Tiled Diffusion

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

Ohad Fried
Reichman University
ofried@runi.ac.il

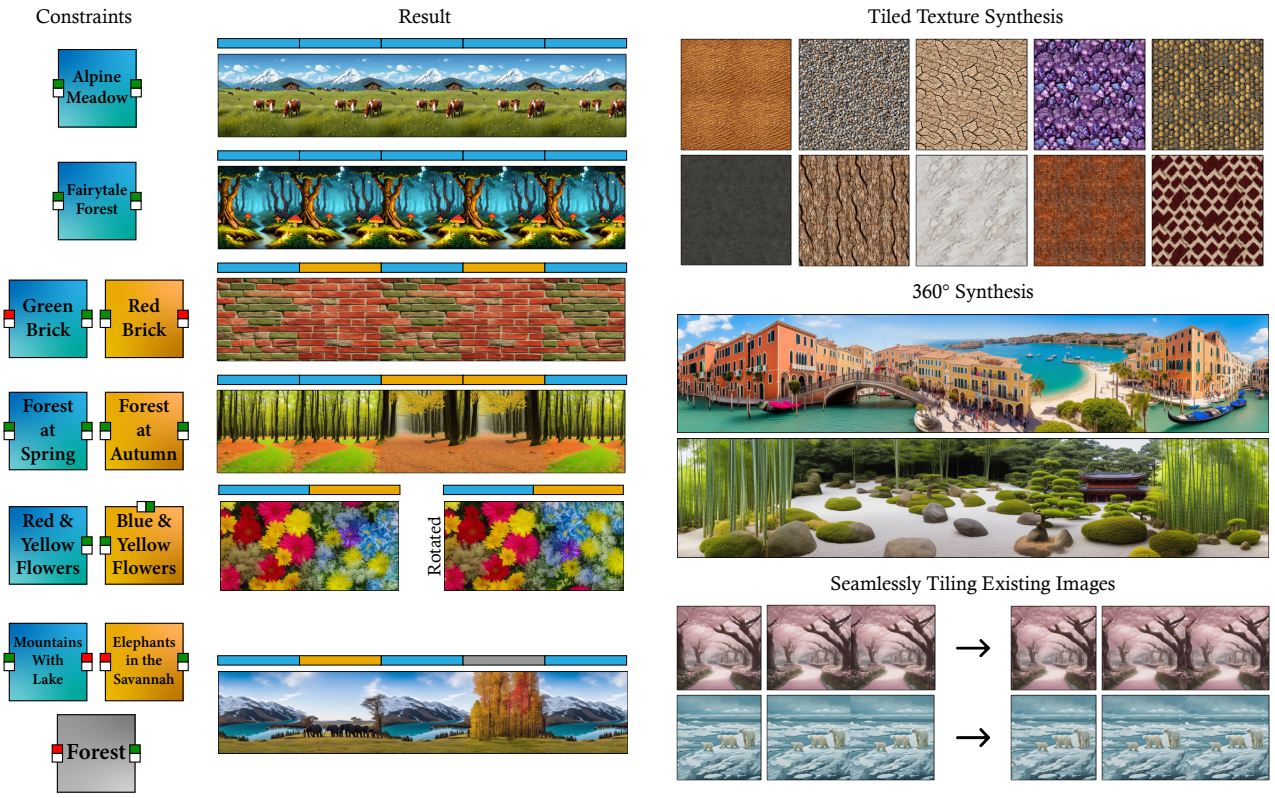

Figure 1. **Tiled Diffusion.** Our method generates seamlessly tileable images using diffusion models. Left: input constraints and their results. Matching color patterns on edges indicate they should tile seamlessly. (We use two colors to convey the direction.) Color strip above the result shows which constraint was used for each area. Right: various applications of our method: tiled texture synthesis, 360° synthesis, and seamlessly tiling existing images. Each texture includes 4 tiles (2x2); the 360° examples wrap on the horizontal axis; the seamlessly tiling example shows original images (left) and their tileable versions (right), both in 1x2 arrangements.

Abstract

Image tiling—the seamless connection of disparate images to create a coherent visual field—is crucial for applications such as texture creation, video game asset development, and digital art. Traditionally, tiles have been constructed manually, a method that poses significant limitations in scalability and flexibility. Recent research has attempted to automate this process using generative models. However, current approaches primarily focus on tiling textures and manipulating models for single-image generation, without inherently supporting the creation of multiple interconnected tiles across diverse domains. This paper presents Tiled Diffusion, a novel approach that extends the capabilities of diffusion models to accommodate the generation of cohesive tiling patterns across various domains of image synthesis that require tiling. Our method supports a wide range of tiling scenarios, from self-tiling to complex many-to-many

connections, enabling seamless integration of multiple images. Tiled Diffusion automates the tiling process, eliminating the need for manual intervention and enhancing creative possibilities in various applications, such as seamlessly tiling of existing images, tiled texture creation, and 360° synthesis.

1. Introduction

Tiling, the seamless connection of images to create coherent visual fields, is crucial in digital image processing and generation. It is essential for applications ranging from texture creation to video game asset development and digital art. Traditionally, manual tile creation has limited scalability and flexibility. Recent research [23, 32] has attempted to automate this process using generative models, but current approaches primarily focus on simple tiling scenarios, mainly single image generation in the field of texture synthesis. We define the tiling problem using a list of constraints. We consider each image I_i as a rectangle with four sides (right ($I_{i|}$), left ($I_{|i}$), top ($I_{\bar{i}}$), and bottom ($I_{\underline{i}}$)) and each constraint C_j is defined by two sets of sides. All the sides from one set can connect to all of the sides of the other set. These sets can include sides from any of the images being tiled. This flexible definition allows for various tiling scenarios. Examples of these scenarios are illustrated in Figure 2:

- Self-tiling: For single images that repeat seamlessly. As shown in Figure 2 (left), I_1 connects only to itself. Formally, we have two constraints: $C_1 = \{\{I_{1|}\}, \{I_{|1}\}\}$ and $C_2 = \{\{I_{\bar{1}}\}, \{I_{\underline{1}}\}\}$.
- One-to-one tiling: Allows each side of an image to connect to a maximum of one side of another image. Figure 2 (middle) demonstrates this for two images on the X-axis. Each constraint's sets contain exactly one element.
- Many-to-many tiling: Supports any number of connections per side, including complex cross-axis connections (X-axis to Y-axis). Figure 2 (right) shows a many-to-many scenario where right sides of both images can connect to left sides of both images. There are no limitations on the number of sides in either set.

Our method, Tiled Diffusion, extends the capabilities of diffusion models to accommodate these diverse tiling scenarios. It addresses several challenges, including ensuring content coherence across diverse tile boundaries, maintaining stylistic consistency between connected images, and preserving both local details and global structure in complex arrangements. We introduce two key innovations: a tiling constraint for global structure consistency and local detail matching, and a similarity constraint to eliminate artifacts in complex tiling scenarios. We demonstrate our method's effectiveness in several applications which are seamlessly tiling existing images, tileable texture generation and 360° synthesis.

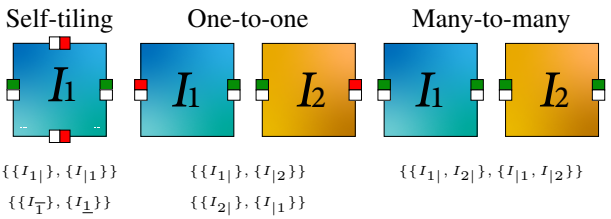

Figure 2. **Tiling scenarios.** Left: Self-tiling, where the image connects only to itself vertically and horizontally. Middle: One-to-one tiling on the X-axis, with each image connecting only to the other image. Right: Many-to-many tiling on the X-axis, where right sides of both images can connect to left sides of both images. Lower: constraints (C_j's) for each tiling scenario.

2. Related Work

We begin by discussing classical tiling techniques, highlighting the evolution of this field. We then explore recent advancements in image generation models, focusing on those applicable to image synthesis and editing. Next, we review tiling in popular applications. Finally, we examine current approaches to automated tiling using generative models, emphasizing the gaps our work aims to address.

2.1. Classical Tiling Techniques

Tiling, the seamless arrangement of patterns or images, has been a significant area of research in computer graphics and image processing. The introduction of Wang Tiles for image and texture generation allowed for non-periodic tiling of larger areas using squares with colored edges [6]. This concept was further expanded with an alternative approach using colored corners [15] and the development of recursive Wang Tiles for real-time blue noise generation [14]. A more theoretical approach demonstrated aperiodic tiling of the plane using a single prototile [29]. These classical approaches laid the foundation for efficient tiling techniques, addressing challenges in creating diverse, seamless patterns while balancing computational efficiency and visual quality.

2.2. Image Generation and Editing Models

Recent years have seen significant advancements in image generation and editing, driven by deep learning models. While Generative Adversarial Networks (GANs) [9] and Variational Autoencoders (VAEs) [13] have shown promise, diffusion models [30] have emerged as a particularly powerful technique for high-quality image generation. Diffusion models operate by gradually adding noise to an image and then learning to reverse this process [11]. This approach has led to improved sample quality and model robustness compared to earlier methods. Diffusion models have demon-

strated impressive results in various applications, including text-to-image synthesis [21], image super-resolution [26], and image editing [1, 2, 18, 33]. For a comprehensive overview of image synthesis and editing using diffusion models, we refer readers to Huang et al. [12] and Cao et al. [5]. We mostly use the Stable Diffusion 1.5 architecture [25], as the generative backbone of our tiling method, but we also use other latent diffusion architectures [7, 19, 34]. This choice allows us to leverage the high-quality image generation capabilities of diffusion models while extending their functionality to support complex tiling scenarios.

2.3. Related Applications of Tiling

2.3.1 Texture Synthesis

There is vast research in creating tileable textures. Most recent approaches rely on input patches [22–24, 27, 38–40], multi-stage pipelines [8, 27, 40], or specialized training [22, 23, 37, 38]. Tiled Diffusion is unique in generating natural tileable textures directly from prompts using any latent diffusion model, not constrained to self-tiling scenarios.

2.3.2 Infinite/Panorama Image Generation

Unlike recent approaches for large-scale image generation [3, 16, 35], Tiled Diffusion generates a fixed set of tiles that can be seamlessly arranged in any order and quantity, offering a time- and space-efficient solution for infinite or panoramic imagery.

2.4. Automated Tiling with Generative Models

Seamless Tile Inpainting (STI) [4] uses inpainting diffusion models to create seamless connections between swapped image quarters for self-tiling, or adjacent images for one-to-one connections. Asymmetric Tiling (AT) [31] modifies the Stable Diffusion architecture by replacing standard padding with circular padding, producing inherently tileable images but restricted to self-tiling scenarios with no rotations.

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, reducing inconsistencies and ensuring seamless connections.

We compare our method to STI and AT in Section 4 as they are the most closely related methods to our work. We demonstrate that our approach offers greater flexibility in handling diverse tiling constraints across multiple images. Importantly, our method is unique in its ability to support complex many-to-many tiling scenarios outside the scope of texture synthesis, offering a significant advancement in the field of automated tileable image generation.

3. Method

Our method generalizes the tiling problem to accommodate complex connections between multiple image sides. In this section, we first define the problem formally, then describe our approach to solving it using diffusion models.

3.1. Problem Definition

Given a set of desired output images $\mathcal{I} = \{I_1, I_2, ..., I_N\}$, where each image I_i has four sides: right ($I_{i|}$), left ($I_{|i}$), top ($I_{\bar{i}}$), and bottom ($I_{\underline{i}}$), we define a set of tiling constraints $\mathcal{C} = \{C_1, C_2, ..., C_M\}$. Each constraint C_j is represented as:

$$C_j = \{A_j, B_j\} \tag{1}$$

Where A_j and B_j are sets of sides. These sets define the tileability of the output images. Specifically, any side from set A_j is designed to seamlessly connect with any side from set B_j, and vice versa. Each side in A_j and B_j could be from any image in $\mathcal{I}$.

3.2. Tiled Diffusion

We use a diffusion model as the backbone of our method, operating on latent representations of images in a $H_{\text{latent}} \times W_{\text{latent}} \times D_{\text{latent}}$ latent space. The generation process begins with random Gaussian noise $\mathcal{N}(0, 1)$ and applies two key constraints at each diffusion step to ensure tiling coherence: a tiling constraint and a similarity constraint.

3.2.1 Tiling Constraint

The tiling constraint ensures global structure consistency across tiled images. This is achieved by copying a portion from a single latent representation in B_j to pad the latents in A_j, and vice versa (copying from A_j to B_j), for each constraint C_j. The way we choose the latent at each diffusion step is explained in Section 3.2.3. The padding mechanism essentially enlarges the original latent representation to a larger size, which is then cropped back in pixel space after decoding the latent at the end of the diffusion process. We use this approach because in the latent space of diffusion models, neighboring regions influence each other. By copying parts of the latent representation, we ensure that the generated images will have consistent content and style.

We define a context window of size w for this operation, where $0 \leq w \leq H_{\text{latent}}/2$ or $W_{\text{latent}}/2$, depends on the tiling direction. The choice of w affects the transition smoothness and variance between tiled images. Larger values of w produce smoother transitions by copying more context, ensuring better global consistency. Smaller values allow for sharper transitions and more variance, preserving local details but potentially at the cost of some global coherence. Figure 4 illustrates the effect of different context window sizes. This constraint is applied at each timestep of

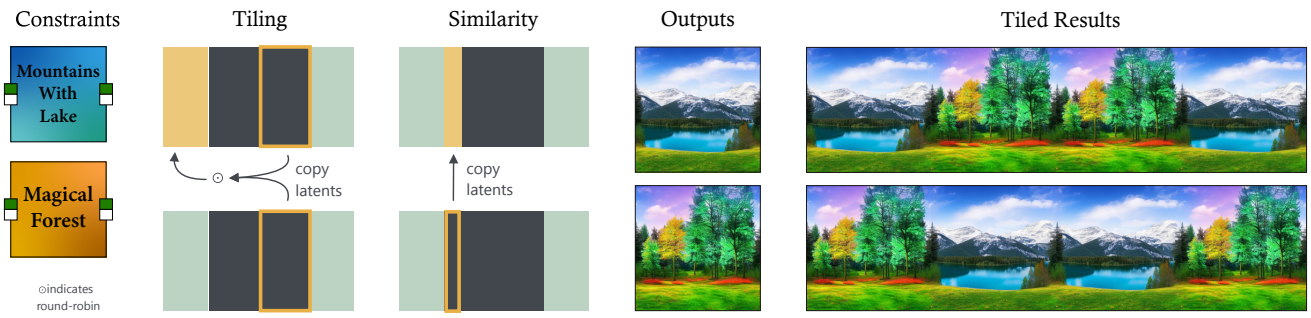

Figure 3. **Method overview and results.** Our method uses two key constraints in latent space: tiling constraints for global consistency and similarity constraints for seamless connections in complex scenarios. Left: Input constraints for many-to-many tiling on the X-axis between two images. Second column: Tiling constraint applied to the left side of the first image, using round-robin context selection. Third column: Similarity constraint propagated from the second image to the first. Fourth column: Output images after decoding and cropping. Right: Two example arrangements demonstrating many-to-many tiling scenarios.

the diffusion process, ensuring that the tiling information is consistently maintained throughout the generation. By applying the constraint at every step, we guide the diffusion process to generate images that inherently satisfy the tiling requirements, rather than trying to enforce tiling as a post-processing step.

3.2.2 Similarity Constraint

The similarity constraint ensures seamless connections in complex many-to-many tiling scenarios, when multiple sides are involved in a single constraint ($|A_j| > 1$ or $|B_j| > 1$). For each set of sides involved in such a constraint, we copy the same latent representation over a small window that starts from the side itself and extends inward. We use a fixed window width of 5 in the latent space, which our experiments have shown to work effectively across all scenarios.

A key difference between the tiling constraint and the similarity constraint is their impact on the final result. The tiling constraint affects the result indirectly by providing context in regions that will be later cropped. In contrast, the similarity constraint directly affects the result by ensuring similarity in regions that are kept after cropping. This difference allows the tiling constraint to use much larger windows compared to the similarity constraint.

3.2.3 Constraint Application Process

For constraints involving different orientations (e.g., a top side and a left side), we rotate the latents accordingly. For constraints involving multiple sides in A_j or in B_j, we employ a round-robin mechanism. At each diffusion step, we cycle through the available sides, ensuring that over the course of the diffusion process, each side is exposed to all of its potential matches. This approach allows the model to balance the influences of different constraints, resulting in coherently tiled outputs.

After applying all the constraints throughout the diffusion process, we decode the latent representations and crop the results to the original image dimensions, producing seamlessly tiled images that satisfy all predefined constraints. We provide visual ablation of both constraints in the supplementary material.

Figure 3 illustrates our Tiled Diffusion method with a many-to-many example, where the constraint $C_1 = \{\{I_{1|}, I_{2|}\}, \{I_{|1}, I_{|2}\}\}$ allows right sides of images 1 and 2 to connect seamlessly to left sides of both images. The figure shows input constraints, application of both constraint types, resulting images after decoding and cropping, and example tiling arrangements.

3.3. Image-to-Image (Img2Img) Generation

While the method described above focuses on text-to-image generation, our system also supports image-to-image (img2img) generation. In this setting, we start with an input image rather than random noise. We begin by encoding the input image $\mathbf{x}$ using the Variational AutoEncoder (VAE) to obtain an initial latent representation $\mathbf{z}_0$:

$$\mathbf{z}_0 = \text{Encoder}(\mathbf{x}) \tag{2}$$

To start the diffusion process from a noisy latent $\mathbf{z}_t$, we add noise to $\mathbf{z}_0$ according to the diffusion schedule:

$$\mathbf{z}_t = \sqrt{\bar{\alpha}_t}\mathbf{z}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim \mathcal{N}(0, \mathbf{I}) \tag{3}$$

where t is chosen to be sufficiently large, and $\bar{\alpha}_t$ is the cumulative product of noise schedule α_t. The diffusion process then proceeds from $\mathbf{z}_t$, applying the same tiling and similarity constraints as in the text-to-image case. For a more detailed explanation of img2img generation using diffusion models, we refer the reader to Rombach et al. [25].

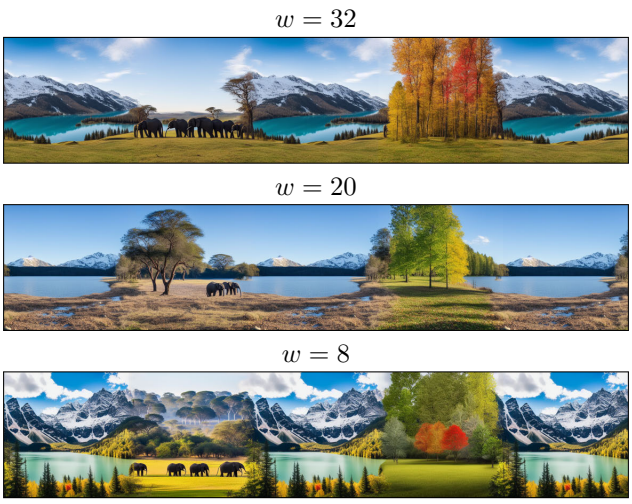

Figure 4. Illustration of the impact of different context window sizes (w) on tiling results. The figure displays panoramic views created by horizontally stacking the results for large (top), medium (middle), and small (bottom) w. We observe that with a large w, the transitions between tiles are smoother and more gradual, resulting in a more coherent overall image with less transition variations. As w decreases, we see sharper transitions variations, but potentially at the cost of global coherence.

In this paper, we primarily demonstrate our results on Stable Diffusion 1.5 and 2.0 [25]. However, we also showcase the applicability of our method to Stable Diffusion XL [19], Stable Diffusion 3 [7], and ControlNet [34] in the supplementary material.

4. Evaluation

Our evaluation includes qualitative and quantitative analyses compared to STI [4] and AT [31]. We visually compare image quality and tiling capabilities, showcasing complex tiling arrangements. The numerical metrics assess our method's performance in the different tiling scenarios, with comparisons to STI and AT where applicable.

4.1. Qualitative Evaluation

All methods support self-tiling. The one-to-one tiling is supported by our method and STI. The many-to-many tiling scenarios are exclusively supported by our method. Figure 6 compares self-tiling results of Tiled Diffusion, STI, and AT. In the landscape examples (rows 1-2), our method and AT maintain logical continuity across tiles, with coherent alignment of elements like trees and clouds. STI, however, shows misaligned elements and artifacts at connection points. For the texture example (row 3), all methods perform well, as it lacks complex high-level structure. Figure 7 demonstrates one-to-one tiling scenarios, comparing Tiled Diffusion and STI. These examples highlight our method's ability to generate coherent macro details between the two images. Our results show logically connected images, while the STI method struggles to maintain consistency across tile boundaries. This is because STI generates each image independantly, while our method shares information accross the images during synthesis. Figure 8 showcases many-to-many tiling scenarios, which are uniquely supported by our Tiled Diffusion method. These examples highlight our method's ability to handle complex tiling arrangements while maintaining visual coherence. Additional qualitative results and comparisons can be found in the supplementary document.

4.2. Quantitative Evaluation

To evaluate our method, we used a random sample of 1,000 captions from the LAION 400M dataset [28]. We assess the performance of our method across three scenarios: self-tiling, one-to-one tiling, and many-to-many tiling. Our evaluation aims to measure the quality of tiling, the fidelity to the input prompts, and the overall image quality. We employ four evaluation metrics to analyze these aspects:

Tiling Score (TS). We propose TS to measure tiling quality across various image domains. While existing methods for assessing tiling quality, like TexTile [23], focus on textures and are unable to generalize to other domains, our method is applicable to a broader range of images (examples in supplementary material). TS is calculated as the maximum of three separate measurements between two tiled images to capture potential discontinuities beyond the immediate connection area. Each measurement is defined as the mean absolute difference of pixel values at specific locations. Lower values indicate better tiling (range: 0.0 to 1.0). For a pair of images I_1 and I_2 tiled along an axis of length h:

$$TS_{pair}(I_1, I_2) = \frac{1}{h} \sum_{y=1}^{h} |I_1(x_1, y) - I_2(x_2, y)| \quad (4)$$

where x_1 and x_2 are the column indices of the adjacent edges. This equation is applied three times: once at the connection area, and twice at fixed offsets to the left and right (or top and bottom) of the connection. The overall TS is the average of these three measurements. This approach helps identify discontinuities that might occur at the boundaries of inpainted regions. To validate TS, we designed a procedure to test its effectiveness on well-connected images. We applied TS to 1,000 images generated from LAION captions using vanilla text-to-image generation (VT2I), as well as to versions of these images with their left and right halves, top and bottom halves swapped (swapped VT2I). The idea behind this procedure is that the swapped images should have perfect connections at the swapped edge, resulting in low

TS scores. As expected, the swapped VT2I images achieved significantly lower TS scores compared to the original VT2I images, confirming the validity of our metric.

Additional Metrics. To measure image-text similarity (CLIP-Score), we use the cosine similarity between the CLIP encodings [20] of the image and the text. For perceptual similarity, we use LPIPS [36], which assesses the diversity of generated images compared to the VT2I images, with higher values indicating more diverse outputs. For overall image quality and realism, we use FID [10] between the generated images and VT2I. We also include a baseline evaluation for VT2I and Swapped VT2I, where we additionally calculate the same metrics for both. This baseline allows us to assess how our method performs relative to standard text-to-image generation. Table 1 shows that our method performs well across all metrics compared to other approaches. For this analysis, we evaluated tileability on the X-axis using the sampled prompts. For self-tiling, we created 1,000 constraints (one per prompt). For one-to-one tiling, we randomly formed 500 pairs of prompts, resulting in 500 constraints. In many-to-many scenarios, each constraint C_j involved sets A_j and B_j with 2 sides each, using 4 prompts per constraint, totaling 250 constraints. Our Tiled Diffusion method performs consistently across all scenarios. The table also demonstrates the impact of removing key components of our method. Without the tiling constraint (TC), we achieve results similar to standard image generation — high quality but poor tiling. Without the similarity constraint (SC), performance remains stable except in many-to-many scenarios, where tiling quality decreases. This highlights the importance of both constraints in our method. Figure 5 illustrates how each metric (TS, CLIP-Score, LPIPS, FID) scales with increasing complexity in many-to-many tiling scenarios. Here, n represents the number of sides within sets A_j and B_j for each constraint C_j, with $n = 1$ corresponding to one-to-one tiling. As n increases, the results remain nearly constant, demonstrating that our method scales effectively to larger sets of constraints while maintaining tileability and quality.

Additionally, we provide texture synthesis analysis in the supplemental material where we compare our method to other methods in example-based texture synthesis, and evaluate our method in generating textures directly from prompts.

5. Applications

Our Tiled Diffusion method enables various applications in image generation and processing. We highlight three: Seamlessly Tiling Existing Images, Tileable Texture Generation, and 360° Synthesis, as illustrated in Figure 1. Additional examples and analysis are provided in the supplementary material.

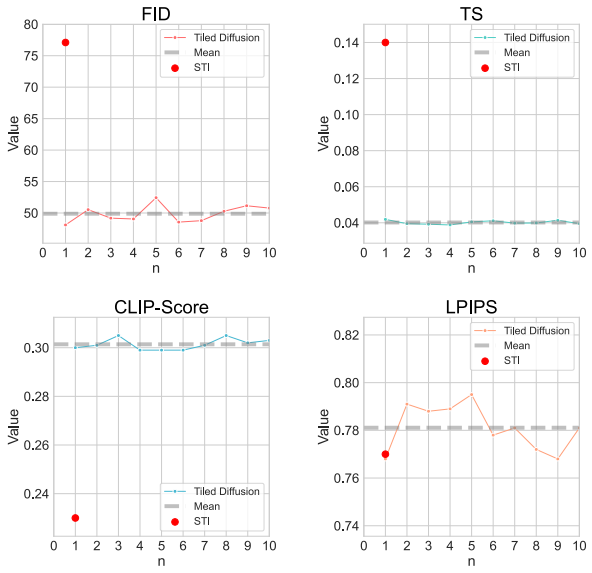

Figure 5. Quantitative analysis of our metrics as we increase the number of sides in A_j and B_j for each constraint C_j in many-to-many tiling scenarios ($n = |A_j| = |B_j|$). The graphs show relatively constant values across different side counts, demonstrating our method's ability to scale effectively. For $n = 1$ (one-to-one) we also include the STI method (red dot).

5.1. Seamlessly Tiling Existing Images

Many digital images are not inherently tileable, limiting their use in repetitive patterns. Our method, combined with Differential Diffusion [17], transforms non-tileable images into seamlessly tileable versions while preserving most of the original content.

5.2. Tileable Texture Generation

Our Tiled Diffusion method, applied to the SDXL model [19], generates seamlessly tileable textures. This enables creation of diverse, coherent texture sets and mix-and-match tiles (Figure 8 top row), enhancing realism in 3D modeling, game development, and digital art.

5.3. 360° Synthesis

Our Tiled Diffusion method creates wraparound panoramic images by generating seamlessly connected edges. This technique produces continuous views for wide-angle landscapes or architectural panoramas.

6. Limitations

Our method has a limitation with cross-axis connections (X-axis to Y-axis or vice versa), where small artifacts may appear in connecting regions. This stems from latent space rotations not perfectly aligning with pixel space rotations (after decoding), as the context copying during tiling constraint

Method	Self-Tiling				One-To-One				Many-To-Many			
	FID ↓	TS ↓	CLIP-Score ↑	LPIPS ↓	FID ↓	TS ↓	CLIP-Score ↑	LPIPS ↓	FID ↓	TS ↓	CLIP-Score ↑	LPIPS ↓
VT2I	✗	0.29	0.31	✗	✗	0.29	0.31	✗	✗	0.29	0.31	✗
Swapped VT2I	✗	0.03	0.14	✗	✗	0.03	0.14	✗	✗	0.03	0.14	✗
AT	49.2	0.03	0.29	0.79	✗	✗	✗	✗	✗	✗	✗	✗
STI	59.2	0.03	**0.31**	0.77	77.1	0.14	0.23	0.77	✗	✗	✗	✗
Tiled Diffusion	**47.9**	0.03	0.30	**0.76**	**48.1**	**0.04**	**0.30**	**0.76**	**50.5**	**0.03**	**0.30**	**0.79**
Tiled Diffusion w/o TC	48.1	0.28	0.30	0.77	48.4	0.31	0.30	0.77	50.9	0.30	0.30	0.79
Tiled Diffusion w/o SC	47.9	0.03	0.30	0.76	48.1	0.04	0.30	0.76	50.5	0.12	0.30	0.79

Table 1. **Quantitative Analysis.** Comparison across self-tiling, one-to-one, and many-to-many scenarios. Top: pseudo ground-truth . VT2I (1000 Stable Diffusion-generated images) and Swapped VT2I (with halves swapped on X and Y axes) are non-tiled and tiled-by-definition images, respectively. Middle: comparisons. Our Tiled Diffusion method outperforms AT and STI in most comparable scenarios (self-tiling and one-to-one), and uniquely supports many-to-many tiling. Bottom: Ablations. Without the tiling constraint (TC), images are not tiled (high TS, similar to VT2I); without the similarity constraint (SC), tiling performance decreases in many-to-many scenarios (higher TS).

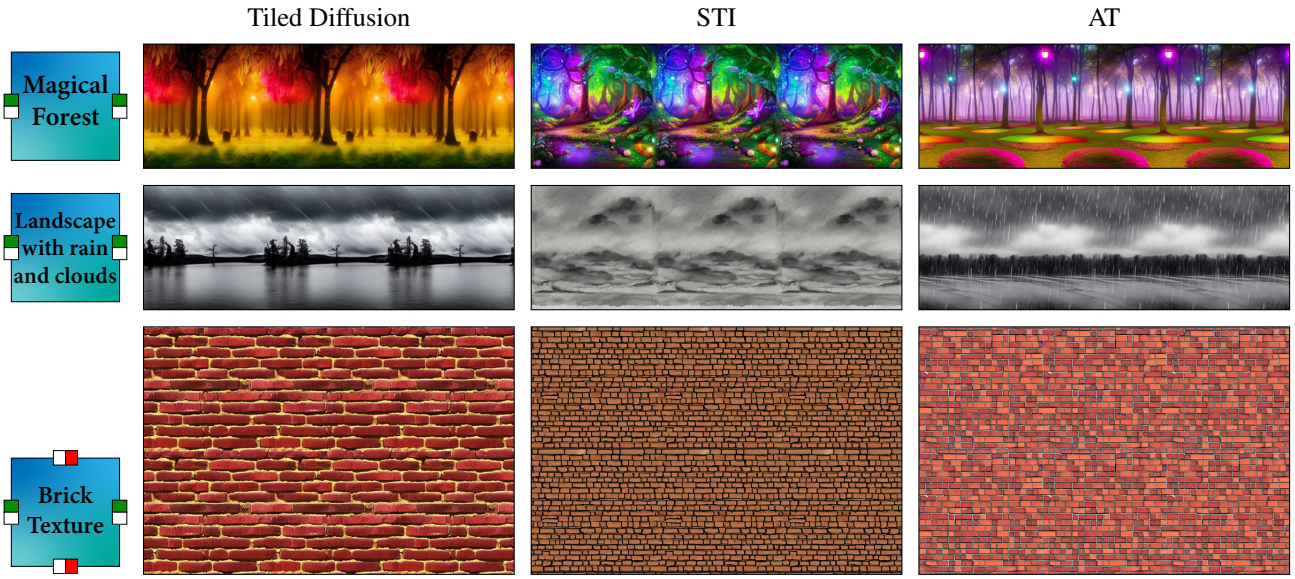

Figure 6. **Qualitative comparison of self-tiling.** The leftmost column shows the tiling constraints, followed by results from Tiled Diffusion, STI, and AT. The first two rows demonstrate 1x3 horizontal tiling of landscapes, while the last row shows 2x3 tiling of a texture. In the landscape examples (rows 1-2), our method and AT maintain logical continuity, while STI shows misalignments and artifacts at connection region. For the texture example (row 3), all methods perform comparably well due to the lack of complex high-level structure.

application requires rotation for cross-axis alignment which may not perfectly translate to pixel space. While these artifacts are typically minor and often imperceptible, future work could explore more sophisticated rotation techniques or post-processing steps to refine cross-axis connections. A detailed overview is provided in the supplementary material.

7. Conclusion

We introduced Tiled Diffusion, a novel approach that extends diffusion models to accommodate complex tiling scenarios - self-tiling, one-to-one, and many-to-many. Our key innovations include a flexible tiling constraint and a similarity constraint, that ensure global structure consistency and eliminate artifacts in complex tiling scenarios. We also introduced a tiling score to measure tiling quality. Our evaluations demonstrate the effectiveness of our approach, with superior performance across various metrics and diverse scenarios. We explored compelling applications in seamlessly tiling existing images, tileable texture generation, and 360° synthesis, showcasing the potential impact on texture creation, digital art, and other fields.

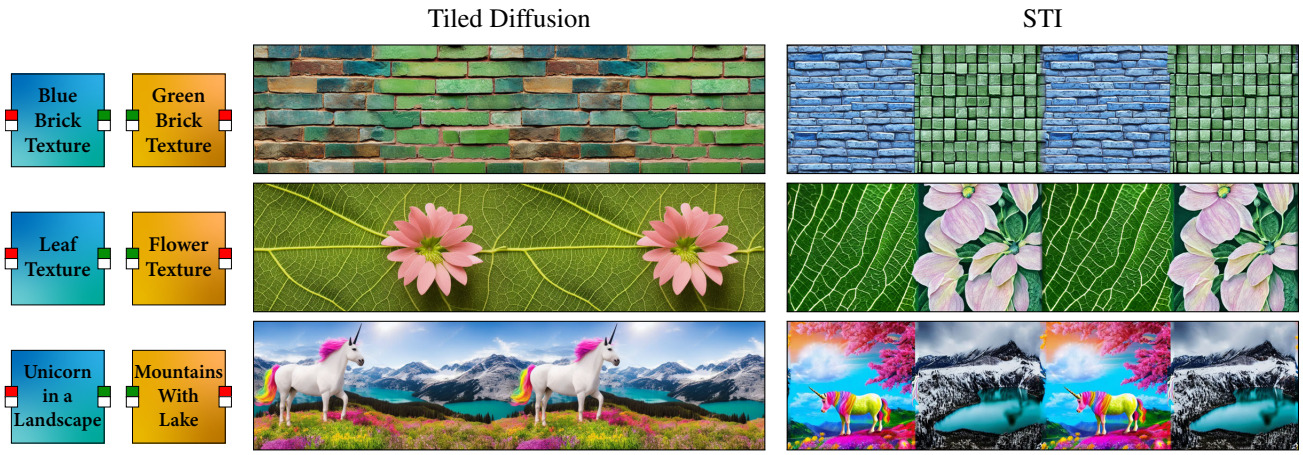

Figure 7. **Qualitative comparison of one-to-one tiling.** The leftmost column displays the tiling constraints, followed by results from Tiled Diffusion and STI, respectively. All outputs are tiled 1x4. Our Tiled Diffusion method demonstrates superior coherence and seamless transitions between tiles. In contrast, the STI method, despite using a relatively large 80-pixel wide connection region between 512x512 pixel images, shows noticeable discontinuities and struggles to maintain consistent content across tile borders.

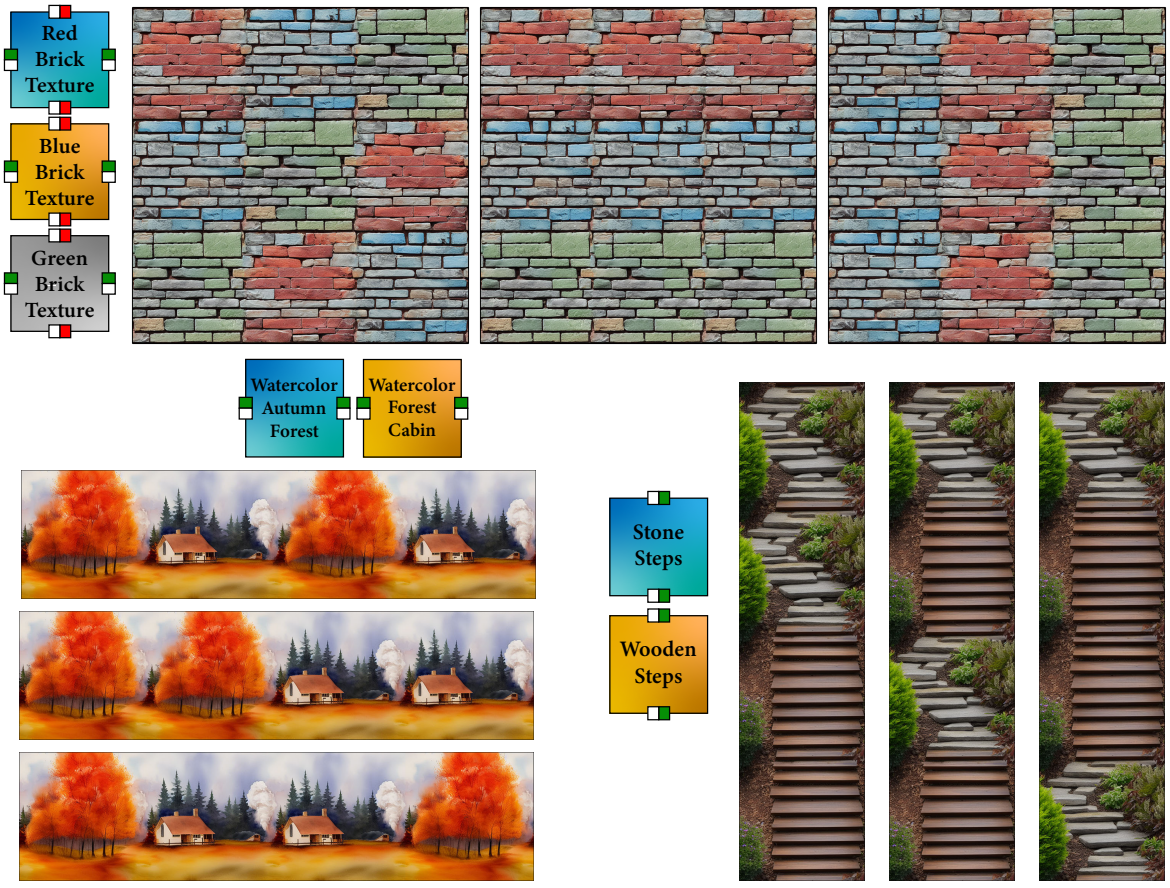

Figure 8. **Qualitative demonstration of Many-to-Many tiling.** Many-to-many connections enable several connection patterns between the resulting images. Top: 3x3 tiling of bricks. Bottom left: 1x4 horizontal tiling of watercolor forest paintings. Bottom right: 4x1 vertical tiling of different steps.

8. Acknowledgments

This work was supported in part by the Israel Science Foundation (grant No. 1574/21).

References

[1] Omri Avrahami, Dani Lischinski, and Ohad Fried. Blended diffusion for text-driven editing of natural images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 18208–18218, 2022. 3

[2] Omri Avrahami, Ohad Fried, and Dani Lischinski. Blended latent diffusion. *ACM Trans. Graph.*, 42(4), 2023. 3

[3] Omer Bar-Tal, Lior Yariv, Yaron Lipman, and Tali Dekel. Multidiffusion: Fusing diffusion paths for controlled image generation, 2023. 3

[4] brick2face. Seamless tile inpainting. https://github.com/brick2face/seamless-tile-inpainting, 2023. 3, 5

[5] Yihan Cao, Siyu Li, Yixin Liu, Zhiling Yan, Yutong Dai, Philip S. Yu, and Lichao Sun. A comprehensive survey of ai-generated content (aigc): A history of generative ai from gan to chatgpt, 2023. 3

[6] Michael Cohen, Jonathan Shade, Stefan Hiller, and Oliver Deussen. Wang tiles for image and texture generation. *ACM Transactions on Graphics*, 22, 2003. 2

[7] Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, Dustin Podell, Tim Dockhorn, Zion English, Kyle Lacey, Alex Goodwin, Yannik Marek, and Robin Rombach. Scaling rectified flow transformers for high-resolution image synthesis, 2024. 3, 5

[8] Anna Frühstück, Ibraheem Alhashim, and Peter Wonka. Tilegan: synthesis of large-scale non-homogeneous textures. *ACM Transactions on Graphics*, 38(4):1–11, 2019. 3

[9] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks, 2014. 2

[10] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium, 2018. 6

[11] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models, 2020. 2

[12] Yi Huang, Jiancheng Huang, Yifan Liu, Mingfu Yan, Jiaxi Lv, Jianzhuang Liu, Wei Xiong, He Zhang, Shifeng Chen, and Liangliang Cao. Diffusion model-based image editing: A survey, 2024. 3

[13] Diederik P Kingma and Max Welling. Auto-encoding variational bayes, 2022. 2

[14] Johannes Kopf, Daniel Cohen-Or, Oliver Deussen, and Dani Lischinski. Recursive wang tiles for real-time blue noise. *ACM SIGGRAPH 2006 Papers*, 2006. 2

[15] Ares Lagae and Philip Dutré. An alternative for wang tiles: colored edges versus colored corners. *ACM Trans. Graph.*, 25:1442–1459, 2006. 2

[16] Yuseung Lee, Kunho Kim, Hyunjin Kim, and Minhyuk Sung. Syncdiffusion: Coherent montage via synchronized joint diffusions. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. 3

[17] Eran Levin and Ohad Fried. Differential diffusion: Giving each pixel its strength, 2023. 6

[18] 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, 2022. 3

[19] 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, 2023. 3, 5, 6

[20] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision, 2021. 6

[21] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents, 2022. 3

[22] Carlos Rodriguez-Pardo and Elena Garces. Seamlessgan: Self-supervised synthesis of tileable texture maps. *IEEE Transactions on Visualization and Computer Graphics*, 29 (6):2914–2925, 2023. 3

[23] 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. 2, 3, 5

[24] Carlos Rodríguez Pardo, Sergio Suja, David Pascual, Jorge Lopez-Moreno, and Elena Garces. Automatic extraction and synthesis of regular repeatable patterns. *Computers and Graphics*, 83:33–41, 2019. 3

[25] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models, 2022. 3, 4, 5

[26] Chitwan Saharia, Jonathan Ho, William Chan, Tim Salimans, David J. Fleet, and Mohammad Norouzi. Image super-resolution via iterative refinement, 2021. 3

[27] Sam Sartor and Pieter Peers. Content-aware tile generation using exterior boundary inpainting. In *ACM Transactions on Graphics*, 2024. 3

[28] Christoph Schuhmann, Richard Vencu, Romain Beaumont, Robert Kaczmarczyk, Clayton Mullis, Aarush Katta, Theo Coombes, Jenia Jitsev, and Aran Komatsuzaki. Laion-400m: Open dataset of clip-filtered 400 million image-text pairs, 2021. 5

[29] Joshua E.S. Socolar and Joan M. Taylor. An aperiodic hexagonal tile. *Journal of Combinatorial Theory, Series A*, 118(8): 2207–2231, 2011. 2

[30] Jascha Sohl-Dickstein, Eric A. Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics, 2015. 2

[31] tjm35. Asymmetric tiling sd webui. https://github.com/tjm35/asymmetric-tiling-sd-webui, 2023. 3, 5

[32] Giuseppe Vecchio, Rosalie Martin, Arthur Roullier, Adrien Kaiser, Romain Rouffet, Valentin Deschaintre, and Tamy Boubekeur. Controlmat: A controlled generative approach to material capture, 2023. 2

[33] Nir Zabari, Aharon Azulay, Alexey Gorkor, Tavi Halperin, and Ohad Fried. Diffusing colors: Image colorization with text guided diffusion, 2023. 3

[34] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models, 2023. 3, 5

[35] Qinsheng Zhang, Jiaming Song, Xun Huang, Yongxin Chen, and Ming-Yu Liu. Diffcollage: Parallel generation of large content with diffusion models, 2023. 3

[36] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric, 2018. 6

[37] Xilong Zhou, Miloš Hašan, Valentin Deschaintre, Paul Guerrero, Kalyan Sunkavalli, and Nima Kalantari. Tilegen: Tileable, controllable material generation and capture, 2022. 3

[38] Yang Zhou, Zhen Zhu, Xiang Bai, Dani Lischinski, Daniel Cohen-Or, and Hui Huang. Non-stationary texture synthesis by adversarial expansion. *ACM Transactions on Graphics*, 37(4):1–13, 2018. 3

[39] Yang Zhou, Kaijian Chen, Rongjun Xiao, and Hui Huang. Neural texture synthesis with guided correspondence. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18095–18104, 2023.

[40] Yang Zhou, Rongjun Xiao, Dani Lischinski, Daniel Cohen-Or, and Hui Huang. Generating non-stationary textures using self-rectification, 2024. 3