

Tiled Diffusion — Supplemental Material

Or Madar
Reichman University
or.madar@post.runi.ac.il

Ohad Fried
Reichman University
ofried@runi.ac.il

A. Additional Evaluations

This section provides a more comprehensive visual comparison between our Tiled Diffusion method and existing approaches, namely Seamless tile inpainting (STI) and Asymmetric Tiling (AT).

A.1. Self-Tiling Comparison

In Figure 1, we present a side-by-side comparison of self-tiling results generated by Tiled Diffusion, STI, and AT. This comparison focuses on the method’s abilities to create seamlessly repeating patterns from a single image. Our Tiled Diffusion method demonstrates consistent performance in maintaining both local details and global structure across connection areas. STI, while capable of producing visually coherent results, sometimes struggles with maintaining logical continuity across the connection areas. AT, designed specifically for self-tiling, mostly performs well in creating seamless transitions.

A.2. One-to-One Tiling Comparison

Figure 2 showcases the performance of Tiled Diffusion and STI in one-to-one tiling scenarios. Our Tiled Diffusion method excels in creating logical and visually appealing connections between different images. STI, while effective in some cases, often struggles to maintain global coherence when connecting disparate images. The transition areas may appear blended but can lack the logical continuity that our method achieves.

A.3. Many-to-Many Tiling Examples

In Figure 3, we present a series of complex many-to-many tiling arrangements exclusively using our Tiled Diffusion method. These examples demonstrate the unique capability of our approach in handling intricate tiling scenarios that go beyond simple self-tiling or one-to-one connections.

B. Performance on Various Diffusion Models

To demonstrate the versatility and adaptability of our Tiled Diffusion method, we apply it to different state-of-the-art diffusion model architectures. Due to computational limitations, we focus on self-tiling scenarios along the X-axis for these experiments.

B.1. Stable Diffusion 3 and XL

We showcase the application of our Tiled Diffusion method to two advanced iterations of the Stable Diffusion model: Stable Diffusion 3 and Stable Diffusion XL. These models represent significant advancements in image generation capabilities, offering improved quality and more nuanced control over generated content. Figure 4 presents a side-by-side comparison of self-tiling results generated using Stable Diffusion 3 and Stable Diffusion XL, both enhanced with our Tiled Diffusion method. The examples demonstrate that our approach seamlessly integrates with these advanced models, preserving their unique strengths while adding tiling capabilities. These results underscore the adaptability of our Tiled Diffusion method, showing that it can effectively enhance the capabilities of cutting-edge diffusion models without compromising their inherent strengths.

B.2. ControlNet

We apply our Tiled Diffusion (TD) method to ControlNet, aiming to maintain tileability on given input tiled images after applying ControlNet’s controlled generation process. Figure 5 illustrates this application with five columns. The leftmost column shows the input tiled image, followed by the control image (canny edge detection or segmentation map) after applying the padding from TD. The middle column presents the result image tiled 1x3, demonstrating maintained tileability in the output when using TD. The fourth column shows the control image without padding, while the rightmost column displays the result image from ControlNet tiled 1x3, where TD was not used in the diffusion process. As evident from the last column, ControlNet alone does not guarantee that a tiled input will remain tileable after the diffusion process. However, by integrating our TD method into the diffusion process, we ensure that the output maintains its tileability while still adhering to ControlNet’s guidance.

C. Additional Method Evaluation

C.1. Texture Synthesis Evaluation

We evaluate patch-based texture synthesis in Table 1. Using the SeamlessGAN dataset, we compare our method with several specialized texture synthesis approaches: Neural Texture Synthesis with TexTile loss, SeamlessGAN, Content Aware Tile Generation [4] (using TexTile for 4-tile selection), SinFusion (with TexTile maximization per diffusion step), and our Tiled Diffusion method using SDXL with Seamlessly Tiling Existing Images application (SDXL + TD).

Our evaluation employs multiple metrics: Si-FID and CLIP-* from the respective papers, as well as TexTile (TT), Tiling Score (TS), and DISTS. Results in Table 1 demonstrate that our general framework performs similarly to or better than these texture-specific alternatives. This is particularly notable given that the competing methods are specifically designed for texture synthesis, while ours is a general tileable image generation framework.

The key distinction is that existing methods typically tile an existing texture patch, whereas our method can produce tileable textures directly from text prompts (as shown in Figure 8). We evaluate this text-to-texture capability in Table 2, comparing our approach against vanilla SDXL. Both TexTile and Tiling Score metrics confirm that our method successfully produces tileable results from text prompts alone.

C.2. Method Visual Ablation

To emphasize visually the necessity of both our constraints, we provide Figure 6, which demonstrates what happens when we remove either the Tiling Constraint or the Similarity Constraint. This visual ablation clearly illustrates that both constraints are essential components of our framework, each addressing different aspects of the tileable image generation problem.

Method	↓ DISTS	↓ Si-FID	↑ CLIP-Score	↑ CLIP-IQA	↑ TT	↓ TS
Neu. Tex. Syn.	0.481	6.5	0.26	0.58	0.684	0.06
SeamlessGAN	0.413	5.7	0.24	0.59	0.712	0.05
CATG (TexTile)	0.443	6.1	0.24	0.58	0.739	0.04
SF + TexTile	0.429	5.8	0.25	0.60	0.753	0.04
SDXL + TD	0.418	5.8	0.28	0.63	0.737	0.03

Table 1. Patch-based texture synthesis comparison

Data	↑ CLIP-Score	↑ TexTile (TT)	↓ Tiling Score (TS)
SDXL Textures	0.32	0.36	0.28
SDXL Textures + TD	0.32	0.74	0.03

Table 2. Prompt-based texture synthesis with and without Tiled Diffusion

D. Extended Application Examples

D.1. Seamlessly Tiling Existing Images

Our Tiled Diffusion method, when combined with Differential Diffusion [2], enables the transformation of non-tileable images into seamlessly tileable versions while preserving the majority of the original content. This application is particularly useful in various fields such as web design, game development, and digital art, where repeating patterns or seamless backgrounds are often required.

The process involves two main steps:

1. Applying Differential Diffusion to gradually modify the image edges, creating a smooth transition for tiling.
2. Using Tiled Diffusion to ensure the modified edges connect seamlessly while maintaining the overall image integrity.

This combined approach allows for minimal changes to the original image while achieving perfect tileability. The result is a versatile tool that can adapt existing images for use in contexts requiring seamless repetition. Figure 7 demonstrates several examples of our Seamlessly Tiling Existing Images application. Each row showcases the transformation from a non-tileable input to a seamlessly tileable output. The second and fourth columns highlight the difference in tileability between the original and processed images. These examples illustrate how our method effectively creates tileable images while preserving the majority of the original content.

D.2. Texture Synthesis

This application highlights our method’s capacity to generate high-quality, seamlessly tileable textures suitable for various fields. Figure 8 presents an array of textures generated using SDXL enhanced with our Tiled Diffusion method. These examples demonstrate self-tiling capabilities on both X and Y axes, producing textures that can be repeated indefinitely in any direction without visible seams or discontinuities. All the results are tiled [2x2] to show the tiling properties of each texture.

D.3. 360° Synthesis

Our Tiled Diffusion method extends to the generation of 360° panoramic images, demonstrating its versatility beyond traditional tiling applications. By implementing specific horizontal tiling constraints, our approach produces images with seamlessly connected edges, creating continuous, wraparound views. Figure 9 illustrates this capability. Each row presents a distinct 360° panorama example, with the first image showing the initial generated scene and the second displaying a horizontal translation, demonstrating the seamless wrap effect.

D.4. Pixel Art Remastering

We explored the use of Diffusion Models for img2img translation to remaster tilesheets from pixel art games. Pixel art games have tiling constraints inherent in game asset design, where certain tiles must seamlessly connect to others. Tiled Diffusion addresses that specific problem.

Our approach involves a two-step process:

1. Upscaling the original tilesheet using a depixelizing algorithm [1].
2. Applying img2img translation with Tiled Diffusion to generate new, high-resolution tiles while preserving the original tiling constraints. Figure 10 demonstrates this process using a tree asset from the game Alex The Alligator [5]. This example illustrates how our method can modernize classic pixel art assets while maintaining their essential tiling properties, potentially enabling game developers to create higher resolution versions of retro games that preserve the original design intentions.

E. Extended Discussion on Related Methods

E.1. TexTile

TexTile [3] introduces a differentiable metric for evaluating texture tileability, which can be used as a loss function in texture synthesis generative models. While innovative, TexTile has limitations in its applicability:

- It is primarily designed for self-tiling scenarios, limiting its use in complex multi-image tiling arrangements.
- The tiling score is trained specifically on texture images, potentially lacking accuracy in assessing tileability for more general images or other domains.
- It relies on a pre-trained model, which may introduce biases based on its training data.

Our analysis of TexTile’s performance on non-texture images revealed significant limitations. We selected the 1,000 VT2I images from the LAION 400M dataset we talked about in the paper and artificially created tileable versions by swapping the

left and right halves along the X-axis, and the top and bottom halves along the Y-axis. Despite this guaranteed tileability, TexTile’s evaluation yielded an average score of only 0.5. Notably, 40% of these tileable images received scores below 0.5, indicating they were incorrectly classified as non-tileable (accuracy = 60%). These results demonstrate TexTile’s reduced effectiveness when assessing tileability outside its intended texture domain, highlighting the need for more versatile tiling evaluation methods in broader image contexts. Figure 11 compares the performance of TexTile and our Tiling Score (TS) across different image types and tiling scenarios. The figure presents three columns: (1) vanilla text-to-image outputs (non-tiled), (2) swapped halves versions (tiled by definition), and (3) Tiled Diffusion outputs. For non-tiled images, both metrics consistently indicate non-tileability. However, for swapped images, which are inherently tileable, TS correctly shows low scores (indicating tileability), while TexTile often fails to recognize the tileability (requires a TexTile score > 0.5). For Tiled Diffusion outputs, TS consistently indicates tileability with low scores, whereas TexTile shows inconsistent results.

E.2. ControlMat

ControlMat [6] focuses on tileable material texture generation by incorporating noise rolling and inpainting techniques into the diffusion process. While effective for material textures, its scope is limited to self-tiling texture generation and specifically targets material capture. Adapting ControlMat for general image tiling would require significant modifications to the core diffusion process.

E.3. Content-aware Tile Generation

Content-aware Tile Generation [4] introduces a method for creating tileable textures using exterior boundary inpainting with diffusion models. The process starts with a text prompt and optionally an example image, or generates one using Stable Diffusion XL. For each tile, they select template patches from the example image, place these on the tile’s exterior as boundary conditions, then use Stable Diffusion 2 Inpainting to generate the tile’s interior. This approach allows them to create various tile types.

While innovative, this method differs significantly from our Tiled Diffusion approach. It employs a multi-stage process with separate inpainting steps, which may suffer from inconsistencies because inpainting is a post processing-step, which cannot influence the initial generation of the images to make them fit together semantically. In contrast, our approach works directly on the latent representations during image generation. By creating all the different tiles simultaneously, we can share the necessary tiling information between the images being generated. This process makes our results inherently tileable across various domains outside the scope of texture synthesis, reducing inconsistencies and ensuring seamless connections, and because of that, we don’t need any post-generation refinements.

F. Failure Cases

As mentioned in the main paper, our method occasionally encounters challenges with cross-axis connections. Figure 12 provides a specific example to illustrate this limitation. During the latent diffusion process, we enforce the same latent representation on the left and top sides of the 2nd image using the similarity constraint. These values are identical but rotated versions of each other. In the figure, the areas that should be similar are marked with blue (left side) and green (top side). Ideally, these marked areas should appear identical. However, as explained in the paper, rotation in the latent space does not guarantee an equivalent rotation in the pixel space after decoding the latent representation. This discrepancy is evident in the figure, where the blue and green marked areas exhibit noticeable differences.

G. Implementation Details

We list the key hyperparameters used in our experiments:

- Output image resolution: 512x512 pixels
- Context window size: 32
- Number of diffusion steps: 40
- Classifier-free guidance scale: 7.5
- Similarity constraint size: 5
- Random seed: 151
- Inpainting region width: 80 pixels

These settings were used consistently across all experiments unless otherwise specified. We use a single instance of the model and generate all images simultaneously, similar to batch generation. At each diffusion step K , the noise is generated for all images at that step.

References

- [1] Johannes Kopf and Dani Lischinski. Depixelizing pixel art. *ACM Trans. Graph.*, 30:99, 2011. [3](#)
- [2] Eran Levin and Ohad Fried. Differential diffusion: Giving each pixel its strength, 2023. [3](#)
- [3] Carlos Rodriguez-Pardo, Dan Casas, Elena Garces, and Jorge Lopez-Moreno. Textile: A differentiable metric for texture tileability. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. [3](#)
- [4] Sam Sartor and Pieter Peers. Content-aware tile generation using exterior boundary inpainting. In *ACM Transactions on Graphics*, 2024. [2](#), [4](#)
- [5] SourceForge. Alex the allegator, 2023. [3](#), [15](#)
- [6] Giuseppe Vecchio, Rosalie Martin, Arthur Roullier, Adrien Kaiser, Romain Rouffet, Valentin Deschaintre, and Tamy Boubekeur. Controlmat: A controlled generative approach to material capture, 2023. [4](#)

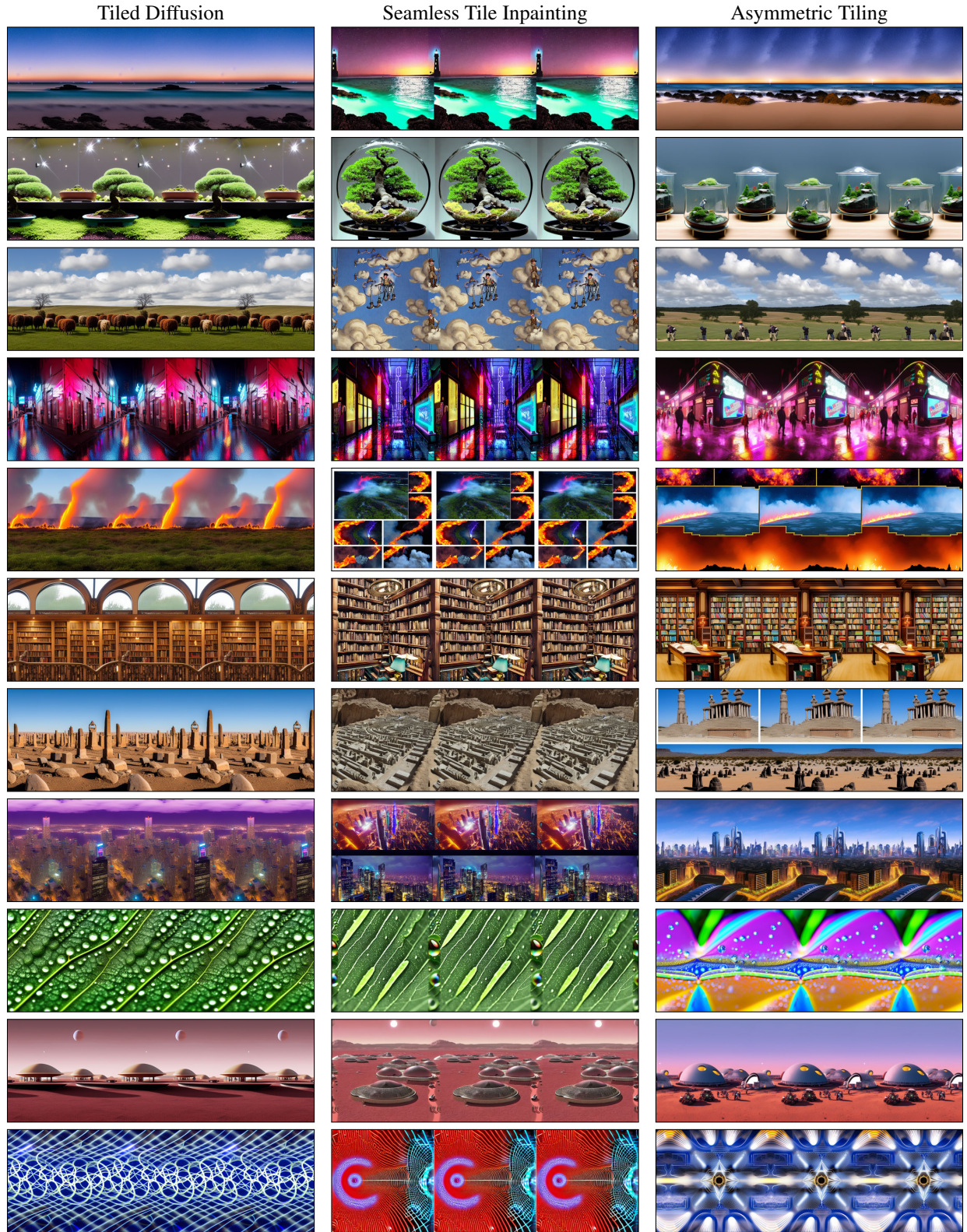


Figure 1. **Additional self-tiling qualitative comparison.** Comparison between Tiled Diffusion, Inpainting and Asymmetric Tiling methods. All images are tiled [1x3].

Tiled Diffusion



Seamless Tile Inpainting

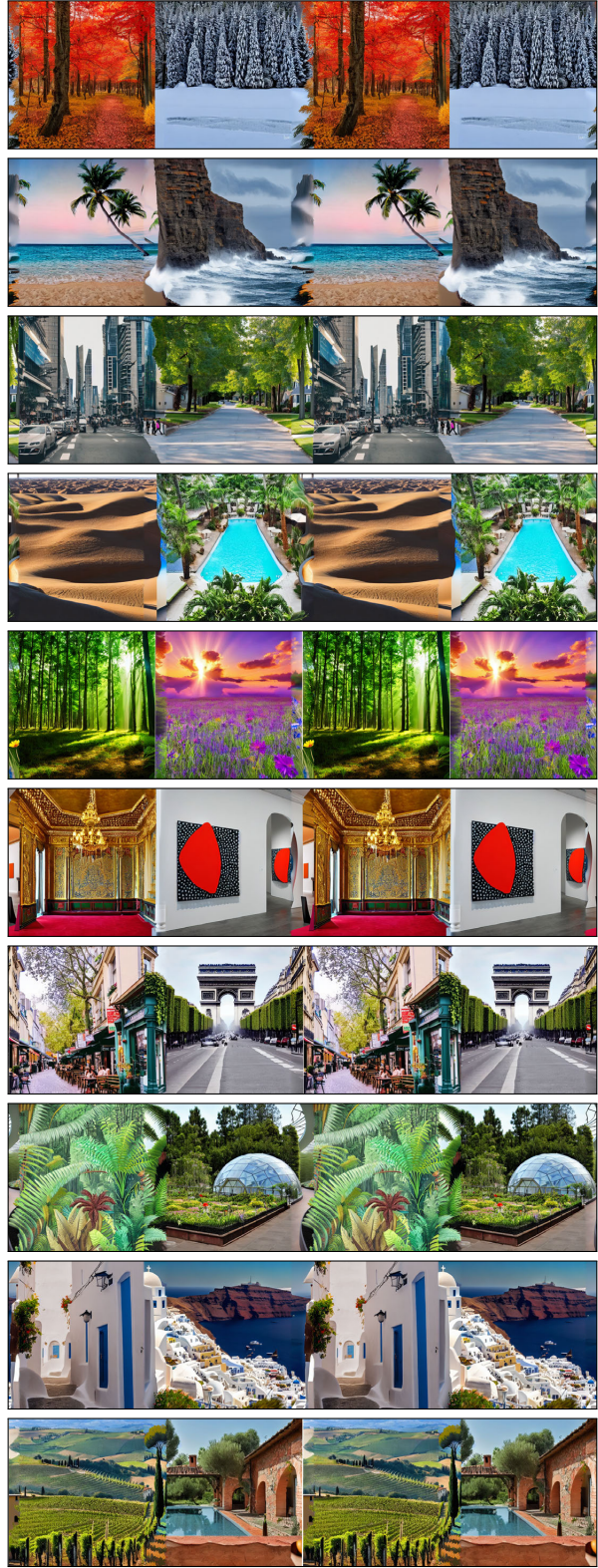


Figure 2. **Additional one-to-one qualitative comparison.** Comparison between Tiled Diffusion and Inpainting methods. All outputs show two distinct images tiled on the X-axis in a 1x4 arrangement.

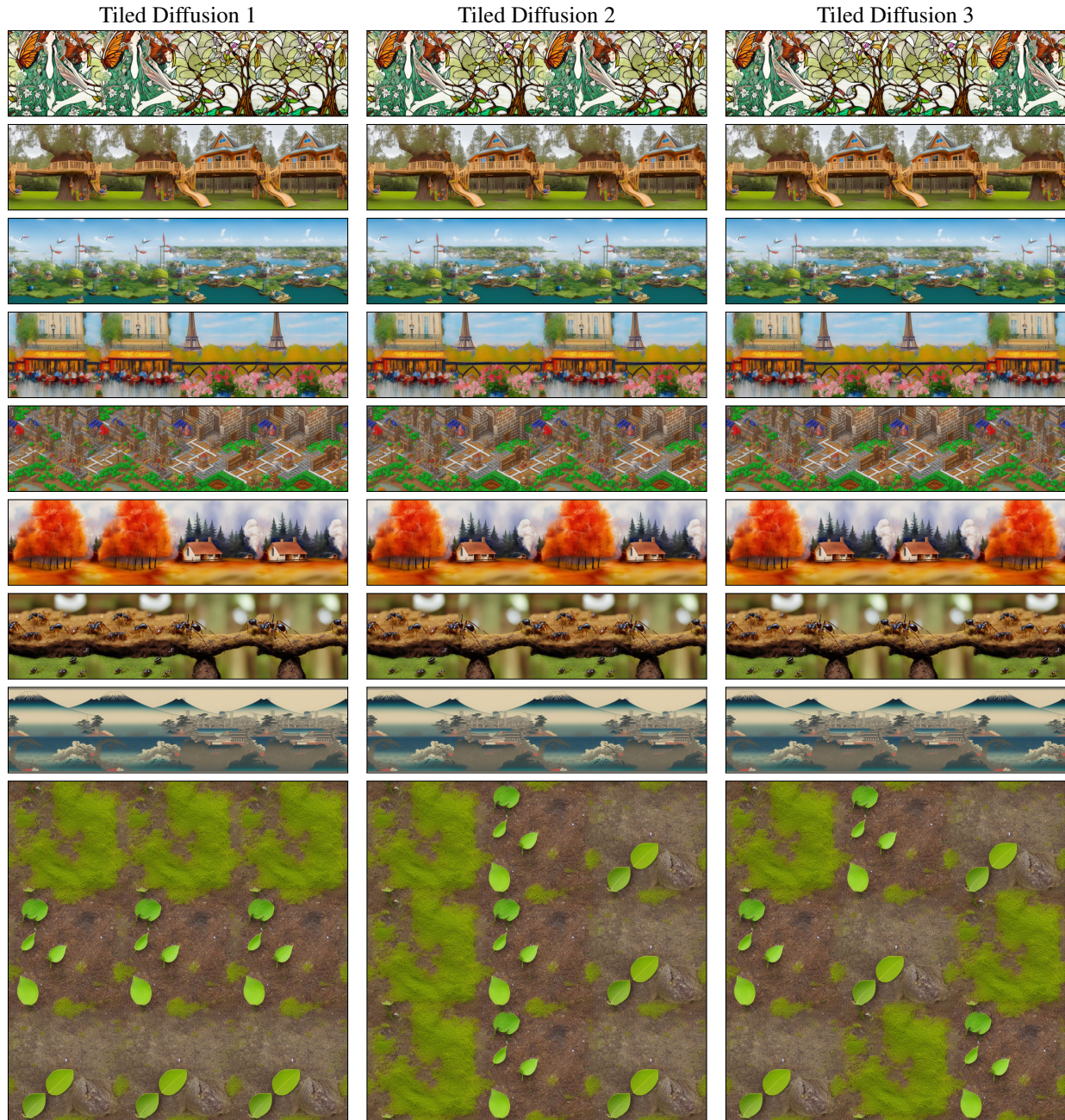


Figure 3. **Additional many-to-many qualitative demonstration.** Results from Tiled Diffusion, showing 1x4 tiling arrangements and one 3x3 arrangement in the last row.

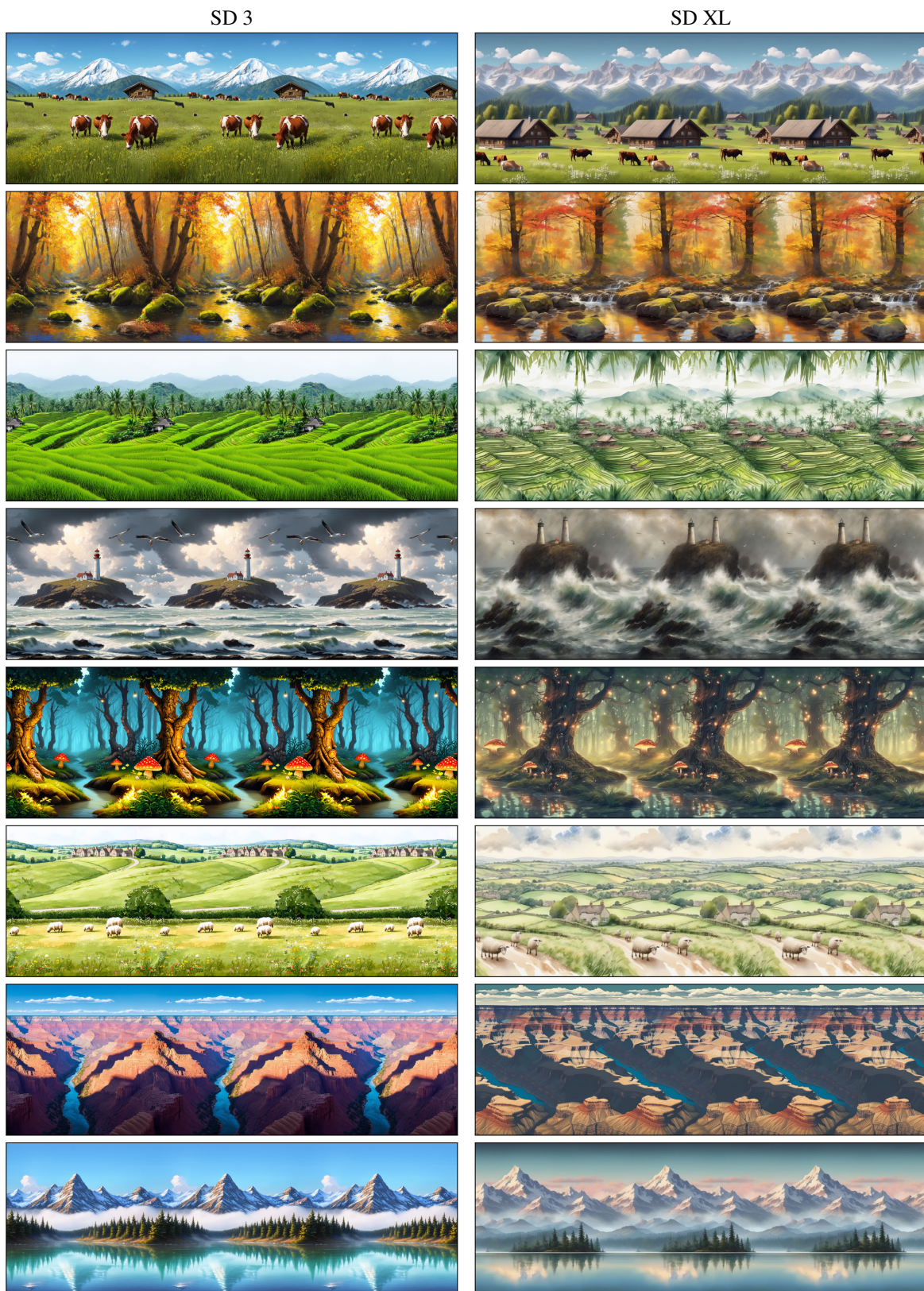


Figure 4. **Qualitative evaluation using different diffusion models.** Results from applying Tiled Diffusion to SD3 and SDXL models. All outputs are tiled 1x3.

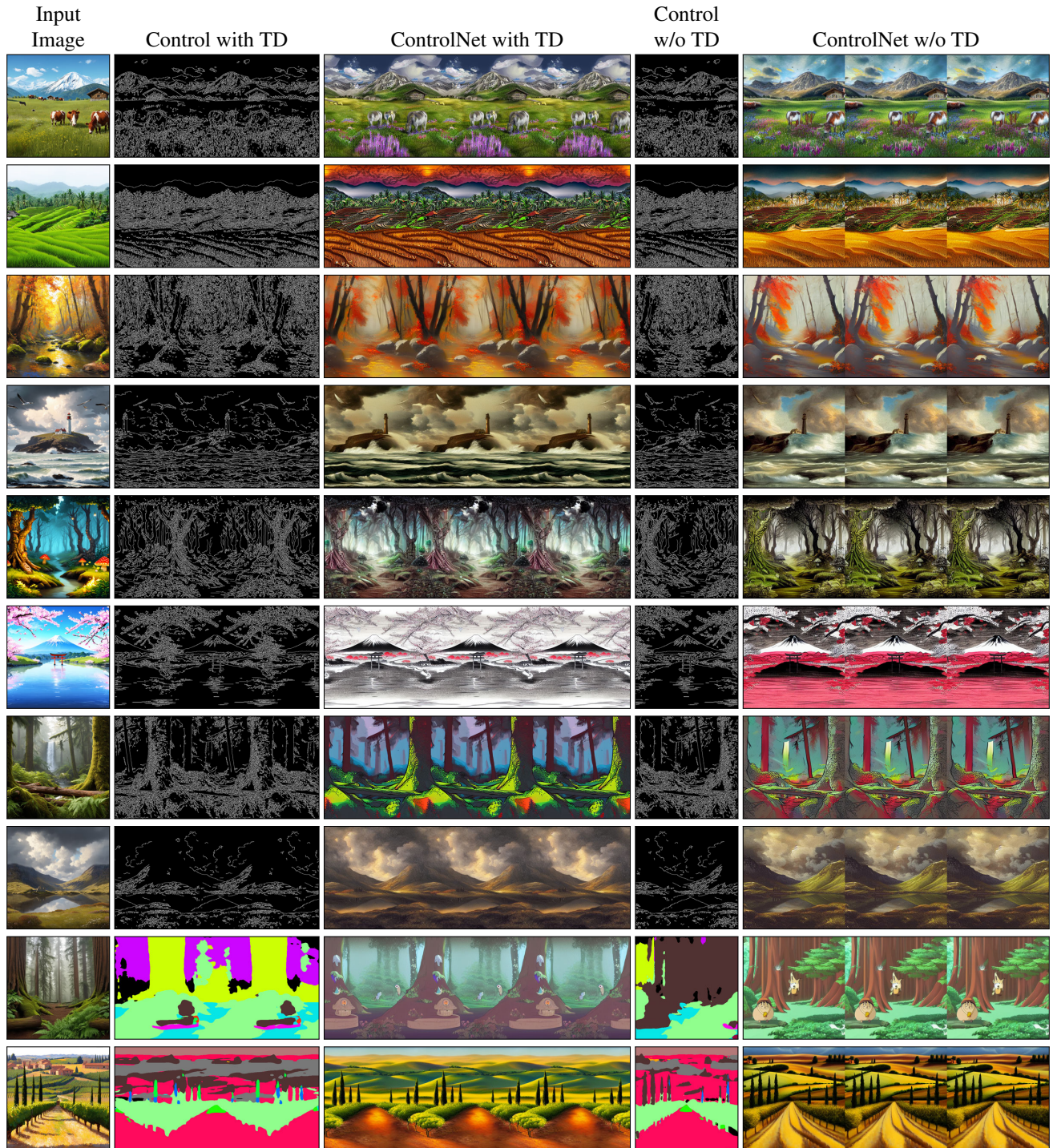


Figure 5. **ControlNet integration with Tiled Diffusion.** From left to right: input tiled image, control image (canny or segmentation) with TD padding, TD+ControlNet result tiled 1x3, control image without padding, and ControlNet-only result tiled 1x3. The comparison demonstrates the effectiveness of TD in maintaining tileability while preserving ControlNet’s capabilities.

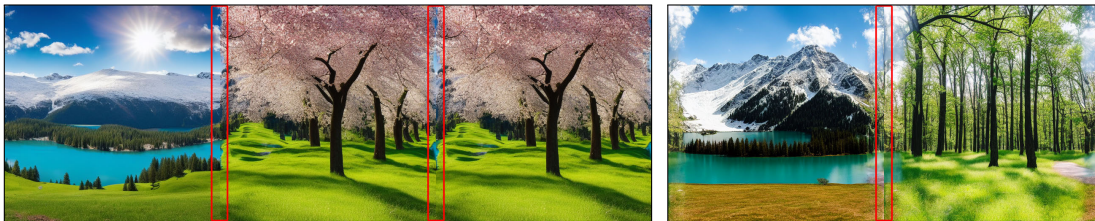


Figure 6. **Visual ablation of both constraints.** Many-to-many tiling w/o Similarity Constraint (left) and w/o Tiling Constraint (right). Red boxes highlight artifacts.



Figure 7. **Seamlessly Tiling Existing Images.** From left to right: input image, attempted 1x3 tiling of input, our method’s output, and successful 1x3 tiling of our output. Our approach makes minimal changes to create seamlessly tileable images.



Figure 8. **Seamless texture generation using SDXL.** Results from combining Stable Diffusion XL with our Tiled Diffusion method. Each texture is shown in a 2x2 tiled arrangement to demonstrate seamless tiling on both X and Y axes.



Figure 9. **360° panoramic image synthesis.** Each row shows a distinct panorama. Left: initial generated scene. Right: horizontally translated view demonstrating seamless wrapping when viewed as a continuous 360° panorama.

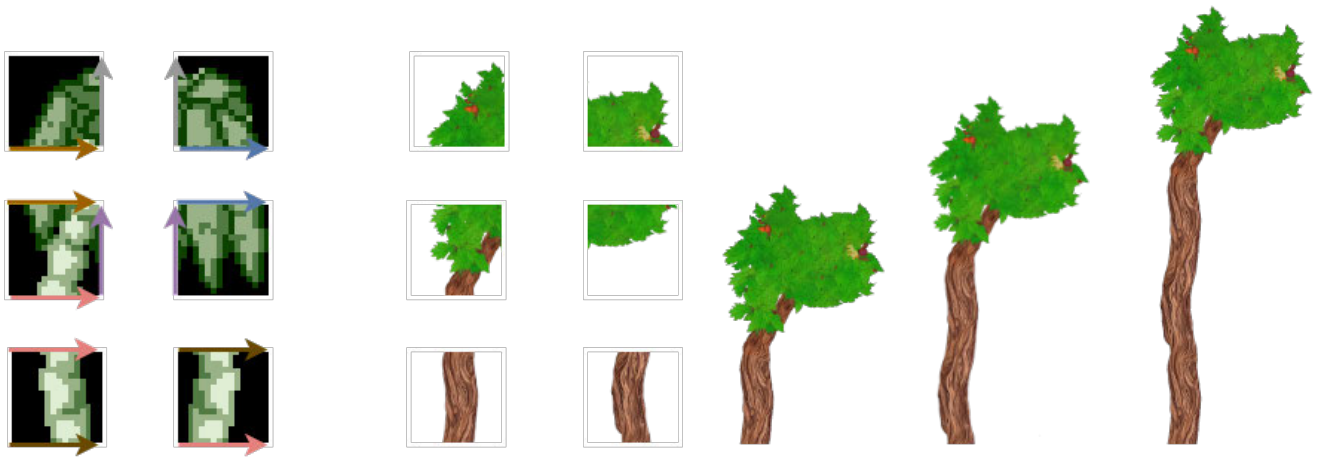


Figure 10. **Pixel Art Remastering.** Left: Original pixel art tree tiles and tiling constraints. Middle: Remastered tiles using our method. Right: Remastered tree assembled with different heights, demonstrating preserved tiling functionality. Asset from Alex The Alligator [5].

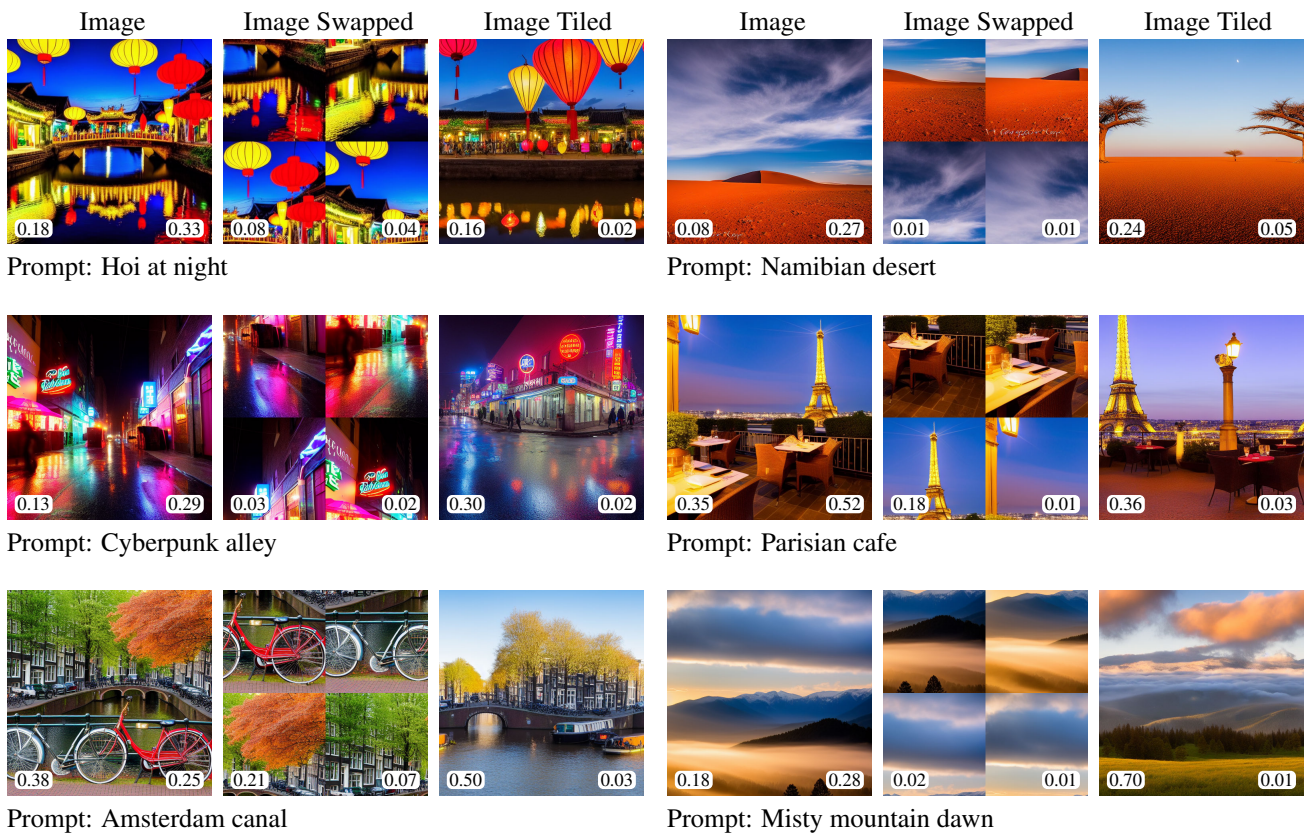


Figure 11. **Comparison of TexTile and Tiling Score performance.** From left to right: Vanilla text-to-image output (non-tiled), swapped halves version (tiled by definition), and Tiled Diffusion output. We attached a short prompt below each example. Scores shown: TexTile score (left) and our Tiling Score (right). Lower scores indicate better tileability for TS, and high (> 0.5) TexTile scores shows tileability. TS consistently reflects actual tileability across all image types, while TexTile struggles with swapped and Tiled Diffusion outputs.

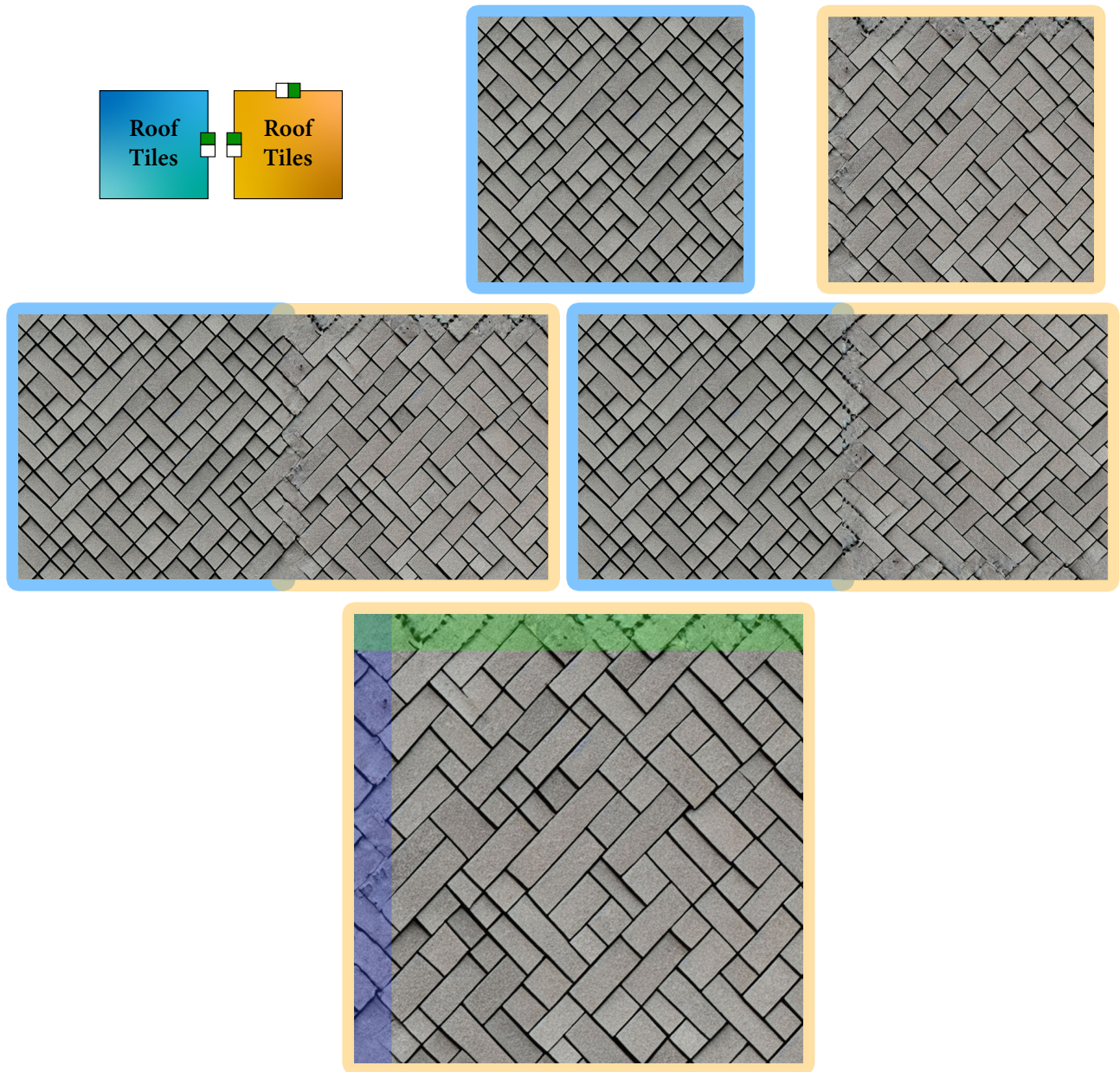


Figure 12. **Failure analysis.** Top row: Tiling constraints (left) and resulting images (right). Middle row: 1x2 tiling of original images (left) and rotated second image (right). Bottom row: Highlighted regions (blue and green) where similarity constraint should apply, showing discrepancies due to latent space rotation characteristics.