training-free control with feature manipulation

Manipulating the intermediate features in DMs inherently changes the resulting images even with a frozen DM.

Alaluf et al. [1] and Chung et al. [6] transfer the style of an image to another image via self-attention with DDIM inversion achieving insufficient control. Furthermore, their main goal is image-to-image (I2I) rather than text-to-image (T2I); naive two-stage T2I2I struggles in trade-off between style reflection and content preservation. StyleAligned [16] similarly shares self-attention between two processes but fails to exclude the original style elements because it keeps the original features. Meanwhile, sharing self-attention features improves temporal appearance consistency over multiple frames [54, 57] in video editing task.

Moving forward, we unveil the multiple missing components for accurate control over style and content, producing outstanding results.

On the other hand, Plug-n-Play [46] and MasaCtrl [3] inject self-attention features from one process to another to convey *structure* and *object*, respectively, of the first process, rather than conveying style elements.

1.3. Inverting images to noise for real image editing in diffusion models

Training-free editing methods in the previous subsection manipulate the features of attention layers. If synthetic images with known latent noises are used as reference images, this process is straightforward. However, when editing with real images, the methods require inverting the images into the initial latent noise maps. A majority of diffusion-based editing work [1, 7, 15, 30, 49] and self-attention variants [1, 3, 16] employs DDIM inversion [43] for its deterministic mapping between noise and image. However, DDIM inversion often suffers error at each step from z_{t-1} to z_t and inverted noise from a real image through DDIM does not follow the standard Gaussian distribution [11, 14, 29, 30].

DDPM sampling is an alternative that produces the standard Gaussian noise [18, 19, 28, 53]. However, they require lengthy iterative sampling for inversion or do not distinguish the reference denoising process from the inference denoising process. Compared to these inverting methods, our stochastic encoding consists of a one-step operation and produces statistically aligned intermediate latents.

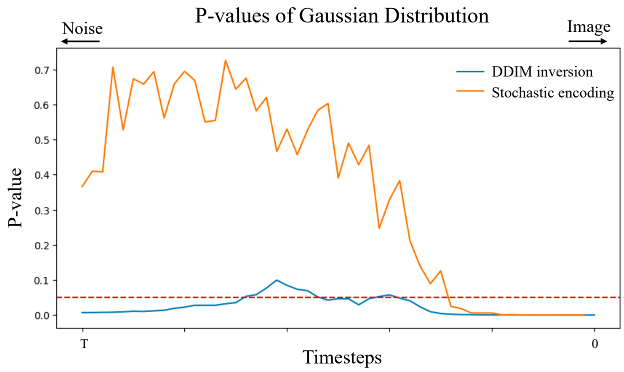
2. Experiments details and metrics

Details We use SDXL [34] as our pretrained text-to-image diffusion model and choose the 24^{th} layer and the after. We also validate our methods on Stable diffusion (SD) v1.5 [38]. Results with SD v1.5 are shown in Figure A8. We set classifier-free guidance as 7.0 and run DDIM sampling with 50 timesteps following the typical setting. The initial noises for a text prompt are identical across competitors for fair comparison. We use the official implementation of IP-Adapter and StyleAligned. For Dreambooth-LoRA[4], we use diffusers pipeline provided by Huggingface. Since the official code of StyleDrop is not available, we use an unofficial implementation[5] provided by Huggingface. All competitors are based on SDXL except StyleDrop (unofficial MUSE). As Dreambooth-LoRA, a training-based approach, requires multiple images, we train the models with five images: the original image and quarter-split patches of the reference image, because using only one image usually leads to destructive results or suffers from overfitting. For IP-adapter, we choose $\lambda = 0.5$ which is the best weighting factor for the task. For the visualization of the attention maps in Figure 6, we average the multi-head attention maps all together along the channel axis at the 20th denoising timestep.

Metrics Following [39, 48], we use DINO (ViT-B/8) embeddings [4] to measure style similarity between a reference image and a resulting image. In addition, we provide a quantitative comparison with Gram matrix [12] to assess style similarity in Table A1. We use CLIP (ViT-L/14)

[4]https://huggingface.co/docs/diffusers/training/lora

[5]https://github.com/huggingface/diffusers/tree/main/examples/amused



(b) Comparing Gaussianity

Figure A1. **Stochastic encoding produces the latents closer to the standard Gaussian distribution compared to DDIM inversion.** A P-value above 0.05 suggests that the data likely follows the standard Gaussian distribution.

embeddings [36] to measure the alignment between text prompts and resulting images. We use LPIPS [61] to measure diversity by average LPIPS between different resulting images in the same text prompt. We use Kolmogorov-Smirnov test [26] to measure gaussianity. For quantitative evaluation and comparison, we prepare 720 synthesized images from 40 reference images, 120 content text prompts (3 contents per 1 reference), and 6 initial noises. The reference images are generated from 40 stylish text prompts. Appendix 4 provides the text prompt set. We also conducted a user study to evaluate the effectiveness of our method in Appendix 5.

Table A1. **Quantitative comparison.** We compare the results for style similarity by utilizing a gram matrix.

Ours	StyleAligned	IP-Adapter	DB-LoRA	StyleDrop
0.791	0.759	0.768	0.759	0.659

3. Algorithm: Visual style prompting with a real image as a reference

Input: Reference latent x_0^{visual}, number of diffusion steps T, color calibration range $[t_{\text{start}}, t_{\text{end}}]$, precomputed constants α_t, noise scale σ_t, model ϵ_θ
Output: Denoised latent x_0

```
Function color_calibration(x_t, x̂_t, x_0^visual):
    x_pred ← (x_t − √(1−α_t)·ε_θ(x̂_t)) / √α_t ;                     // predicted x_0
    x_dir ← √(1 − α_{t−1} − σ_t^2) · ε_θ(x̂_t) ;                      // direction pointing to x_t
    ε ~ N(0, I) ;                                                     // Generate random noise
    x_noise ← σ_t · ε ;                                               // random noise
    x̂_pred ← adain(x_pred, x_0^visual) ;                              // calibrated predicted x_0
    x_{t−1} ← √α_{t−1} · x̂_pred + x_dir + x_noise ;                   // updated x_{t−1}
    return x_{t−1}
Function adain(x, x^visual):
    μ_x, σ_x ← channel_wise_mean_std(x) ;                             // mean and std of x
    μ_x^visual, σ_x^visual ← channel_wise_mean_std(x^visual) ;        // mean and std of x^visual
    x_norm ← (x − μ_x) / σ_x ;                                        // normalize x
    x_adain ← σ_x^visual · x_norm + μ_x^visual ;                      // scale and shift
    return x_adain
Function stochastic_encoding(x, t):
    ε ~ N(0, I) ;                                                     // Generate random noise
    x_t ← √α_t · x + √(1 − α_t) · ε ;
    return x_t
Initialize x_T ← N(0, I);
Initialize x_T^visual ← stochastic_encoding(x_0^visual, T);
Initialize t ← T;
for t = T to 1 do
    x̂_t ← CFG_NQG_with_swapping_self_attention(x_t, x_t^visual);  // (Fig 2) In denoising process, x_t
     and x_t^visual are swapped
    ;                                                                 // and predict ε̂_t
    if t_start ≤ t ≤ t_end then
        x_{t−1} ← color_calibration(x_t, x̂_t, x_0^visual) ;
    end
    else
        x_pred ← (x̂_t − √(1−α_t)·ε_θ(x̂_t)) / √α_t ;                 // predicted x_0
        x_dir ← √(1 − α_{t−1} − σ_t^2) · ε_θ(x̂_t) ;                  // direction pointing to x_t
        ε_t ~ N(0, I) ;                                               // Generate random noise
        x_noise ← σ_t · ε_t ;                                         // random noise
        x_{t−1} ← √α_{t−1} · x_pred + x_dir + x_noise ;               // updated x_{t−1}
    end
    Decrease t by 1;
end
return x_0
```

Algorithm 1: Visual style prompting with a real image as a reference

4. Style-Content prompt list

1. the great wave off kanagawa in style of Hokusai: (1) book (2) cup (3) tree
2. fire photography, realistic, black background: (1) a dragon (2) a ghost mask (3) a bird
3. A house in stickers style.: (1) A temple (2) A dog (3) A lion
4. The persistence of memory in style of Salvador Dali: (1) table (2) ball (3) flower
5. pop Art style of A compass . bright colors, bold outlines, popular culture themes, ironic or kitsch: (1) A violin (2) A palm tree (3) A koala
6. A compass rose in woodcut prints style.: (1) A cactus (2) A zebra (3) A blizzard
7. A laptop in post-modern art style.: (1) A man playing soccer (2) A woman playing tennis (3) A rolling chair
8. A horse in colorful chinese ink paintings style: (1) A dinosaur (2) A panda (3) A tiger
9. A piano in abstract impressionism style.: (1) A villa (2) A snowboard (3) A rubber duck
10. A teapot in mosaic art style.: (1) A kangaroo (2) A skyscraper (3) A lighthouse
11. A robot in digital glitch arts style.: (1) A cupcake (2) A woman playing basketball (3) A sunflower
12. A football helmet in street art graffiti style.: (1) A playmobil (2) A truck (3) A watch
13. Teapot in cartoon line drawings style.: (1) Dragon toy (2) Skateboard (3) Storm cloud
14. A flower in melting golden 3D renderings style and black background: (1) A piano (2) A butterfly (3) A guitar
15. Slices of watermelon and clouds in the background in 3D renderings style.: (1) A fox (2) A bowl with cornflakes (3) A model of a truck
16. pointillism style of A cat . composed entirely of small, distinct dots of color, vibrant, highly detailed: (1) A lighthouse (2) A hot air balloon (3) A cityscape
17. the garden of earthly delights in style of Hieronymus Bosch: (1) key (2) ball (3) chair
18. Photography of a Cloud in the sky, realistic: (1) a bird (2) a castle (3) a ship
19. A mushroom in glowing style.: (1) An Elf (2) A dragon (3) A dwarf
20. The scream in Edvard Munch style: (1) A rabbit (2) a horse (3) a giraffe
21. the girl with a pearl earring in style of Johannes Vermeer: (1) door (2) pen (3) boat
22. A wild flower in bokeh photography style.: (1) A ladybug (2) An igloo in antarctica (3) A person running
23. low-poly style of A car . low-poly game art, polygon mesh, jagged, blocky, wireframe edges, centered composition: (1) A tank (2) A sofa (3) A ship
24. Kite surfing in fluid arts style.: (1) A pizza (2) A child doing homework (3) A person doing yoga
25. anime artwork of cat . anime style, key visual, vibrant, studio anime, highly detailed: (1) A lion (2) A chimpanzee (3) A penguin
26. A cactus in mixed media arts style.: (1) A shopping cart (2) A child playing with cubes (3) A camera
27. the kiss in style of Gustav Klimt: (1) shoe (2) cup (3) hat
28. Horseshoe in vector illustrations style.: (1) Vintage typewriter (2) Snail (3) Tornado
29. play-doh style of A dog . sculpture, clay art, centered composition, Claymation: (1) a deer (2) a cat (3) an wolf
30. A cute puppet in neo-futurism style.: (1) A glass of beer (2) A violin (3) A child playing with a kite
31. the birth of venus in style of Sandro Botticelli: (1) lamp (2) spoon (3) flower
32. line art drawing of an owl . professional, sleek, modern, minimalist, graphic, line art, vector graphics: (1) a Cheetah (2) a moose (3) a whale
33. origami style of Microscope . paper art, pleated paper, folded, origami art, pleats, cut and fold, centered composition: (1) Giraffe (2) Laptop (3) Rainbow
34. A crystal vase in vintage still life photography style.: (1) A pocket watch (2) A compass (3) A leather-bound journal
35. The Starry Night, Van Gogh: (1) A fish (2) A cow (3) A pig
36. A village in line drawings style.: (1) A building (2) A child running in the park (3) A racing car
37. A diver in celestial artworks style.: (1) Bowl of fruits (2) An astronaut (3) A carousel
38. A frisbee in abstract cubism style.: (1) A monkey (2) A snake (3) Skates
39. Flowers in watercolor paintings style.: (1) Golden Gate bridge (2) A chair (3) Trees (4) An airplane
40. A horse in medieval fantasy illustrations style.: (1) A castle (2) A cow (3) An old phone

5. User study

For more rigorous evaluation, we conducted a user study with 62 participants. We configured a set with a reference image, a content text prompt, and six synthesized images with different initial noises per method from five competitors. The participants answered below question for 20 sets: Which method best reflects the style in the reference image AND the content in the text prompt? As indicated in Table A2, the majority of participants rated our method as the best. The lower ratings of the IP-Adapter in the user study may be attributed to its poor text alignment, as illustrated in Figure 9 despite its high style similarity. Examples of the user study are provided in Figure A2.

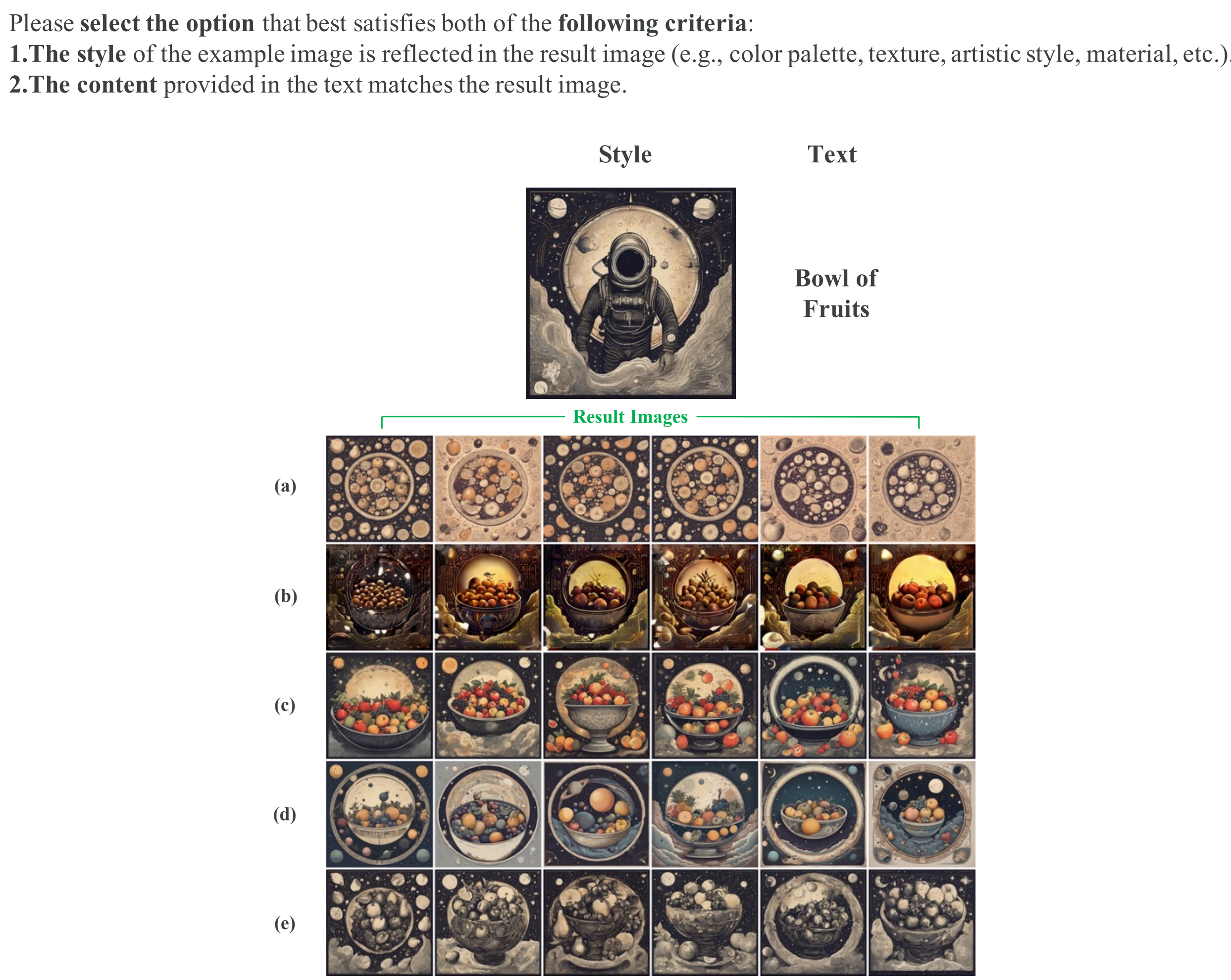


Figure A2. **Example of a user study.** Each row of images represents the result obtained by different method. The user had to assess which row is better in terms of style alignment and text alignment.

Table A2. **User study comparison.** We asked participants: Which method best reflects the style in the reference image AND the content in the text prompt?

Ours	StyleAligned	IP-Adapter	DB-LoRA	StyleDrop
58.15%	13.15%	18.47%	7.66%	2.58%

6. Additional figure results

Figure A3. Selective cross style-attention is important to avoid content leakage while preserving style similarity. Content leakage decreases diversity and text alignment.

Figure A4. Selective cross style-attention is important to avoid content leakage while preserving style similarity. Content leakage decrease diversity and text alignment.

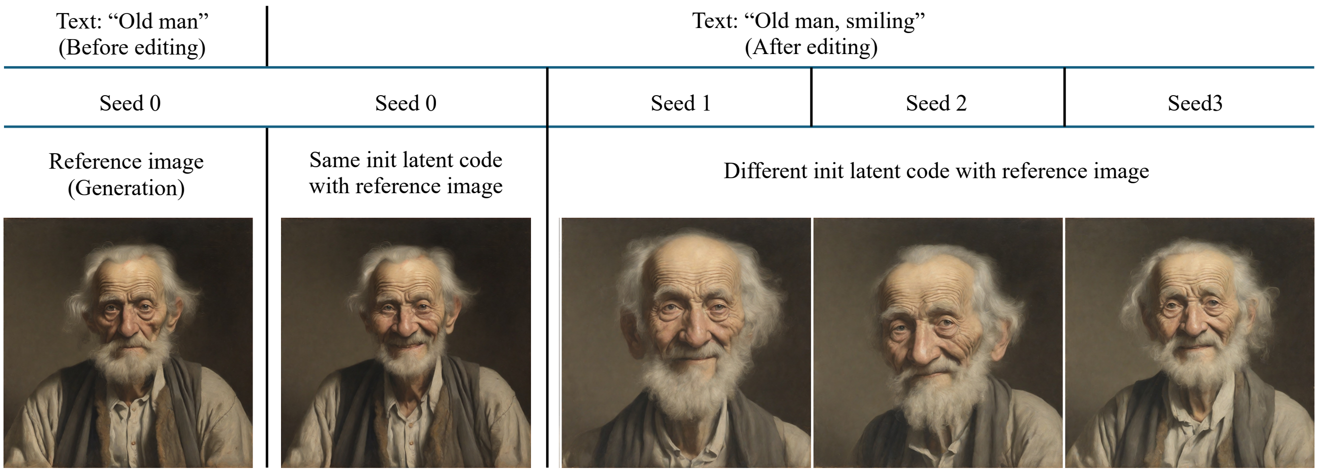


Figure A5. MasaCtrl [3] employ the same initial latent for the reference and inference denoising process for preservation of the target identity. Different initial latent infer the different identities. In contrast, due to the differences in the tasks themselves, ours do not share the same initial latent between the reference denoising process and the inference denoising process.

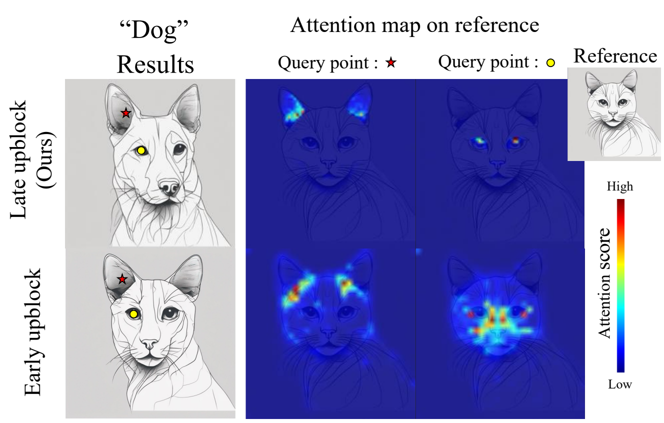


Figure A6. **Attention map visualization over late and early upblock layers.** The late upblock better focuses on the style-corresponding region than the early upblock, leading to more freedom to reassemble small parts. The early upblock attends a larger region leading to content leakage.

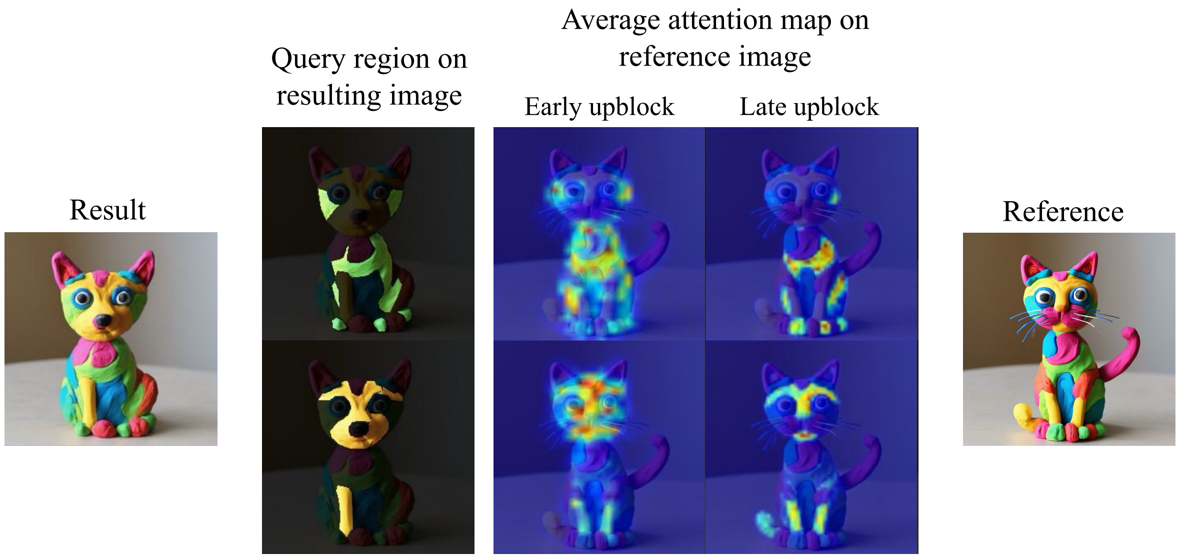


Figure A7. In Figure 6, we only show attention maps of 2 query points. Here, we provide the average attention map of multiple query points on the corresponding query region. At the late upblock, the query point region of the resulting image corresponds to the same style region of the reference image. On the other hand, at the early upblock, the query point region matches not only with the corresponding style region but also with the wider region

Figure A8. **Qualitative result of StyleKeeper on stable diffusion v1.5** Ours also works on the other pretrained diffusion models.

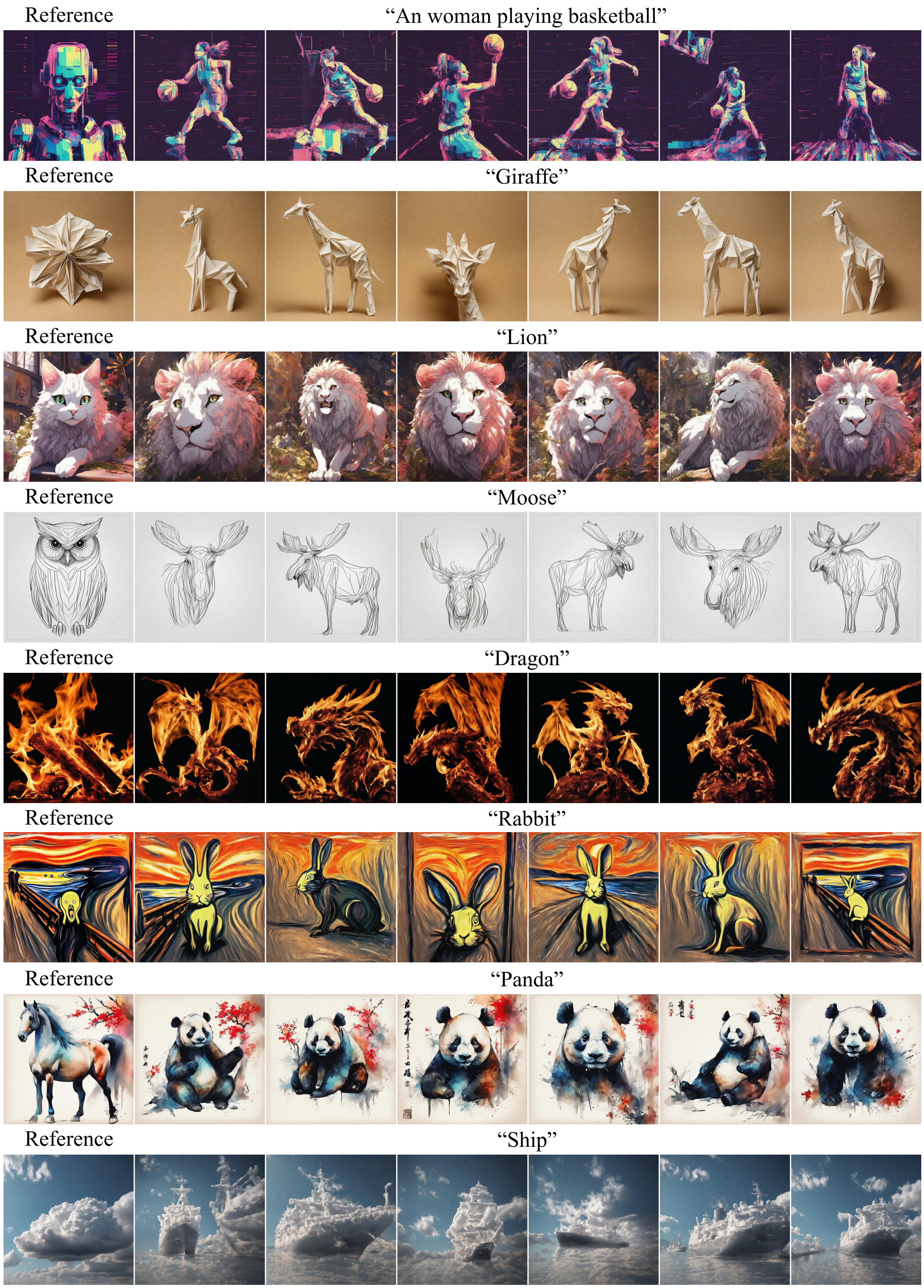


Figure A9. **Qualitative result of StyleKeeper within a prompt.** Ours can generate diverse layouts, poses and composition within a prompt.

Figure A10. Definition of style is different between ours and StyleAligned.

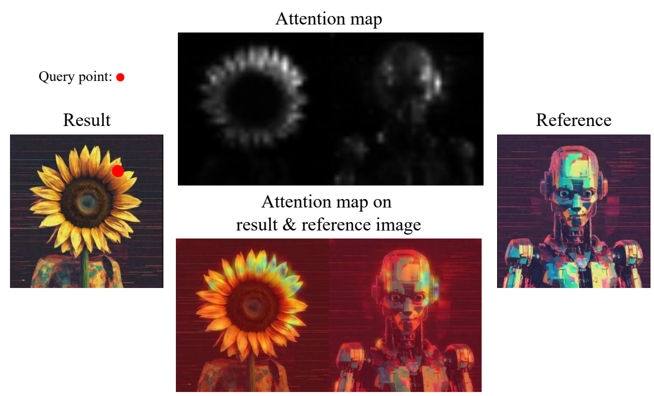


Figure A11. StyleAligned attends both on a reference and a resulting image for shared self-attention mechanism. In contrast, StyleKeeper only attends on a reference features which leads to better reflection of style in the reference image.

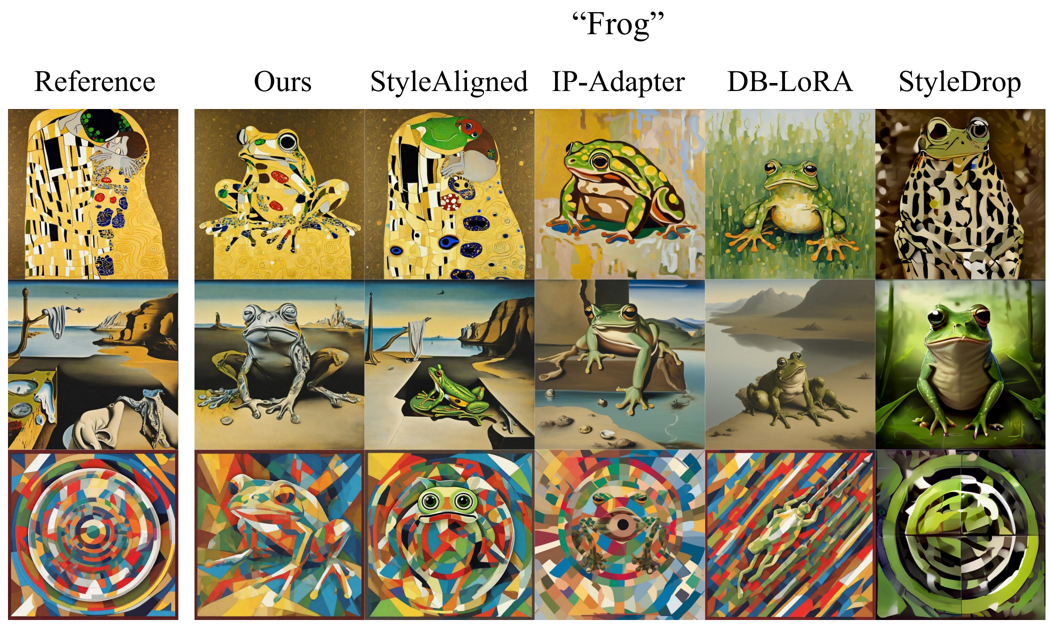


Figure A12. **Qualitative comparison with varying styles and fixed content.** StyleKeeper reflects style elements from various reference images to render "frog" while others struggle.

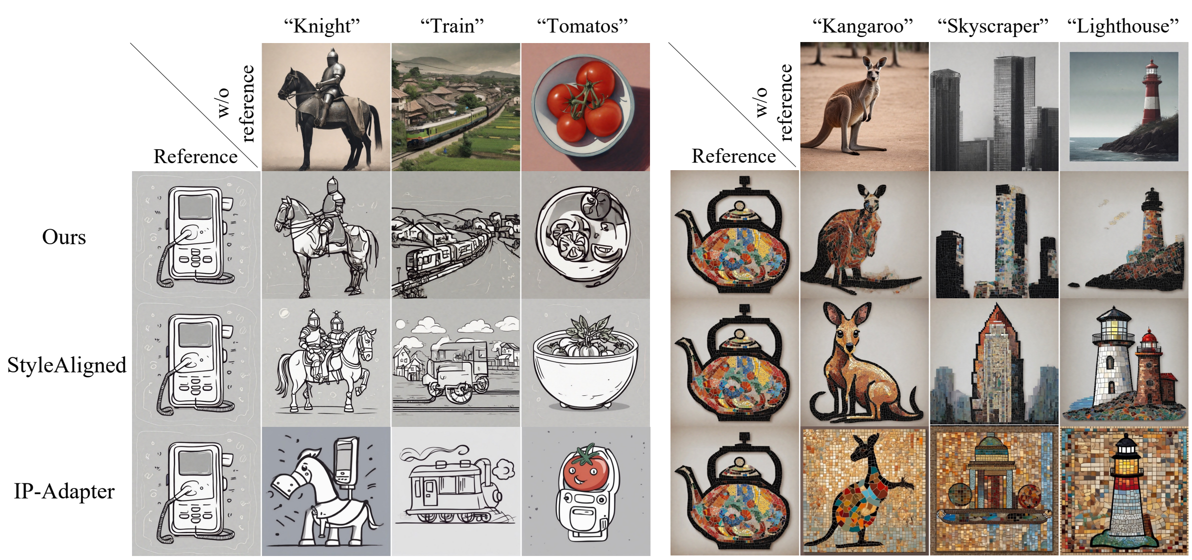


Figure A13. **Comparison of contents change while reflecting the style in the reference.** Each column shares the same initial noise. StyleKeeper reflects the style in the reference with minimal changes in the contents of the original denoising process while others produce drastic changes.

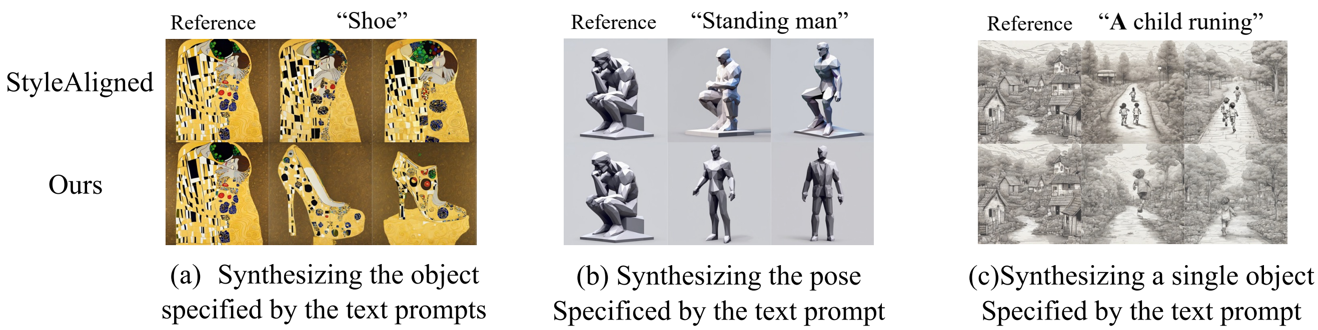


Figure A14. **Comparison for content leakage.** While StyleAligned suffers from content leakage from the reference, results from ours clearly align with the text prompts

Figure A15. More comparison of DDIM inversion with stochastic encoding, and with stochastic encoding and color calibration. Stochastic encoding reduces artifacts in the generated images, while color calibration better preserves the colors of the reference image.

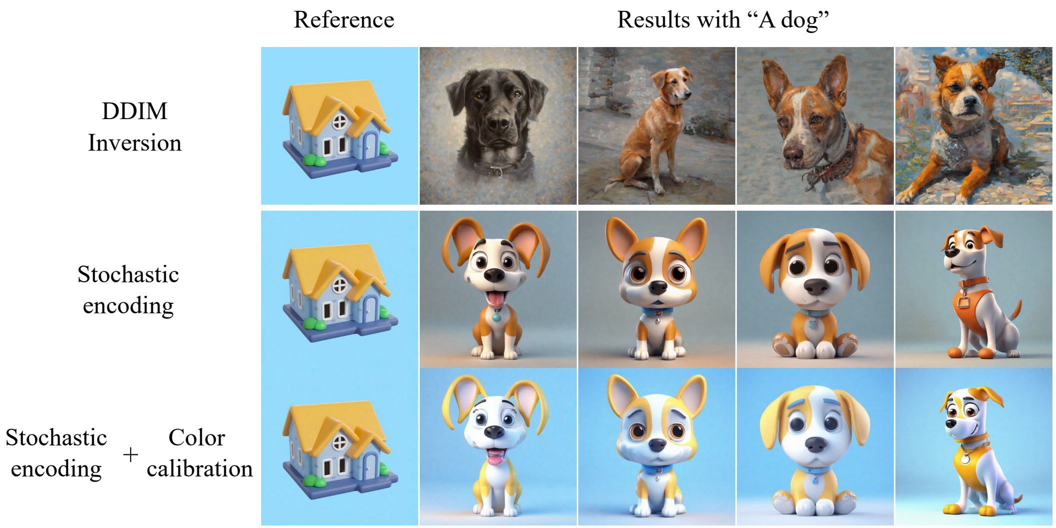


Figure A16. **Ablation study with StyleAligned [16].** Our strategy for a real reference image is compatible with self-attention variants for boosting the reflection of style elements.

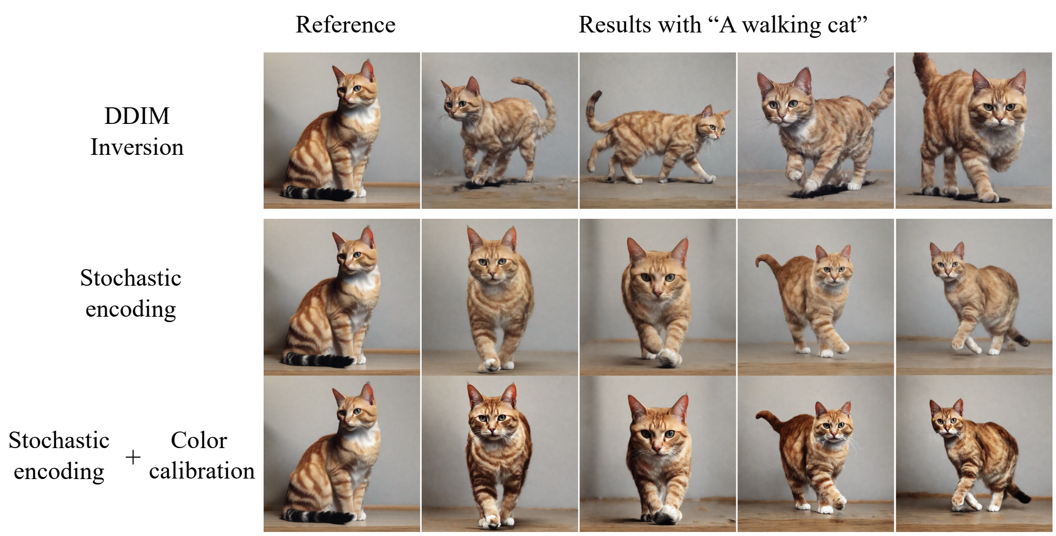


Figure A17. **Ablation study with MasaCtrl [3].** Our strategy for a real reference image is compatible with self-attention variants for reducing artifacts and boosting the reflection of visual elements.

Figure A18. **Color calibration ablation study with generating process from random noise.**. Color calibration also decreases the minor discrepancy of color between the reference image and the resulting image that barely occurs in the generation setting.

Figure A19. **Content leakage in DEADiff [35]** The results of DEADiff suffer from content leakage (e.g., the frontal face of "A moose" and the "An owl" reference image), which reduces diversity within a text prompt. StyleKeeper do not suffer content leakage while reflecting style.

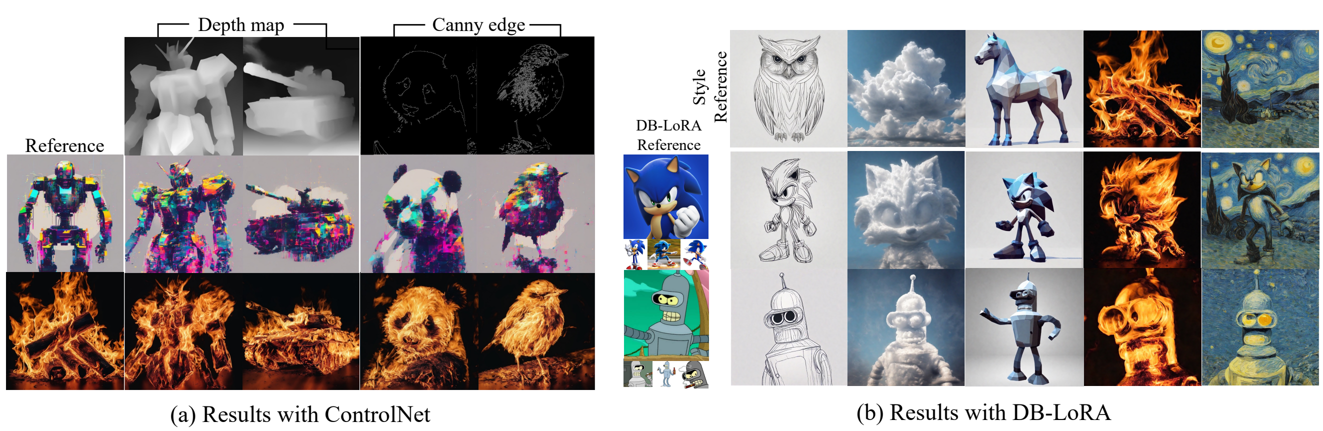


Figure A20. **Visual style prompting with existing techniques.** Our method is compatible with ControlNet and Dreambooth-LoRA.

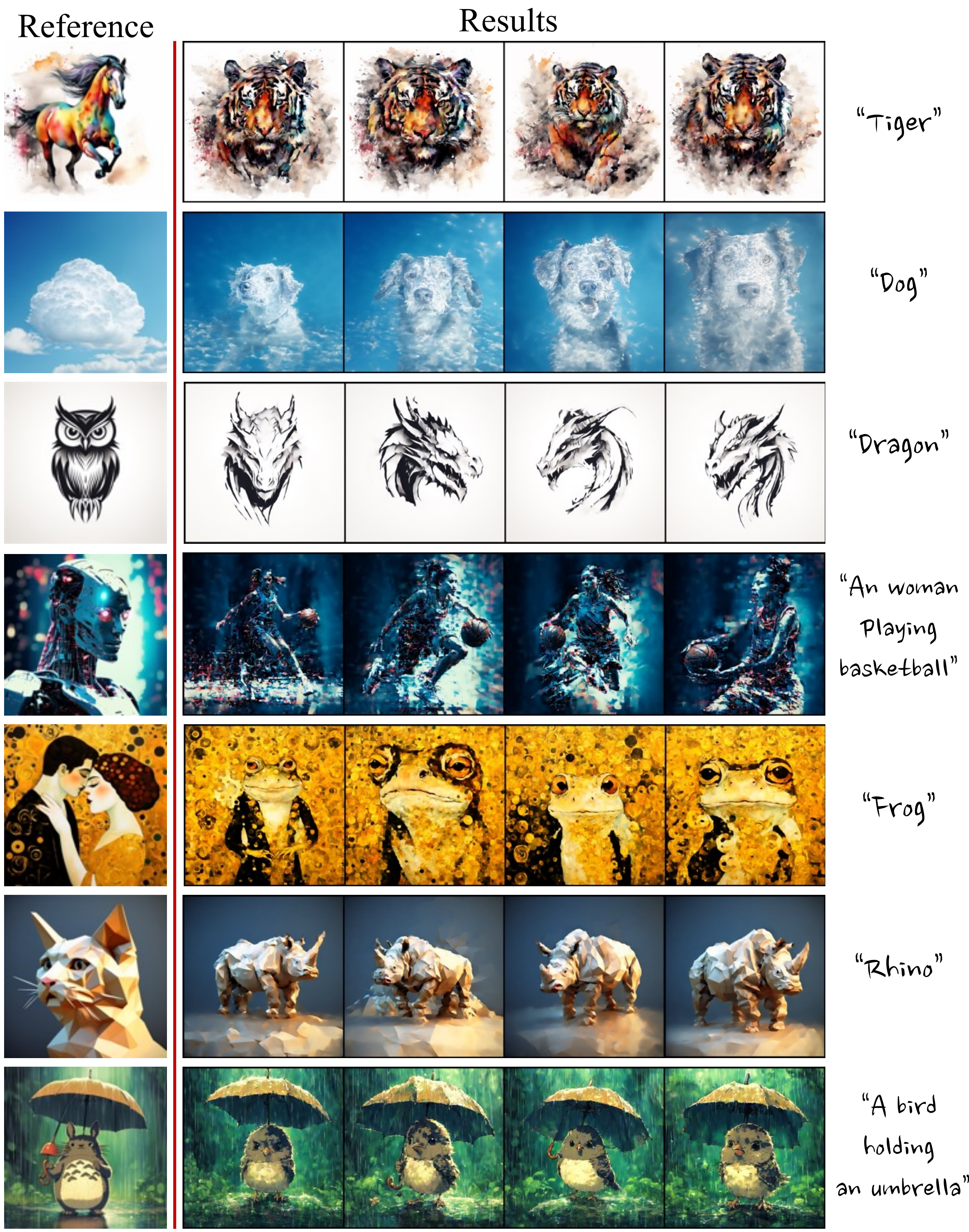


Figure A21. Qualitative result of visual style prompting in Pixart-α.

Figure A22. Qualitative result of visual style prompting in complex text prompts.

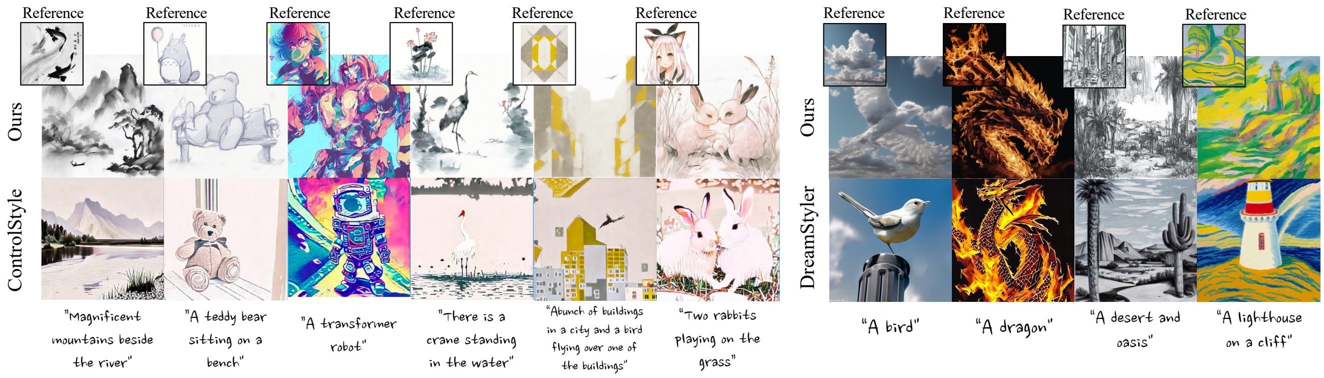


Figure A23. Comparison of ours with DreamStyler and ControlStyle.

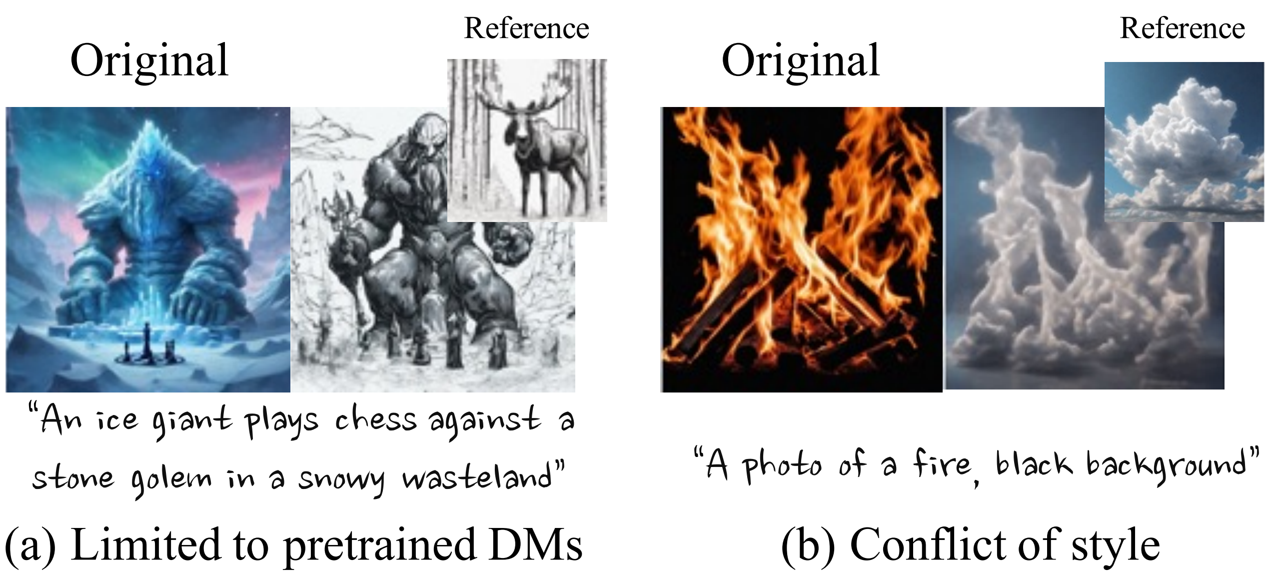


Figure A24. **Limitation: (1) Text Alignment Failure**: Missing 'stone golem' from the prompt (a), and **(2) Prompt Conflict**: 'Cloud in the sky' vs. 'fire, black background' (b).

StyleKeeper: Prevent Content Leakage using Negative Visual Query Guidance

Supplementary Material

7. Rationale

Having the supplementary compiled together with the main paper means that:

- The supplementary can back-reference sections of the main paper, for example, we can refer to **??**;
- The main paper can forward reference sub-sections within the supplementary explicitly (e.g. referring to a particular experiment);
- When submitted to arXiv, the supplementary will already included at the end of the paper.

To split the supplementary pages from the main paper, you can use Preview (on macOS), Adobe Acrobat (on all OSs), as well as command line tools.