ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

Dipartimento di Informatica - Scienza e Ingegneria (DISI)

CORSO DI LAUREA MAGISTRALE IN INGEGNERIA INFORMATICA

DIFFUSION-BASED MODEL FOR TEXTILE GENERATION

Tesi di laurea magistrale in
Fondamenti di Computer Graphics M

Relatrice

Prof.ssa Serena Morigi

Presentata da

Federica Di Giaimo

Correlatore

Paolo Zuzolo

Sessione marzo 2026

Anno Accademico 2025/2026

Contents

Introduction

As real-time rendering technology has matured, the video game industry has witnessed a fundamental shift in production priorities. AAA (Triple-A) titles, high-budget productions developed by major studios with teams of hundreds of professionals, have pushed the pursuit of photorealism to increasingly higher standards, developing production pipelines in which the management of textile materials across character wardrobes and environmental surfaces represents a key technical and artistic challenge. The production of these materials represents a critical bottleneck in contemporary game development, representing a substantial investment and requiring teams of specialized artists to achieve the visual fidelity expected by today's audiences. As hardware capabilities continue to expand, the principal challenge has shifted from raw rendering power to the generation of sufficient high-quality content to populate digital worlds without visual repetition.

Digital textiles represent a unique challenge within this landscape. Unlike solid surfaces such as metal or stone, fabrics exhibit hierarchical structure from the macro-scale weave pattern down to individual fiber interactions. This complexity manifests in microfacet scattering, anisotropic reflectance and subsurface light transport that require specialized mathematical formulations. The visual properties of textiles emerge from geometric structure rather than pigmentation alone, making them particularly difficult to synthesize through conventional procedural or example-based methods.

Texture synthesis has evolved from procedural noise to neural approaches. While Generative Adversarial Networks enabled photorealistic synthesis, training instability led to the adoption of Latent Diffusion Models, which offer superior stability through iterative denoising in compressed latent space. Contemporary diffusion-based methods have demonstrated native tileability through architectural modifications and visual conditioning mechanism have enabled control beyond text prompts, allowing direct specification of desired material properties through reference images. While these advances represent significant progress, the application of general-purpose solutions to textile synthesis reveals domain-specific challenges that warrant further investigation.

The expansion of small textile samples to high-resolution tileable outputs while preserving structural coherence remains problematic. Existing methods produce tiling artifacts at boundaries, generate visual repetition patterns detectable to the human eye, or require extensive manual post-processing to achieve production quality. Furthermore, the diversity of textile topologies, from highly structured geometric weaves to irregular organic surfaces like leather and felt, necessitates adaptive synthesis strategies rather than monolithic approaches. This gap between general-purpose image generation capabilities and production-ready textile synthesis motivates the present work.

The objective of this thesis is the evaluation of a latent diffusion model based pipeline for expansion of textile textures. The work investigates whether automated methods can produce 1024×1024 resolution textures that are natively tileable and structurally consistent with 256×256 source samples, achieving the quality and controllability traditionally associated with manual creation.

The central hypothesis posits that coordinated interventions at different stages of the generation process can resolve the competing demands of tileability, structural coherence and visual quality. The methodology introduces three techniques: spatial manipulation to prevent boundary artifacts, architectural modifications to enforce seamless periodicity and structured initialization to preserve source characteristics during expansion.

The experimental framework operates within a large-scale diffusion model trained on diverse visual data, leveraging visual guidance mechanisms to enable precise material specification beyond text-based descriptions. This approach addresses the fundamental limitation of natural language in describing complex microscopic surface properties.

Given the substantial diversity of textile patterns, the workflow accommodates three distinct categories. For textures with regular patterns, such as structured weaves and geometric fabrics, and for textures with high-frequency details, like denim and jersey, the methodology employs the complete pipeline described above, utilizing visual guidance to maintain material identity and structural consistency throughout expansion. For textures with irregular patterns, such as organic materials, leather and non-periodic surfaces, the approach shifts to direct outpainting from 512×512 samples to 1024×1024 resolution. In this configuration, visual guidance enables control over pattern density and orientation, ensuring coherent visual continuity despite the absence of strict geometric periodicity.

The work demonstrates that native tileability without post-processing is achievable while maintaining geometric integrity throughout semantic upscaling. The dual-pathway framework proves effective across both pattern types, with each workflow preserving material identity and structural coherence appropriate to its topology. The validation of temporally coordinated interventions establishes precise timing windows where architectural modifications yield optimal results, providing a toolkit that allows artists to control output characteristics by selecting visual references.

The main contributions of this work are:

1. A dual-pathway synthesis framework that adapts generation strategies based on textile topology, enabling both structured geometric weaves and irregular organic materials to be processed through topology-appropriate workflows.

2. Temporal coordination of architectural interventions within the diffusion process, establishing precise timing windows where tileability modifications yield optimal results without degrading visual quality or structural coherence.

3. Integration of visual guidance mechanisms for material-specific control, providing artists with a toolkit to influence output properties through reference image selection rather than complex parameter tuning.

4. Empirical validation demonstrating that native tileability without post-processing is achievable while maintaining geometric integrity throughout semantic upscaling from source samples to production-resolution assets.

This thesis is organized into five chapters that progress from theoretical foundations to practical implementation and empirical evaluation.

- Chapter 1: The Texture Synthesis Task in Computer Graphics establishes the industry context, exploring Physically Based Rendering principles, AAA production workflows and mathematical challenges in rendering anisotropic materials.

- Chapter 2: Generative AI Architectures provides mathematical foundations for Variational Autoencoders, Generative Adversarial Networks and Diffusion Model forward and reverse processes.

- Chapter 3: State of the Art in Texture Generation critically examines historical methods from procedural noise to neural style transfer, identifying limitations that motivate this thesis.
- Chapter 4: A Diffusion-Based Model for Textile Generation constitutes the technical core, describing toolkit development, IP-Adapter integration and algorithms for seamless latent manipulation.
- Chapter 5: Results, Discussion and Conclusion evaluates model performance across diverse textile samples, discusses implications for industrial practice, acknowledges limitations and outlines future research directions.

The complete source code of the project is available on GitHub:
https://github.com/federicadigiaimo/textile-generation

Chapter 1

The texture synthesis task in Computer Graphics

Texture synthesis represents one of the most widespread challenges in computer graphics, serving as a fundamental pillar for enhancing the realism of virtual objects and optimizing the management of visual detail in fields such as gaming and film production. At its core, the process involves taking a source texture as input and generating new output images that maintain visual consistency and structural continuity with the original sample. The primary objectives of these techniques are typically twofold: to facilitate spatial expansion through outpainting and to accelerate the artistic production pipeline by automating the creation variations.

Traditionally, texture synthesis was addressed through procedural or example-based algorithms (see Section 3.1) that relied on local pixel neighborhoods or patch-based optimization to reconstruct patterns. However, these methods often struggle with complex structures. With the advent of deep learning, the paradigm has shifted toward generative modeling, where neural networks learn the underlying distribution of the properties of atexture to synthesize new instances.

Modern synthesis workflows leverage the power of generative architectures, a class of machine learning methods that learn the underlying patterns and probability distributions of training data to create new original instances. By navigating a high-dimensional latent space, these models can capture both the micro-geometric weave of a textile and its variations, providing a more robust solution for asset production.

1.1 Asset Production in Game Development

1.1.1 The AAA Workflow

In the context of AAA game development, the production of assets is a complex workflow that accounts for a significant portion of the development budget. The standard pipeline for asset production typically involves several stages, beginning with the definition of concept art and the collection of real-world material references to ensure visual authenticity.

Artists then proceed to high-poly mesh modeling, often utilizing digital sculpting software to capture details such as seams, pleats and the folds of the fabric. Once the high-resolution geometry is finalized, a subsequent phase of retopology is needed to generate an optimized low-poly version of the mesh suitable for real-time rendering. This stage is

crucial for ensuring that the asset maintains a manageable triangle count while preserving the silhouette.

To bridge the gap between these two versions, the low-poly model undergoes UV unwrapping, which is the process of flattening the 3D surface into a 2D coordinate system. This allows for the baking process, where the features of the high-poly model are projected onto the low-poly surface and stored in texture maps, such as Normal and Ambient Occlusion maps.

However, the creation of textures for fabrics remains particularly challenging due to the complex interaction of light with the geometry of woven materials. The demand for photorealism in modern titles has driven artists to seek more sophisticated methods for texture generation. Characteristics like imperfections of natural fibers and weave patterns demands a level of granularity that is difficult to achieve efficiently through conventional processes.

1.1.2 The Complexity of Textile Representation

In modern game development, textile representation has evolved from simple maps to multi-layered shading systems. Unlike solid surfaces, fabrics possess a hierarchical structure composed of fibers. The response to light requires mathematical approaches to achieve photorealism, as standard approximations fail to capture the nuances of woven materials.

Traditional Physically Based Rendering (PBR) models are optimized for solid surfaces where light behavior is confined to the surface interface. However, fabrics such as cotton, wool and velvet exhibit light scattering and inter-fiber occlusion. Anisotropy is a defining characteristic of these materials, where the surface appearance changes based on its rotation around the normal, caused by the directional alignment of threads. Capturing this behavior necessitates advanced shading techniques, such as the Ashikhmin-Shirley formulation [1], which accounts for fiber orientation. While visually superior, these frameworks increase the complexity of the rendering pipeline and the memory footprint, a challenge in real-time applications where performance is paramount. The implementation of these anisotropic models relies fundamentally on Microfacet Theory, which provides the statistical basis for modern reflectance. This framework assumes that a surface consists of numerous microscopic mirrors, each reflecting light according to its local orientation. The overall appearance is governed by a distribution function $D(m)$, which describes the alignment of these facets relative to the surface normal.

While standard microfacet models utilize isotropic distributions for solid materials, textile representation requires a directional approach. In fabrics, the microfacets are not randomly oriented but follow the longitudinal axis of the fibers. This alignment is essential to simulate the specular highlights that appear on individual yarns. By integrating microfacet logic with the anisotropic characteristics discussed earlier, the rendering pipeline can capture the complex interaction between light and weave. This synthesis allows for the digital reconstruction of the shifting brightness observed in silk.

1.2 Modern Real-time Rendering for Textures

1.2.1 Detail Mapping

In AAA titles, storing high-resolution textures for every asset is unsustainable, as it would consume gigabytes of Video RAM (VRAM). To mitigate hardware limitations, technical artists rely on Detail Mapping to increase perceived resolution and sharpness without increasing the memory footprint. Rather than baking a 4096×4096 texture for a garment, the workflow leverages a lower-resolution macro-map, such as 1024×1024, to define large-scale features like seams and color variations. The innovation resides in the spatial decoupling of texture frequencies, whereby the macro-map is then blended within the shader with a tiling micro-map containing the fine-grained pattern.

Modern engines implement this technique through specialized shading systems. In *Unity's High Definition Render Pipeline (HDRP)*, detail distribution is managed via a Mask Map [2]. This packed texture requires specific channels, typically the blue channel, to act as a Detail Mask, as shown in Figure 1.1, controlling the intensity and placement of micro-maps across the surface. Similarly, *Unreal Engine 5* uses Detail Textures [3] within the Material Graph, often employing secondary normal maps to maintain surface fidelity at close range. This layered approach ensures that assets maintain visual clarity regardless of camera proximity, as the tiling micro-map compensates for the lower texel density of the macro-map.

Generative AI bridges the gap between the efficiency of tiling and the need for interesting details. While procedural tools rely on mathematical noise, AI models can synthesize features tailored to specific asset geometry, ensuring the macro-map remains non-periodic. This approach not only reduces production time compared to manual painting but also provides the structural foundation required for high-density assets, effectively scaling the creation of textiles within modern memory constraints.

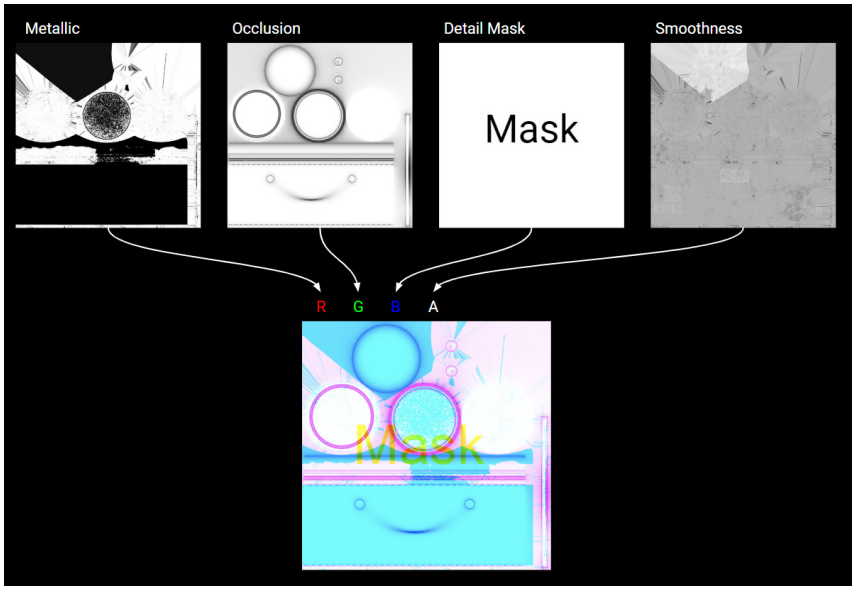


Figure 1.1: An example of Mask Map in Unity. Source: [2]

1.2.2 Micro-Detail Maps

In advanced production, basic mapping is often insufficient to represent complex textiles. Studios such as Naughty Dog, in the development of titles like *Uncharted 4* (2016) and *The Last of Us Part II* (2020), utilize micro-detail maps to simulate the interaction between light and fibers [4]. These maps harness extreme tiling, ranging from 64 to 128 repetitions, to represent fabrics. The objective is to define the response of the material rather than its color.

Unlike macro-maps, whose purpose is to influence colours and normals, micro-detail textures focus on specular occlusion. This process suppresses highlights within fabric cavities, preventing an unnatural appearance under intense lighting. By modulating the specular response, these maps eliminate the plastic effect common in interactive rendering. This approach allows for a separation between materials based on their reflectance. Consequently, the integration of these maps ensures that fabric shading remains consistent with physical principles, enhancing the realism of character garments.

Recent titles, including *Alan Wake 2* (2023) and *Senua's Saga: Hellblade II* (2024), have evolved this concept by integrating these maps into shader parameters such as fuzz and sheen. These attributes simulate light transport across fibers, replicating the glow observed on wool. In the Northlight Engine, the framework behind *Alan Wake 2* [5], a specialized model simulates the back-scattering of light, replicating the soft appearance of surfaces. This effect is evident in the garments of the protagonists, where highlights emerge at grazing angles due to anisotropy. By combining detailed data with these reflectance models, modern engines achieve a level of tactile realism that surpasses traditional shading. This granular information prevents specular highlights from appearing uniform, preserving the organic texture of the weave. By distributing occlusion data, these modern shaders maintain a physical response that scales with proximity.

1.2.3 Multilayer Shaders

Multilayer shaders represent an advanced paradigm in material authoring, designed to maximize visual variety while minimizing Video RAM (VRAM) consumption. A prominent implementation of this technology is found in *Cyberpunk 2077* (2020), developed by CD Projekt RED, where the rendering of complex environments and character garments relies on a layered material architecture rather than unique textures. In this title, the layered structure ensures that character clothing reacts realistically to ray-traced lighting. The innovation in current pipelines resides in the complexity of this material stacking. While earlier titles exploited a limited number of detail maps, modern hardware allows for multiple concurrent layers. This enables the simultaneous representation of fabric weave, surface dust and localized wear.

The system functions by combining three primary components within a single shader. First, a library of Material templates provides standardized PBR data for base materials such as cotton, leather or metal. Second, Layer Masks are grayscale textures that define the spatial distribution of these materials across the asset. Finally, a multilayer setup configures the blending order and parameters for up to 20 individual layers. To maintain aesthetic consistency at close range, the system also implements Microblends, which are tiled normal maps that add high-frequency detail during the transition between layers.

By integrating generative synthesis into this multilayer pipeline, developers can produce textiles that exhibit variations while adhering to the strict memory constraints of real-time engines.

1.2.4 Evolution of Texture Compression

Texture expansion aligns with emerging compression technologies designed to handle high-density data. While current pipelines utilize detail mapping to manage memory, neural-driven optimization allows for the storage of expansive surfaces without traditional constraints. This suggests a framework where generative synthesis provides visual variety and neural compression ensures real-time viability.

NVIDIA presented Neural Texture Compression (NTC) at SIGGRAPH 2023 [6], introducing a neural-based approach to random-access texture compression. While much of the technology is still in the research phase, it has already seen early production adoption in custom engine pipelines, notably within the Anvil engine for titles such as *Assassin's Creed Mirage*, from Ubisoft. Instead of storing pixel data, NTC transforms a PBR texture set, comprising up to 16 channels, into a compact representation consisting of weights for an MLP (Multilayer Perceptron) decoder and a spatial tensor of features.

During rendering, the shader samples the feature tensor and executes the MLP inference to reconstruct the texel color. This technology demonstrates that AI-driven expansion is a critical component of the future pipeline, as it provides the high-density content that neural compressors are designed to handle. By compressing these channels as a single bundle, NTC achieves bitrates of 5 bits per texel (bpp), providing quality comparable to block compression while reducing the VRAM footprint by 70% to 90%. Consequently, hardware memory is no longer a barrier to complexity, but a frontier expanded by the integration of neural networks throughout the asset pipeline.

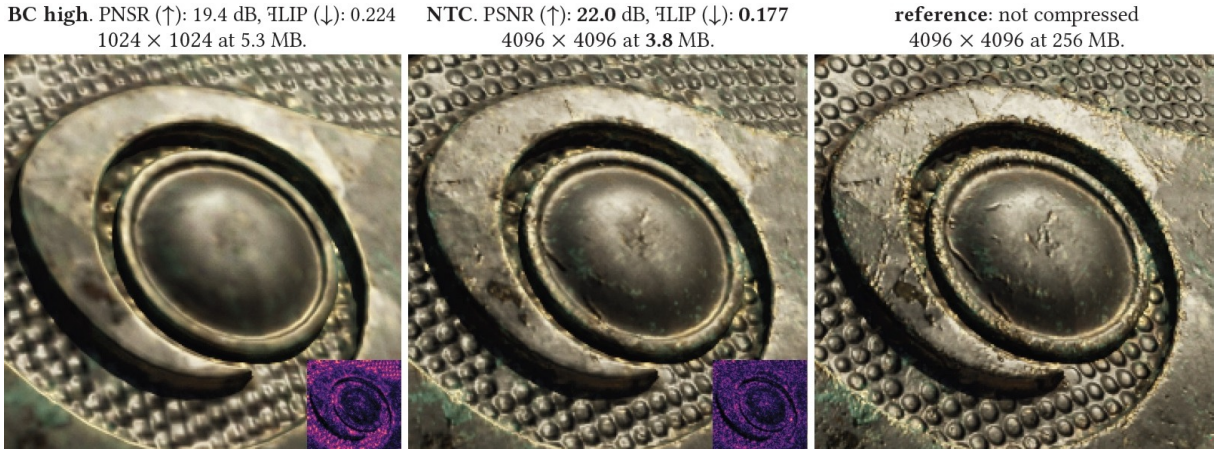


Figure 1.2: Comparison of texture fidelity with and without NTC in production environments. Source: [6]

1.3 Geometric Texture and Image Texture

Geometry analysis is central to computer graphics, particularly regarding geometric texture. This term describes physical details, consisting of shape variations, that remain present across scales. In a textile context, geometric textures include leather pores, wear creases, or the individual yarns of jute fabrics. Unlike purely visual features, geometric texture is an inherent tri-dimensional property characterized through geometric fields, such as normals and curvatures.

To manifest these features in a digital environment, rendering engines utilize techniques such as displacement mapping and tessellation. These methods effectively move or generate surface vertices to match the geometric data, providing high fidelity at the cost of significant computational burden on the GPU.

Image textures represent a more efficient alternative, where reflectance and shading are used to simulate geometric depth. Instead of altering the mesh, these textures operate as a stack of data channels. For instance, normal maps modify the orientation of surface normals to create an optical illusion of depth, while occlusion maps bake shadows into the texture to simulate the crevices between fibers.

The resolution of these maps is measured in texels, the individual units of a texture. Technical artists focus on texel density to ensure clarity at various distances. This decoupling of visual detail from vertex density allows for the representation of high-frequency micro-structures, such as thread weaves, at a fraction of the processing cost required by true geometry.

Figure 1.3: Difference between image and geometric texture. Source: [7]

The industry standard for real-time applications involves a compromise where image textures are preferred over geometric displacement. This choice is dictated by performance constraints, as simulating individual textile fibers via geometry is often unsustainable. However, traditional pipelines often fail to capture the structural richness of physical fabrics through static layers. As established in Section 1.1, generative models offer a new paradigm by synthesizing visual representations that maintain the semantic depth of true geometry, providing a more sophisticated transition from faked appearance to structural presence.

1.4 PBR texture generation

Physically Based Rendering (PBR) is the industry standard for visual consistency under varying lighting conditions. Unlike earlier shading models, PBR simulates light-surface interaction using physical principles, primarily the law of energy conservation [8]. This principle dictates that the total reflected light cannot exceed the amount of light received, preventing materials from appearing unrealistically bright under intense illumination.

The workflow relies on independent maps that isolate specific material properties from environmental lighting [9]. In the Metal-Roughness workflow, Base Color defines the surface's intrinsic reflectance without baked-in lighting information. Roughness describes microscopic surface irregularities, determining the diffusion of reflected light, while Metallic

distinguishes conductive from dielectric surfaces. Additionally, the Normal map encodes surface orientations to simulate depth, as discussed in Section 1.2.

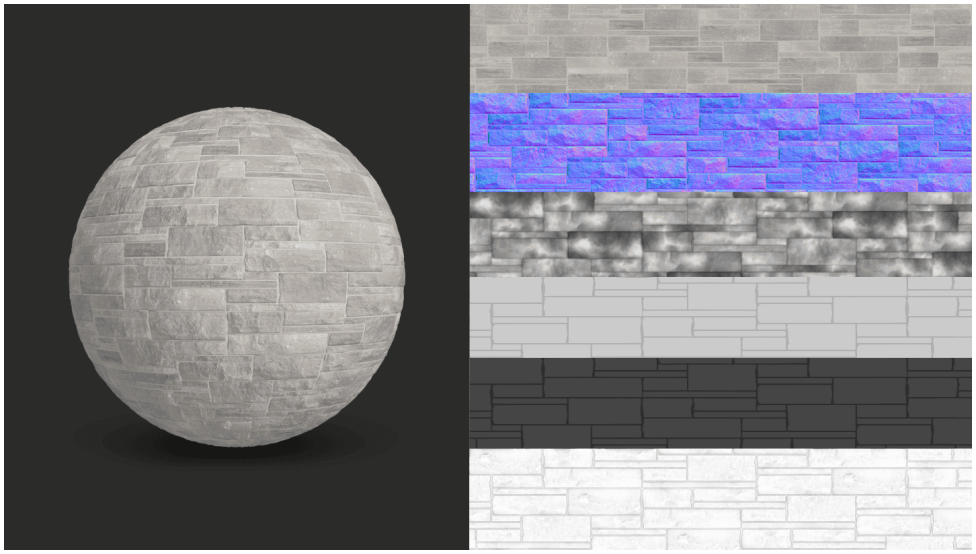


Figure 1.4: PBR material. Source: [10]

Generating these sets requires maintaining spatial coherence between channels. For example, weave patterns in textiles must align perfectly across the Base Color, Normal and Roughness maps to avoid visual artifacts. While precise, this process requires significant expertise to manually simulate realistic fabric imperfections.

Although not addressed in this work, generative synthesis involves the co-generation of complete PBR sets and could produce inherently aligned maps from a single reference. This approach ensures semantic consistency across the material stack, eliminating the need for manual map derivation. Automating this interdependence bridges the gap between procedural authoring and automated production, establishing a robust pipeline for digital textiles.

1.5 Traditional tiling and its limitations

Tiling is the process of repeating a texture sample across a mesh to cover large surfaces with minimal memory consumption. By repeating a seamless texture tile, developers can simulate continuous materials while maintaining a low memory footprint. This approach is the industry standard for assets where the texture's visual scale is significantly smaller than the surface area, allowing for high-detail mapping without the need for unique textures.

The technical implementation of tiling relies on the UV coordinate system of the mesh. When the UV wrap mode is set to repeat, the rendering engine continues to sample the texture beyond the 0 to 1 range. This allows a single 512×512 texture to cover an entire asset. However, this efficiency introduces the primary limitation of the technique, which is the visible repetition of patterns.

The human visual system is highly efficient at detecting periodic patterns. When a texture tile contains recognizable features, the repetition creates a visible grid. This "tiling artifact" breaks visual immersion, as the surface appears as a collection of identical blocks

rather than a continuous material. Even when the edges of a tile are perfectly seamless, the recurrence of internal details reveals the artificial nature of the surface.

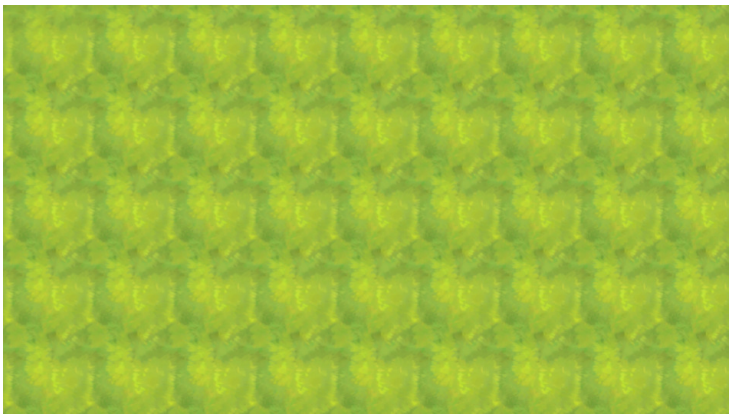


Figure 1.5: The so-called "wallpaper effect" with a texture representing a landscape. Source:[11]

Textiles are particularly sensitive to these artifacts due to their non-stationary properties. In physical fabrics, weave density and dye variation occur across the surface, creating fluctuations. Traditional tiling forces these variations into a geometric repetition. This process results in diagonal seam lines and block boundaries that are clearly visible to the eye. Consequently, the stationary assumption of classical tiling fails to capture the richness of textile materials.

This failure necessitates the adoption of dynamic rendering strategies. To mitigate periodic artifacts, modern engines integrate stochastic methods such as texture bombing. This technique addresses the static nature of tiling by distributing multiple instances of a texture across the surface. By utilizing random parameters, the shader breaks the regularity of the grid within each virtual cell. For every cell, the algorithm applies an offset and rotation to the texture sample, obscuring the repetitive landmarks.

While texture bombing solves the problem of repetition, it requires a careful balance between visual variety and rendering performance. The GPU executes multiple texture fetches for every pixel, which increases the computational load. However, this overhead is often acceptable to preserve VRAM in current pipelines. While effective for unstructured materials, this method struggles with woven textiles where thread orientation must remain consistent. Despite these challenges, texture bombing represents a standard solution for surfaces, providing a foundation for more advanced synthesis methods.

1.6 Tileable Textures

A texture is seamless when its margins exhibit continuity, allowing for repetition without visible boundaries. This property is achieved by aligning pixel data across opposite edges, typically through manual or procedural methods. While seamlessness is a technical requirement for textiles, it does not guarantee a natural appearance.

The perception of periodicity depends on the scale of the pattern relative to the surface area. High-frequency details, such as wool fibers, are inherently repetitive, yet they appear uniform when the pattern scale is small relative to the total area. In a wool knit, the structural repetition of the yarn is visible, yet the material doesn't necessarily appear periodic to the observer. However, intermediate-scale patterns, including directional

weave flows or color shifts, create recognizable landmarks. When these landmarks repeat across a grid, the visual system detects the underlying tile structure regardless of edge alignment. To maintain immersion, a tileable texture must balance margin continuity with the suppression of recognizable features.

Historically, texture mapping relied on this single-layer tiling approach, which often resulted in a lack of visual identity. To overcome the limitations of the seamless paradox, the industry evolved toward a multi-layered workflow like those mentioned in Section 1.2. However, the manual creation of these layers remains a bottleneck in production, defining the need for more efficient synthesis methods.

Despite the rise of unique textures, tiling remains a fundamental technique for hardware efficiency. Modern pipelines treat the tile as a modular primitive rather than a static block. Strategic implementations, such as texture bombing, demonstrate that tiling can support expansive surfaces when managed through procedural distribution. This approach allows for high resolution details without the memory cost of unique maps. Within this framework, generative models do not replace tiling but rather optimize its components. Learning-based methods serves to produce superior samples and intricate masks, providing variations that preserve the efficiency of the rendering engine.

1.7 Texture Expansion

Texture expansion is a synthesis task that extends the boundaries of a source image while maintaining structural continuity. Unlike tiling, which relies on the repetition of a fixed sample, expansion generates new pixel data that evolves from the original context. This process circumvents periodicity, as the synthesized area incorporates features that don't necessarily recur across the surface.

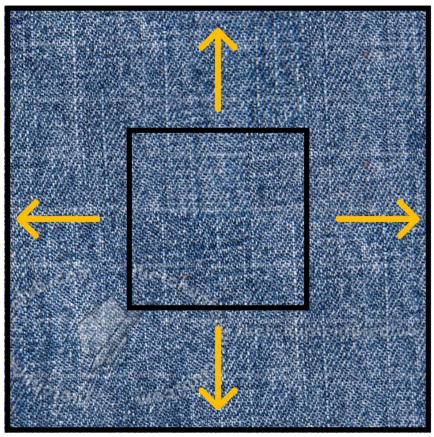


Figure 1.6: The process of outpainting.

The mechanism relies on generative outpainting, where models predict content beyond the initial frame. This extrapolation preserves the spatial logic of the source material. By analyzing the neighborhood, the algorithm ensures that the transition between the original seed and the expanded region remains imperceptible. This continuity creates a surface that maintains coherence without seams.

The efficacy of this process resides in the ability to manage diverse geometric scales. For high frequency details, generative models ensure statistical variance. Rather than

duplicating identical pixels, the algorithm synthesizes instances that preserve the material density without mathematical perfection. While current implementations still face challenges regarding structural drift or micro-detail sharpness, often due to the loss of global context in long-range outpainting, expansion represents a significant evolution.

These capabilities extend to intermediate scale patterns. By understanding the structure of the weave, the synthesis facilitates the evolution of patterns across expansive areas. This approach also resolves the challenge of non-stationary textures. Physical fabrics often exhibit global gradients or subtle variations that tiling cannot replicate. Generative models capture these high-level characteristics, producing a signal that remains diverse yet consistent. By synthesizing variety, expansion provides the density necessary for digital environments, transforming a static sample into a continuous surface.

Beyond the generation of pixels, the system enables creative control through latent space manipulation. Generative models represent textures within a multidimensional manifold where coordinates correspond to physical attributes. By navigating this space during the process, artists can modulate the characteristics of the expanded region. This control allows for the introduction of localized variations without breaking the underlying structure. Unlike procedural noise, latent manipulation preserves the integrity of the fabric while providing artistic flexibility. This integration transforms texture expansion into a tool for material design, offering a level of customization that static samples cannot achieve.

The expanded surface then functions as a generator for diverse seeds. By extending a small sample into a larger surface, the algorithm produces a reservoir for data. For instance, expanding a 256×256 sample into a 1024×1024 image creates a vast area of aperiodic data. From this region, artists can extract multiple patches to serve as independent inputs for further synthesis. Each extracted region maintains local variations, creating a library of consistent assets. Consequently, expansion transforms a limited sample into a collection of varied textures, maximizing the utility of the source data.

Chapter 2

Generative AI Architectures

In the field of machine learning, two primary probabilistic approaches are used to model the relationship between input data (x) and output data (y): generative models and discriminative models. These represent a fundamental distinction in how the model characterizes the data and, consequently, the type of task it is designed to solve.

Discriminative models focus directly on learning the conditional distribution $p(y \mid x)$, which represents the probability of an output given the inputs. This approach is intrinsically linked to the task of classification: the model learns to identify the decision boundary that best separates different classes in the feature space, disregarding how the data was generated and focusing only on the features that distinguish one label from another. For this reason, they are the standard choice for classification and regression tasks.

Generative models, on the other hand, aim to learn the underlying distribution of the data, either by modeling the marginal distribution $p(x)$ or the joint distribution $p(x, y)$ between inputs and outputs. This shift from identifying boundaries to modeling density enables the transition from classification to generation. Instead of merely labeling an input, these models acquire the ability to synthesize samples that are statistically consistent with the training set, a capability that has led to powerful tools for synthesizing images, videos and music.

Another fundamental distinction concerns the learning paradigm employed by these models. The most widely adopted form is supervised learning, where a model learns a mapping from inputs to outputs based on a dataset of labeled pairs. This is the framework where discriminative classification typically operates. However, this approach is often bottlenecked by the need for millions of human-labeled examples. Given that the vast majority of real-world data is unlabeled, there has been a significant shift toward unsupervised learning to reduce the reliance on human intervention.

A cornerstone of unsupervised learning is generative modeling. In this framework, training samples are assumed to be drawn from an unknown distribution $p_{data}(x)$. The objective is to construct a mathematical proxy, known as the Probability Density Function (PDF), denoted as $p_{model}(x)$.

In probability theory, a PDF is a function that describes the relative likelihood of a continuous random variable x taking on a specific value. For high-dimensional data like textile images, the PDF assigns higher values to configurations of pixels that represent realistic textures and near-zero values to random noise. In the context of deep learning, this distribution is parameterized by θ, written as $p_{model}(x; \theta)$. Here, θ represents the set of all learnable parameters within the neural network, specifically the weights and biases.

By changing θ, the geometry of the density function p_{model} is reshaped to better match the patterns observed in the real data.

The goal of the learning process is to minimize the discrepancy between the true distribution $p_{data}(x)$ and the parameterized model $p_{model}(x;\theta)$. This is typically achieved by minimizing the Kullback-Leibler (KL) divergence, a measure of how one probability distribution p_{model} diverges from an expected distribution p_{data}. For continuous distributions, it is defined as:

$$D_{KL}(p_{\text{data}} \parallel p_{\text{model}}) = \int p_{\text{data}}(x) \log \left(\frac{p_{\text{data}}(x)}{p_{\text{model}}(x;\theta)} \right) dx \tag{2.1}$$

Intuitively, this value represents the information loss incurred when p_{model} is used to approximate p_{data}; a divergence of zero indicates that the two distributions are identical. The training process thus becomes an optimization problem: the cyclic update of the weights and biases θ until the model's PDF best fits the empirical data distribution.

However, evaluating $p_{model}(x;\theta)$ directly in deep neural networks is often computationally intractable. To be a valid PDF, the function requires solving complex multidimensional integrals. To circumvent this and learn θ without solving these integrals directly, different strategies have emerged:

- Energy-Based Models address intractability by avoiding normalization altogether. They define a non-normalized scalar energy function, where low energy corresponds to high compatibility and high energy to low compatibility.
- Variational Autoencoders rely on variational inference, designing a learning algorithm based on an approximation of the density function.
- Generative Adversarial Networks represent a shift toward implicit modeling: instead of explicitly evaluating or approximating a density function, they use a game theory approach where a generator learns to draw samples that a discriminator cannot distinguish from real data.
- Diffusion Models tackle intractability by decomposing the generation process into a sequence of iterative denoising steps. Instead of calculating the density in a single pass, they learn to reverse a progressive noise process, effectively navigating the distribution space through score-based modeling.

Table 2.1: Comparison between Discriminative and Generative Approaches.

Feature	Discriminative Approach	Generative Approach
Learning Paradigm	Mostly Supervised Learning	Mostly Unsupervised / Self-supervised
Probability Goal	Maps input to labels: $p(y \mid x)$	Models data distribution: $p(x)$ or $p(x, y)$
Primary Task	Classification and Regression	Generation and Synthesis
Objective	Finding decision boundaries	Learning the underlying structure
Textile Example	Identifying fabric types (e.g., silk vs. cotton)	Synthesizing new patterns and textures

2.0.1 Convolutional Neural Networks

Before exploring how specific models like Variational Autoencoders or Diffusion Models navigate the probability space, the Convolutional Neural Network (CNN) must be defined. While originally perfected for discriminative classification, the ability of the network to extract hierarchical features represents the core mechanism that generative models leverage to synthesize complex images. Unlike fully connected layers, which disregard the spatial arrangement of pixels, a CNN applies a set of learnable kernels that slide across the input image. This operation produces a feature map: a spatial representation that encodes the presence and relevance of specific visual patterns across the input. Each value within this map, known as an activation, indicates the degree of correlation between a local neighborhood of pixels and the learned kernel, effectively translating raw visual data into a structured grid of detected features. Mathematically, for a 2D input image I and a kernel K of size $m \times n$, the value at position (i, j) of the resulting feature map S is given by:

$$S(i,j) = (I * K)(i,j) = \sum_m \sum_n I(i+m, j+n)K(m,n) \tag{2.2}$$

By stacking multiple convolutional layers, the network builds a hierarchical representation of this data. In this progression, initial layers typically capture low-level features, while deeper layers combine these activations to represent semantic structures and patterns.

In this context, Variational Autoencoders employ convolutional encoders to compress these high-dimensional textures into structured latent spaces, while Generative Adversarial Networks rely on convolutional discriminators to evaluate the authenticity of synthesized patterns. Similarly, Diffusion Models navigate the distribution space through U-Net architectures, which are composed entirely of convolutional blocks designed to iteratively denoise spatial representations.

The effectiveness of these operations depends heavily on the receptive field, which defines the spatial extent of the input that influences a specific activation. In generative pipelines, the interaction between convolutional layers and padding strategies determines the continuity and coherence of the final output. Since standard zero-padding often introduces boundary discontinuities, the specialized techniques discussed in Section 4.1.3, will build upon this convolutional foundation to achieve seamless tiling and infinite texture synthesis.

2.1 Energy-Based Models

Energy-based models (EBMs) represent a class of generative models that capture the dependencies between variables by associating a scalar value, termed energy, with each configuration of the variables [12]. In this framework, energy serves as a measure of compatibility between variables: configurations that are consistent with the training data are assigned low energy values, while inconsistent ones are assigned higher energy.

Unlike traditional neural networks, which are typically designed to map inputs to specific outputs, the EBM approach treats inference as an energy minimization task. During training, the model minimizes a loss functional designed to shape the energy function, ensuring that the system learns to prefer correct configurations over incorrect ones.

2.2 Variational Auto-Encoders

A Variational Auto-Encoder (VAE) is a generative model that extends the classical autoencoder framework by incorporating principles from Bayesian variational inference [13]. While a deterministic autoencoder maps an input to a single deterministic point in a latent space, a VAE models the latent variables as a continuous probability distribution.

The architecture consists of an encoder $q_\phi(z|x)$, which approximates the distribution of the latent variables z given an input x and a decoder $p_\theta(x|z)$, which reconstructs the input from the latent samples. As established in Section 2.0.1, both the encoder and decoder are typically implemented as CNN to leverage their hierarchical feature extraction capabilities. Typically, the latent distribution is assumed to be a multivariate Gaussian, where the encoder outputs the mean μ and the variance σ.

A fundamental challenge in training VAEs is that the sampling process ($z \sim \mathcal{N}(\mu, \sigma^2)$) is non-differentiable, preventing the use of standard backpropagation. To overcome this, the reparameterization trick is employed. Instead of sampling directly from the distribution, z is expressed as a deterministic transformation of a random noise variable ϵ:

$$z = \mu + \sigma \odot \epsilon, \quad \text{where } \epsilon \sim \mathcal{N}(0, 1) \tag{2.3}$$

This formulation allows gradients to flow through the parameters μ and σ, enabling the optimization of the network using stochastic gradient descent.

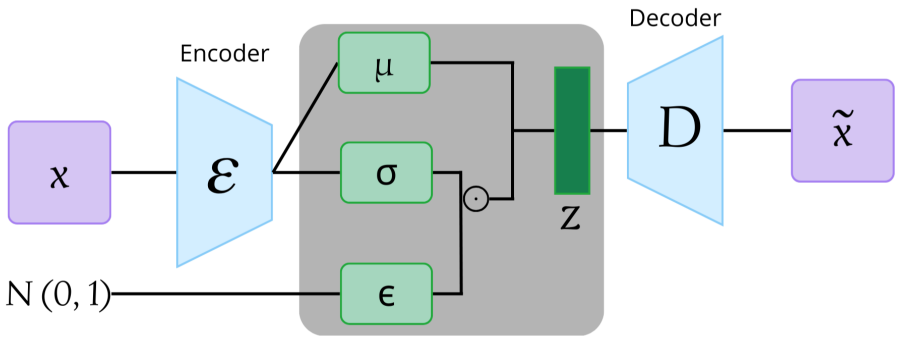


Figure 2.1: VAE illustration. The grey part of the model is the reparameterization trick to compute the latent z. Adapted from [14]

2.2.1 Optimization and the ELBO

The VAE is trained by minimizing a multi-objective loss function derived from the Evidence Lower Bound (ELBO):

$$\mathcal{L}_{VAE} = \mathbb{E}_{q_\phi(z|x)}[\log p_\theta(x|z)] - D_{KL}(q_\phi(z|x)||p(z)) \tag{2.4}$$

The first term is the reconstruction loss (often Mean Squared Error), which ensures the output $\hat{x}$ is as similar as possible to the input x. The second term is the KL divergence, which acts as a regularizer by forcing the learned latent distribution to align with a prior distribution $p(z)$, usually a standard normal distribution $\mathcal{N}(0, I)$.

This regularization ensures that the latent space is continuous and complete: points that are close in the latent space correspond to similar high-level features in the data space. This property enables smooth interpolation between generated samples and provides a structured manifold for diffusion models to operate within.

2.2.2 Perceptual Compression and Limitations

In the context of modern generative pipelines, the VAE acts as a tool for perceptual compression. It filters out high frequency noise and other redundancies, preserving only the essential information. This reduces the dimensionality of the data, significantly lowering the computational requirements for generative steps.

However, a well-known limitation of VAEs is the production of blurry reconstructions. This occurs because the standard reconstruction loss (MSE) penalizes large deviations but tends to average out multiple possible configurations of high-frequency details. This creates a trade-off: while the KL-regularization ensures a well-behaved latent space, it often requires additional components, such as the GAN-based discriminators discussed in Section 2.3, to recover the sharpness required for high fidelity applications like textile synthesis.

2.3 Generative Adversarial Networks

First introduced in 2014 by Goodfellow et al. [15], Generative Adversarial Networks (GANs) represent a landmark in generative modeling. The architecture is rooted in game theory, conceptualizing the training process as a competitive interaction between due neural networks: a generator (G) and a discriminator (D).

The generator learns to map a latent noise vector z, typically drawn from a Gaussian distribution, to the data space. Its objective is to produce synthetic samples $G(z)$ that are indistinguishable from real data. Simultaneously, the discriminator is trained to act as a binary classifier, distinguishing between real samples x from the dataset and fake samples x' generated by G. This adversarial interaction is architecturally grounded in the convolutional principles; specifically, the discriminator leverages hierarchical kernels to scrutinize the structural integrity of the output, while the generator employs a mirrored convolutional structure to synthesize spatial patterns from latent representations. Formally, this relationship is expressed as a minimax objective function $V(D, G)$:

$$\min_G \max_D V(D,G) = \mathbb{E}_{x\sim p_{data}(x)}[\log D(x)] + \mathbb{E}_{z\sim p_z(z)}[\log(1 - D(G(z)))] \tag{2.5}$$

During training, the discriminator aims to maximize the probability of correctly labeling both real and fake examples, while the generator is trained to minimize the probability that the discriminator correctly identifies its outputs as fake.

The primary advantage of GANs is their ability to synthesize high-frequency details and sharp edges. This is a direct consequence of the adversarial loss, which forces the generator to produce textures that can satisfy the structural scrutiny of the discriminator.

Unlike models based purely on pixel-wise reconstruction losses, such as Mean Squared Error (MSE), GANs do not suffer from the regression to the mean effect. While MSE tends to produce blurry outputs by averaging potential pixel configurations to minimize the squared error, the adversarial framework encourages the model to commit to specific visual patterns. This characteristic makes GANs particularly effective for generating photorealistic local textures.

Despite their generative capabilities, GANs are notoriously challenging to optimize due to the non-convex nature of the minimax game. A common failure mode is mode collapse, a condition in which the generator discovers a limited subset of samples that successfully fool the discriminator. In such cases, the model fails to capture the full diversity of

the training distribution, producing repetitive or highly similar outputs regardless of the input noise. Furthermore, the training process is sensitive to the relative learning rates of the two networks, often leading to vanishing gradients if the discriminator becomes too efficient too early in the process.

In modern implementations like Latent Diffusion Models, the adversarial framework is frequently utilized as a refinement mechanism within a larger pipeline. Instead of evaluating the global structure of an image, the system may employ a PatchGAN discriminator, which utilizes a convolutional architecture to penalize discrepancies at the scale of local image patches rather than the entire frame. This approach allows the primary architecture to handle global coherence while the adversarial component ensures the high-fidelity reconstruction of fine-grained details. In the context of textile synthesis, this ensures that intricate patterns remain sharp and perceptually consistent during the final decoding stage.

2.4 Diffusion Models

Diffusion models represent a class of generative probabilistic models designed to learn a data distribution $p(x)$ by iteratively denoising a normally distributed variable. The primary innovation of these models lies in the decomposition of the generation process into a sequence of reversible denoising steps. While previous generative architectures attempted to learn the data distribution directly, diffusion models focus on learning to invert a gradual signal degradation process.

2.4.1 Forward and Reverse Diffusion Phases

The framework is characterized by two distinct phases: the forward diffusion process and the reverse (or generative) process. In the forward phase, Gaussian noise is incrementally added to the training data across T timesteps until the signal becomes indistinguishable from pure noise. This degradation follows a predefined Markov chain where each step adds a small amount of variance, effectively erasing the structural information of the original input.

During the training phase, the model learns the reverse process: it is trained to estimate the noise added at each step t to recover the cleaner version of the signal. Consequently, the model becomes capable of transforming a sample of pure Gaussian noise into a high-fidelity image. This reverse trajectory can be guided by external signals, such as text or image conditioning, allowing for controlled synthesis.

2.4.2 Adversarial Comparison and Computational Trade-offs

Unlike GANs, which rely on a competitive discriminator to guide the generator, diffusion models directly model the data distribution through the aforementioned denoising sequence. This approach offers several advantages, most notably improved training stability and the mitigation of mode collapse.

However, these advantages come at a cost: diffusion models are computationally intensive and require significantly longer inference times compared to GANs. This is due to the iterative nature of the reverse sampling process, which requires multiple evaluations of the neural network to produce a single final sample, whereas GANs can generate an image in a single forward pass.

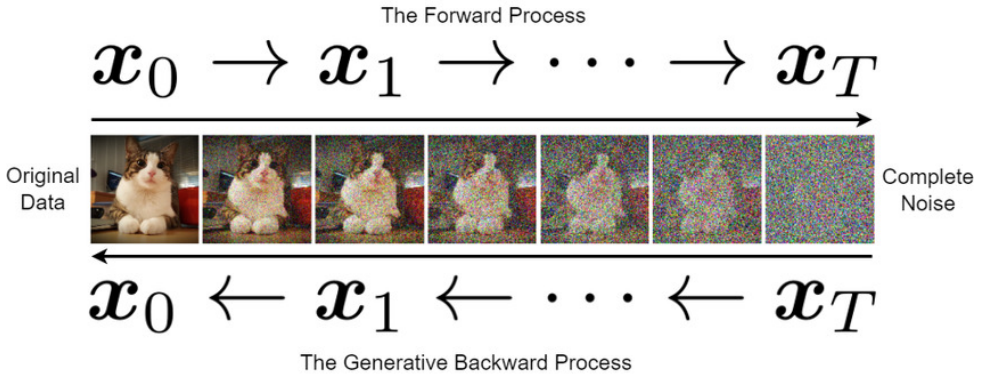


Figure 2.2: Forward and reverse process in diffusion models. Source: [16]

2.4.3 The Noise Prediction Objective

The core of the generative process is typically managed by a neural network, such as a U-Net. This network is trained on a noise prediction objective, where it learns to estimate the specific noise component ϵ added at any given timestep t. By subtracting this noise, the model effectively recovers the underlying structure of the data distribution from an initially chaotic state. This objective allows the model to learn the gradients of the data distribution.

2.4.4 Latent Diffusion Models

One of the primary challenges in generative modeling is the substantial computational overhead and memory requirements associated with processing high-resolution images in the pixel space. Latent Diffusion Models (LDMs), introduced by Rombach et al. [17], mitigate this issue by operating within a compressed latent space rather than performing the diffusion process directly on pixels. This paradigm shift significantly reduces the computational burden while maintaining, or even enhancing, the quality of the generated samples.

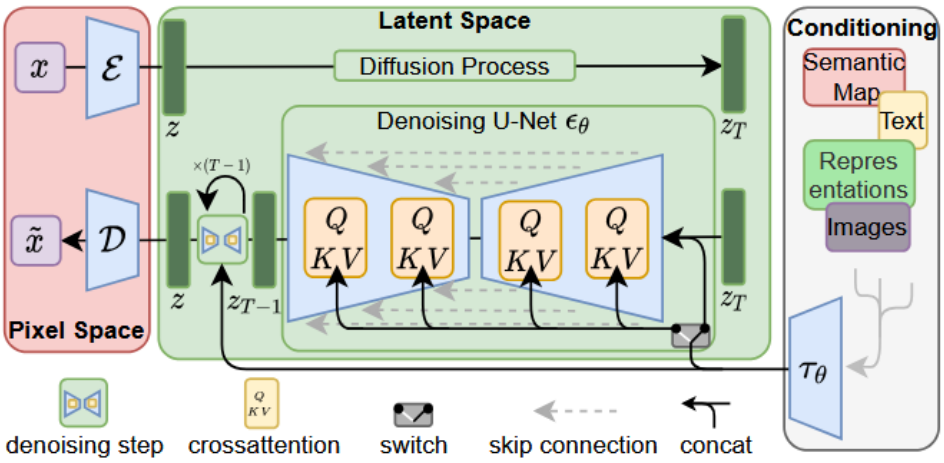


Figure 2.3: LDM architecture. Source: [17]

The LDM pipeline can be decomposed into two distinct operational phases:
Training Phase

1. Perceptual Compression: The VAE encoder $\mathcal{E}$ compresses the input image x from pixel space to a lower-dimensional latent representation $z_0 = \mathcal{E}(x)$.

2. Forward Diffusion: Gaussian noise is progressively added to z_0 over T timesteps, producing increasingly noisy latent states $z_1, \ldots, z_T$.

3. Conditioning Preparation: External modalities (text, class labels, images) are encoded into conditioning signals τ_θ via pretrained encoders.

4. Noise Prediction: A U-Net backbone ϵ_θ with cross-attention mechanisms learns to predict the noise added at each timestep t, conditioned on both the timestep and the external signal τ_θ.

5. Optimization: The model is trained to minimize the difference between predicted and actual noise across all timesteps.

Inference Phase

1. Noise Initialization: Begin with pure Gaussian noise $z_T \sim \mathcal{N}(0, \mathbf{I})$ in latent space.

2. Iterative Denoising: For $t = T$ down to 1, the conditioned U-Net progressively removes noise, guided by the conditioning signal τ_θ.

3. Decoding: The VAE decoder $\mathcal{D}$ maps the final denoised latent z_0 back to pixel space, producing the output image $\tilde{x} = \mathcal{D}(z_0)$.

The following subsections provide a detailed examination of the architectural components underlying this workflow.

Perceptual Compression and the VAE

In the LDM framework, the VAE plays a pivotal role. This module is typically a hybrid architecture: it employs the probabilistic framework of VAEs to manage the latent space through KL Regularization, while simultaneously incorporating a GAN objective to ensure photorealistic decoding.

The VAE serves as a spatial compression mechanism, enabling the architecture to perform the diffusion process efficiently in a reduced-dimensional space. It consists of an encoder $\mathcal{E}$ that maps images from pixel space to latent space and a decoder $\mathcal{D}$ in Figure 2.3) that reverses this transformation. During training, the encoder maps an image $x \in \mathbb{R}^{3\times H\times W}$ into a latent representation $z = \mathcal{E}(x)$, where $z \in \mathbb{R}^{c\times h\times w}$. Here, H and W denote the spatial dimensions of the image in the pixel space, while c represents the number of latent feature channels used to encode visual information. The relationship between the two spaces is defined by a relative downsampling factor $f = H/h = W/w$.

To overcome the characteristic blurring artifacts associated with traditional Variational Autoencoders, the VAE of the LDM architecture adopts a multi-objective loss function. Beyond standard pixel-wise reconstruction objectives, the model integrates a Learned Perceptual Image Patch Similarity (LPIPS) loss. This approach leverages a pretrained deep neural network to extract high-level features, ensuring that the reconstructed image maintains structural consistency with human visual perception.

By projecting the image into this lower-dimensional space, the VAE effectively performs perceptual compression, filtering out high-frequency details that are less perceptible to the human eye while preserving the semantic integrity of the visual data. Consequently,

the diffusion process, including the iterative denoising backbone, operates exclusively on the latent representation z, which serves as a compact and information-dense abstraction of the original signal.

Latent Denoising and U-Net Architecture

Upon completion of the denoising trajectory, the decoder $\mathcal{D}$ is employed to project the refined latent representation back into the pixel space, yielding the final reconstructed image $\hat{x} = \mathcal{D}(z)$. This strategy significantly mitigates the VRAM footprint and inference latency. Moreover, it allows the diffusion backbone to prioritize global structural coherence and high-level conceptual features rather than being computationally overwhelmed by low-level pixel noise.

Within the latent space, the iterative denoising process is performed by a U-Net backbone. Originally introduced by Ronneberger et al. [18] for biomedical image segmentation, the U-Net architecture was subsequently adapted as the primary noise estimation module for diffusion models by Ho et al. [16]. The primary objective of this convolutional neural network is to predict the noise component ϵ_θ that was added to the latent representation at a specific timestep t. The architecture incorporates Residual Network (ResNet) blocks, which utilize skip connections to preserve spatial information and ensure stable gradient flow throughout the diffusion steps.

These blocks are also conditioned on the timestep t via temporal embeddings. This conditioning allows the U-Net to dynamically adapt its behavior based on the current state of the denoising process; for instance, the strategy required to reconstruct structure from a high-variance Gaussian distribution in the initial phases differs significantly from the fine-grained detail refinement performed in the final stages. In the specific context of Latent Diffusion Models, this backbone is further enhanced with attention mechanisms to facilitate conditional generation [17].

Conditioning via Cross-Attention

To enable multi-modal conditioning, the U-Net architecture within the LDM framework is augmented with Transformer blocks interleaved between its ResNet-based convolutional layers. In this context, Transformer blocks are architectural modules that integrate self-attention and cross-attention layers to process long-range dependencies within the data. While convolutional layers excel at extracting local features, these blocks allow the model to establish global spatial coherence and, crucially, to align the internal visual representations with external conditioning signals, such as text embeddings.

The core of these blocks is the Cross-Attention mechanism, which allows the model to fuse the spatial information of the intermediate feature maps with the semantic information of the prompt. The system computes the alignment between each spatial region of the image (Q) and each token of the prompt (K) via a scaled dot-product. Formally, for query inputs $Q \in \mathbb{R}^{N_q \times d}$, key inputs $K \in \mathbb{R}^{N_k \times d}$ and value inputs $V \in \mathbb{R}^{N_k \times d_v}$, the cross-attention head computes:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d}}\right)V \tag{2.6}$$

The resulting attention map represents the semantic relevance between the visual and textual components: higher attention scores ensure that the model prioritizes specific tokens when synthesizing the corresponding regions of the image. This mechanism is

essential for the precise injection of stylistic and material information, ensuring that the generated textile patterns are not only visually coherent but also semantically aligned with the user's descriptive prompt.

Chapter 3

Classical and Learning-based methods for texture generation

This chapter analyzes the technical evolution of texture generation methodologies, categorized into classical model-based approaches and contemporary learning-based frameworks. Historically, the field has been defined by the distinction between procedural models and example-based synthesis. Procedural techniques utilize mathematical algorithms to generate resolution-independent imagery, frequently employing pseudo-random functions such as Perlin Noise to achieve visual realism with minimal storage requirements. Conversely, example-based methods, including pixel-based Markov Random Field models and patch-based Image Quilting, perform synthesis by iteratively sampling and rearranging elements from a source exemplar to preserve local patterns.

The implementation of Deep Learning transitioned the synthesis process from raw pixel manipulation toward the use of parametric neural descriptors. Techniques such as Neural Style Transfer and Texture Networks utilize the hierarchical feature space of CNNs to extract statistical correlations via Gram Matrices. Further developments in GANs, specifically Spatial GANs and SinGAN, introduced mechanisms for capturing spatial stationarity and internal patch recurrence across multiple scales.

The current state of the art is represented by LDMs, which facilitate a shift from statistical reconstruction to semantic synthesis. By operating within a compressed latent manifold and utilizing multimodal alignment through CLIP, these frameworks decouple semantic structure from spatial redundancy. This transition allows for texture generation guided by linguistic and visual priors, addressing the historical limitations of structural drift and lack of granular control in professional production workflows.

3.1 Classical methods for texture generation

Before the advent of Deep Learning, two groups of techniques constituted the state of the art for creating virtual worlds: procedural approaches, which rely on mathematical definitions and deterministic or stochastic algorithms and example-based approaches, which attempt to extend a source sample while preserving its visual characteristics.

3.1.1 Procedural Models

Procedural textures represent a class of synthetic imagery generated through mathematical algorithms rather than pre-recorded bitmap data. Unlike traditional image-based

mapping, which relies on the projection of predefined two-dimensional bitmap images (textures) onto a 3D surface through a system of UV coordinates, procedural models compute the appearance of a surface or volume on-the-fly during the rendering process. This approach offers significant advantages in terms of memory efficiency, as the texture is defined by a compact set of instructions and parameters rather than a fixed grid of pixels, making them particularly effective for generating textures that exhibit a high degree of regularity.

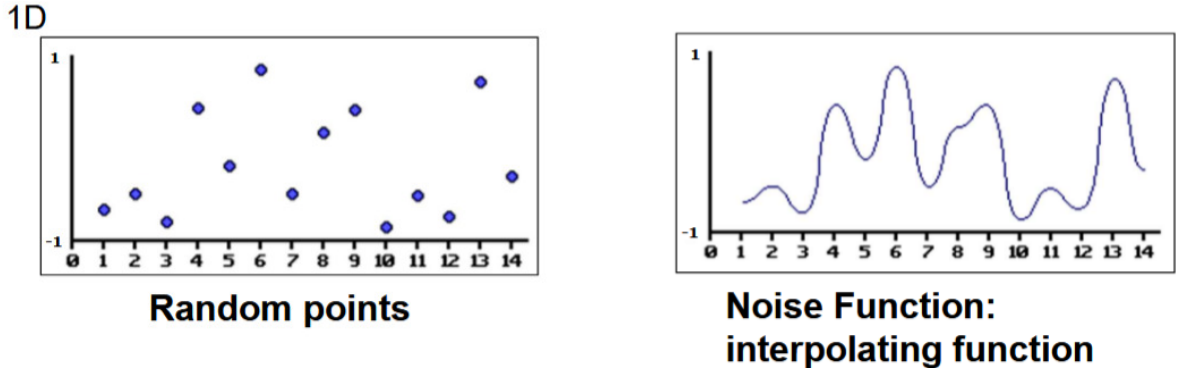
A fundamental characteristic of procedural texturing is its resolution independence. Since the texture is defined by a continuous mathematical function $f(p)$, it can be evaluated at any spatial frequency. This allows for virtually infinite detail, avoiding aliasing artifacts that typically occur when zooming into low-resolution bitmapped images. Furthermore, these methods are frequently implemented as solid textures, where the function is defined in three-dimensional space $f(x, y, z)$. This property allows objects to be treated as if they were carved out of a solid volume, effectively bypassing the complex issues related to UV parameterization and surface discontinuities.

The core of most procedural models lies in the integration of pseudo-random functions to introduce controlled irregularities. These functions are designed to break the artificial perfection of computer-generated surfaces by providing a source of coherent randomness. Consequently, procedural modeling remains a cornerstone of modern computer graphics for achieving visual realism with minimal storage overhead.

Perlin Noise

The most significant contribution in this area is Perlin Noise, developed by Ken Perlin in 1983 [19]. It is a pseudo-random function used to add controlled noise to surfaces, avoiding the artificial smoothness of synthetic geometry.

Mathematically, the function $Noise(x, y, z)$ returns a single pseudo-random number in the range $[-1, 1]$ for any given point in space. The algorithm creates a lattice (grid) with pseudo-random values assigned to each node. The value for any point inside the lattice cells is calculated through interpolation, typically linear or cubic (see Figure 3.1).



(a) Pseudo-random points generated on the integer coordinates of the grid.

(b) Continuous noise function obtained by interpolation of the points.

Figure 3.1: Visual representation of the Perlin Noise generation process. Source: [20]

The visual characteristics of the noise can be adjusted using three main parameters: amplitude, which controls the intensity of the effect; frequency, which determines the scale of the detail; and phase, which shifts the location of the noise peaks. To create complex,

natural-looking effects such as clouds or fire, multiple layers of noise, known as octaves, are summed together. In this multi-layered approach, each subsequent layer typically doubles the frequency and halves the amplitude of the previous one.

Perlin Noise is utilized in various applications, including bump mapping, where the noise's differential ($DNoise$) modifies surface normals to create the illusion of geometric detail without altering the underlying mesh. Additionally, 3D noise textures are essential for volumetric effects, such as rendering fog or smoke, by gathering opacity and lighting data within a bounding shape.

Despite their versatility, the primary limitation of these methods lies in their inability to generate contextual details. A noise function lacks the structural knowledge required to correctly place semantic features, such as the seams of a garment or specific wear patterns resulting from physical processes.

3.1.2 Example-Based Models

Example-based synthesis methods prioritize the replication of existing patterns over explicit mathematical modeling. Starting from a source sample, these algorithms synthesize a new texture while attempting to avoid visible repetitions and tiling effects. The process typically fills a target image by iteratively selecting elements that match the local context of the already synthesized areas. While this local focus allows for the reconstruction of complex patterns, it can lead to visible seams, especially when the source sample is small or contains non-stationary structures.

Two foundational approaches exemplify the evolution of example-based synthesis: pixel-based methods, which operate at the finest granularity by synthesizing individual pixels through Markov Random Field modeling and patch-based methods, which improve efficiency and structural coherence by resampling larger blocks with optimized boundary transitions. These two paradigms represent distinct trade-offs in the synthesis process and serve as canonical examples of how spatial resampling can be implemented at different scales.

Pixel-Based Synthesis

The pixel-based approach, established by Efros and Leung in 1999 [21], treats texture synthesis within the framework of a Markov Random Field (MRF) [22]. An MRF is a stochastic model used to describe spatial dependencies, founded on the principle that the probability distribution of a specific variable, such as a pixel, is conditioned exclusively upon the values of its immediate neighbors rather than the global set. This model operates on the assumption that texture is a local and stationary process, meaning that the visual characteristics of a small patch are sufficient to predict the rest of the pattern. Consequently, the algorithm synthesizes new textures by copying pixels from the source data, maintaining local statistical consistency.

The effectiveness of this method is primarily determined by the size of the square neighborhood window, which serves as the algorithm's only critical parameter. This window must be large enough to capture the characteristic scale of the largest structural element, or texel, within the sample. For structured textures, the window size is fundamental: it must encompass at least one full period of the pattern to maintain spatial consistency. If the neighborhood is too small, the algorithm fails to recognize the underlying geometry of the texture, resulting in a disorganized distribution of pixels rather than a coherent global pattern.

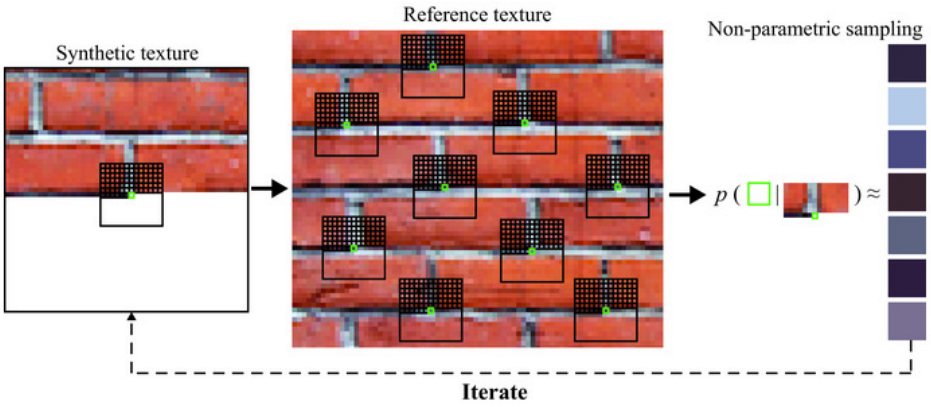


Figure 3.2: Overview of the pixel-based synthesis process. On the left, the algorithm searches the sample image for neighborhoods matching the context of the pixel to be synthesized. On the right, the stochastic selection process: a candidate is randomly chosen from a set of matches within an acceptable error threshold to prevent exact repetition and tiling artifacts. Source: [21]

To determine the color of a new pixel, as illustrated in Figure 3.2, the method analyzes its already synthesized local neighborhood and searches the original sample for all similar configurations. The similarity is measured using techniques like the Sum of Squared Differences (SSD) between the intensity of the pixels in the neighborhood and the candidate patches in the source. While computationally straightforward, reliance on raw pixel intensity makes the SSD metric inherently fragile to variations in lighting and contrast, a fundamental limitation that later motivated the shift toward illumination-invariant neural descriptors. Rather than choosing the single best match, the algorithm identifies all instances where the SSD is lower than a specific threshold and randomly selects one of these matching pixels to be copied into the target image. This stochastic selection is crucial for avoiding exact tiling and preserving the variations found in surfaces.

The synthesis is an iterative process that can follow different execution orders, such as a spiral growth from an initial seed. In cases where a pixel does not have sufficient neighboring context, as occurs at the very beginning of the process or along the boundaries, the algorithm can randomly sample values directly from the input texture to provide a starting point. However, since each new pixel selection depends on previously synthesized values, any approximation error in the matching process can propagate through the image. This leads to a phenomenon known as slippage or cumulative drift, where the lack of global constraints causes the structure of the texture to gradually dissolve into incoherent patterns over large scales.

For textile modeling, this limitation creates a clear distinction in performance based on the material's properties. The pixel-based approach is highly effective for stochastic or non-woven fabrics, such as felt or fleece, where the visual appearance is dominated by a random distribution of fibers. Conversely, for highly structured textiles like twill or satin, the lack of global constraints often causes the alignment of the threads to break or deviate, compromising the geometric integrity of the fabric's surface over large areas.

Patch-Based Synthesis

To address the limitations of pixel-by-pixel synthesis, Efros and Freeman introduced Image Quilting in 2001 [23]. This technique shifts the synthesis unit from individual pixels to entire blocks, or patches, extracted from the source sample. By operating on a macroscopic

scale, the algorithm is significantly more efficient and better equipped to preserve the structural integrity of the texture, which is often lost in pixel-based approaches due to cumulative matching errors.

The synthesis process relies on a constrained tiling strategy. Each new patch is selected based on its compatibility with previously placed blocks within an overlapping region. The algorithm searches the source sample for patches whose overlapping boundaries minimize the SSD with the current output. To maintain variety and avoid periodic repetitions, a stochastic selection is applied among the candidates that fall within a small error tolerance of the optimal match.

The defining innovation of this method is the minimum error boundary cut. Even with an optimal patch selection, a straight vertical or horizontal transition between blocks often results in visible seams and edge discontinuities. To mitigate this, the algorithm computes a non-linear path through the overlap region where the difference between the two overlapping patches is minimal, as illustrated in Figure 3.3. This path is determined using dynamic programming, which identifies the surface of least resistance between the two signal structures. By following the natural contours of the texture patterns, the cut effectively masks the transition, creating a seamless visual flow.

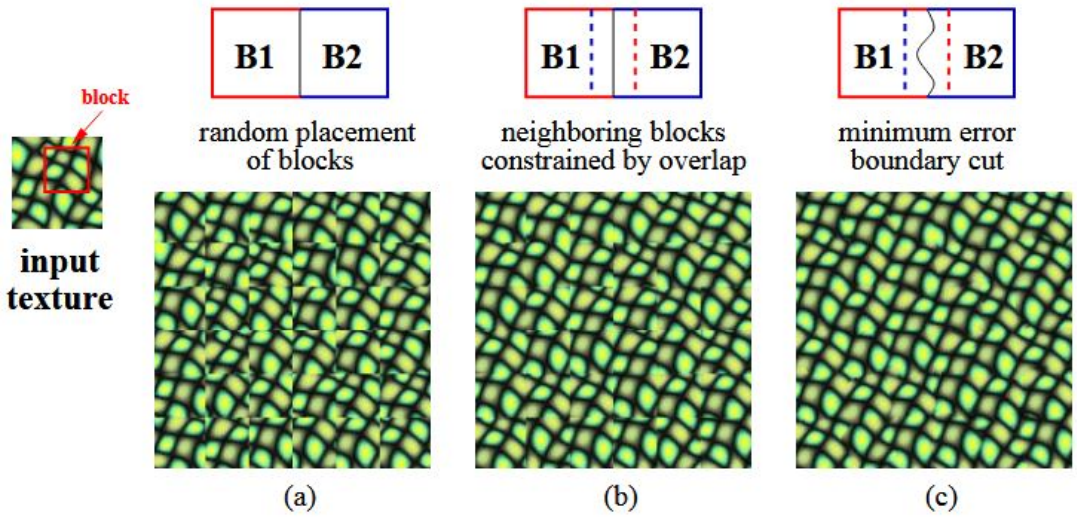


Figure 3.3: The minimum error boundary cut process. The algorithm identifies a non-linear path through the overlap region where the intensity difference between patches is minimal. This dynamic programming approach effectively masks the transition between blocks, preserving the structural continuity of the texture. Source: [23]

For the synthesis of textiles, Image Quilting represents a substantial improvement. The preservation of entire patches ensures that the geometric consistency of the weave remains intact, preventing the structural dissolution typical of pixel-based methods. In fabrics with complex interlacing, the minimum error boundary cut allows the algorithm to align the yarn structures across different blocks, maintaining the continuity of the threads. However, the method's effectiveness is closely tied to the block size parameter; a patch that is too small may fail to capture the global pattern, while one that is too large may lack the flexibility required to adapt to non-stationary textures.

3.2 Learning-based image generation

The transition from handcrafted statistical models to learning-based synthesis represents a fundamental shift in texture generation methodology. Rather than relying on explicitly defined feature extractors or direct spatial resampling, these approaches leverage the representational capacity of deep neural networks to learn hierarchical descriptors directly from data.

This paradigm offers critical advantages over classical methods. Learned features exhibit greater robustness to variations compared to pixel-based metrics, as CNNs implicitly normalize for changes in lighting and contrast. For textile synthesis specifically, their hierarchical architecture enables the disentanglement of material properties from imaging conditions, allowing networks trained on diverse samples to separate intrinsic fabric appearance from extrinsic factors. However, this shift introduces new technical challenges. The depth and complexity of modern CNNs create vast parameter spaces that must be carefully navigated, raising questions of optimization stability, computational efficiency and controllability. Different learning-based architectures represent distinct trade-offs between these factors: early methods required iterative optimization with significant latency, later feed-forward designs traded flexibility for speed and adversarial training brought visual fidelity at the cost of training instability.

The following subsections trace this evolution through three key developments. Neural Style Transfer first demonstrated how pre-trained classifiers could be repurposed as fixed feature extractors for iterative synthesis. Feed-forward architectures subsequently internalized these statistics during training to enable real-time generation. Adversarial frameworks then introduced discriminator-driven objectives to enforce perceptual realism, a mechanism that persists in modern latent diffusion models as a refinement tool for high-frequency detail recovery.

3.2.1 Neural Style Transfer

Theoretical Foundation

The introduction of Neural Style Transfer (NST) by Gatys et al.[24] represented a turning point by repurposing the hierarchical feature extraction capabilities of CNNs for generative objectives. In the field of texture synthesis, NST demonstrated that a pre-trained CNN, specifically the VGG-19 architecture [25], could serve as a parametric model for visual patterns. While Section 2.0.1 defines feature maps as spatial indicators of detected patterns, NST reinterprets these activations as statistical signatures. By shifting the optimization target from pixel intensities to high-level statistical descriptors within the learned feature space of the CNN, NST changed the role of the network from a discriminative classifier to a fixed descriptor used to guide the synthesis of textures.

This approach is rooted in the Stationary Hypothesis, which posits that a texture is defined by spatial statistics that remain invariant to translation across the image plane. In the NST framework, this is mathematically encapsulated by the Gram Matrix G^L computed from the CNN activations. As illustrated in the analysis phase of Figure 3.4 (left), for a given convolutional layer L with N_L feature maps of vectorized size M_L, the Gram Matrix entry G^L_{ij} is defined as the inner product between the vectorized feature maps F^L_i and F^L_j:

$$G_{ij}^{L} = \sum_{k} F_{ik}^{L} F_{jk}^{L} \tag{3.1}$$

By computing the correlation between different kernels (indexed by i and j) across all spatial locations k, this operation effectively marginalizes the spatial coordinates. This process facilitates the extraction of a global descriptor that, by abstracting away the localization of features, preserves exclusively their statistical distribution.

Capturing the complexity of a texture requires matching these CNN statistics across a diverse set of network depths rather than relying on a single layer. This multi-scale reconstruction process utilizes the shallow layers of the CNN to capture local features, such as pixel-wise color distributions and high-frequency edges, while the deeper layers extract the broader macro-structure. The simultaneous optimization of these levels of abstraction ensures that the synthesized output maintains consistency across multiple scales.

However, the spatial marginalization inherent in the Gram Matrix introduces a limitation for anisotropic materials. Because global orientation information is discarded by the CNN feature correlations, standard NST can struggle to maintain the strict directionality required for certain weave patterns. To address this, extensions of this method incorporate rotated versions of the source exemplar during the loss calculation, enabling the CNN to recognize and preserve the directional motif.

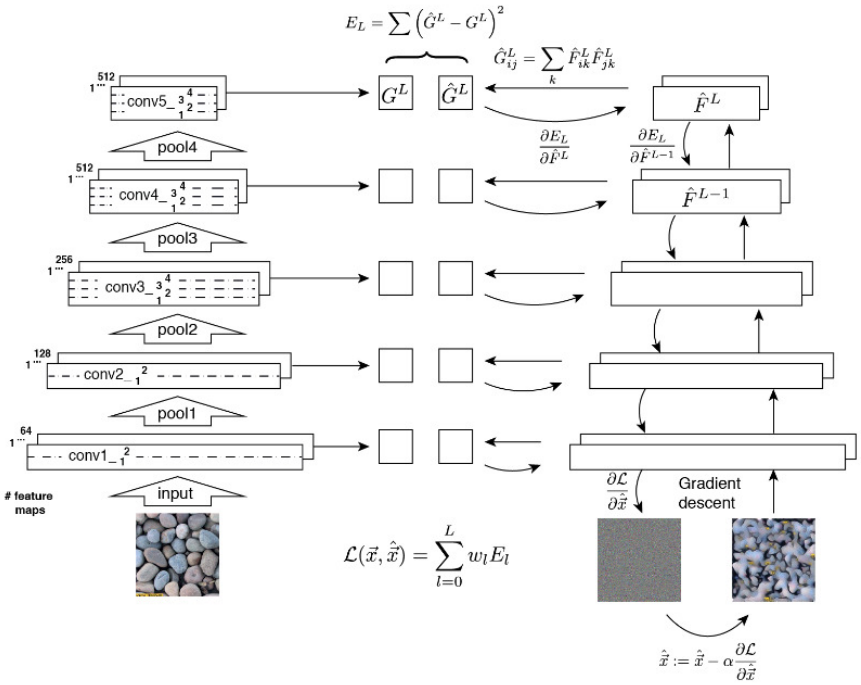


Figure 3.4: Schematic overview of the Neural Style Transfer framework for texture synthesis. The analysis phase (left) extracts statistical descriptors by computing Gram matrices from the feature responses of a pre-trained CNN. The synthesis phase (right) utilizes an iterative optimization process: starting from a white noise initialization, the image is progressively updated via gradient descent to minimize the loss between its internal activation correlations and the reference Gram matrices. Source: [24].

Optimization Mechanics

The synthesis mechanics of NST differ fundamentally from the feed-forward generative models. Instead of a single-pass generation, it utilizes an iterative inversion process, depicted in Figure 3.4 (right), where an image initialized as white noise is transformed through backpropagation. Crucially, unlike the training phase of a CNN where weights are updated to minimize classification error, here the weights of the VGG-19 network remain frozen. The image itself, initialized as white noise, becomes the set of learnable parameters. The algorithm calculates the gradient of the style loss with respect to the input pixels, typically employing a second-order optimizer such as L-BFGS [26] to progressively refine the noise until its internal correlations align with the reference Gram matrices.

While the NST framework provides a robust model for feature representation, its integration into professional production pipelines is limited by several technical constraints. The iterative nature of the process results in significant computational latency, which is unsuitable for real-time workflows. Furthermore, the original formulation lacks native support for periodic boundary conditions, complicating the generation of the tileable surfaces discussed in Section 1.6. To address this, spatial continuity can be enforced by replacing standard zero-padding with circular padding within the convolutional layers. By mapping the boundaries of the feature maps to their opposite edges during the convolution operation, the network effectively treats the image as a toroidal surface, ensuring that the statistical correlations remain consistent across the seams.

The stochastic instability caused by the random noise initialization also limits fine-grained control over the output. To address issues such as chromatic saturation, the optimization can be stabilized using a histogram loss [27], ensuring the pixel value distribution and dynamic range remain aligned with the original sample. Beyond efficiency, the evolution of the state of the art has also addressed the limitations of generic descriptors. While early NST relied on architectures such as VGG-19 pre-trained for object recognition, contemporary frameworks have shifted toward using descriptors extracted from discriminators trained directly within the material domain. This transition, as seen in deep geometric texture synthesis, allows the model to capture features specifically relevant to the geometric intricacies and physical properties of the texture, rather than patterns optimized for general-purpose classification.

3.2.2 Texture Networks

To overcome the inherent inefficiencies of iterative synthesis, the field transitioned toward feed-forward architectures, redefining the relationship between network weights and texture statistics. An important contribution to this shift was the introduction of Texture Networks by Ulyanov et al. [28], which decoupled the optimization process from the final synthesis. By shifting the computational burden to an offline training phase, this approach replaces the per-pixel iterative updates with a generative model trained to map stochastic noise vectors directly into the texture domain.

During this training phase, depicted in Figure 3.5, the generator is guided by a frozen descriptor network (typically VGG-19) that calculates the style loss. Crucially, the gradient descent no longer modifies the image pixels, but instead updates the generator's weights to internalize the target statistical distribution. Consequently, once optimized, the model can produce high-fidelity textures in a single forward pass. This inversion of the generative pipeline achieves execution speeds several orders of magnitude faster than

earlier methods, effectively enabling the integration of neural synthesis into real-time professional workflows.

The efficacy of these architectures stems from the multi-scale, fully convolutional design seen in the generator network of Figure 3.5. By injecting noise tensors Z at different resolutions ($3 \times 16 \times 16$, $3 \times 32 \times 32$, etc.) and joining these streams through upsampling and concatenation, the network can capture features at various spatial frequencies simultaneously. Furthermore, because the architecture is entirely convolutional, it is not constrained by a fixed output resolution; a single trained model can generate textures of arbitrary dimensions by adjusting the spatial extent of the input noise tensor, a feature of significant value for the production of expansive virtual surfaces.

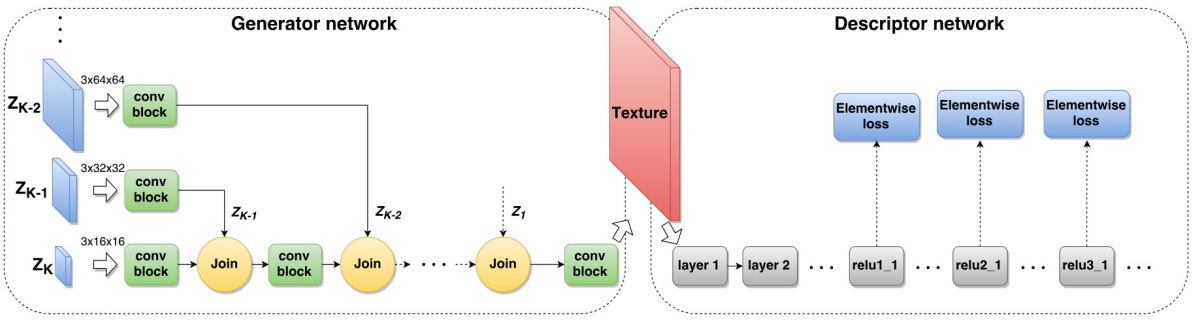


Figure 3.5: Architecture of a Texture Network as proposed by Ulyanov et al. The generator network (left) employs a multi-scale approach where noise tensors at different resolutions are processed through convolutional blocks and joined to synthesize the texture. The descriptor network (right) is utilized during training to compute the perceptual loss across multiple layers. Adapted from [28].

A critical advancement in this context was the introduction of Instance Normalization [29]. Unlike Batch Normalization, which can homogenize stylistic details by calculating statistics across an entire dataset, Instance Normalization allows the network to isolate and preserve the specific statistical signature of an individual texture. This architectural modification ensures that feed-forward models maintain the visual fidelity of optimization-based methods while meeting the performance requirements of real-time asset production. However, it should be noted that these early feed-forward models remained specialized: a single network had to be trained for each specific material, a limitation that would later be addressed by the more flexible adversarial and latent generative frameworks.

3.2.3 Adversarial Frameworks

Building upon the adversarial framework established in Section 2.3, the application of GANs to texture synthesis has focused on capturing spatial stationarity and structural regularity. A fundamental contribution is represented by Spatial GANs (SGAN) [30], which expand the latent space from a single vector to a spatial noise tensor. This fully convolutional design allows for the synthesis of images with arbitrary dimensions, where each output region is locally determined by a corresponding portion of the input tensor, effectively operationalizing the stationarity bias for expansive surfaces.

For textiles and regular patterns, the Periodic Spatial GAN (PSGAN) [31] introduced a crucial refinement by integrating periodic dimensions into the latent space. This allows the model to explicitly characterize the cyclicity of motifs such as honeycombs or woven structures, ensuring the generation of perfectly tileable surfaces.

Another significant milestone is SinGAN [32], which learns the internal patch distribution of a single reference image. As illustrated in Figure 3.6, this is achieved through a hierarchical pyramid of generators ($G_N \dots G_0$) and discriminators ($D_N \dots D_0$). The training progression moves from the coarsest scale at the bottom to the finest at the top; at each level, the generator G_n learns to synthesize a fake image $\tilde{x}_n$ by receiving both a noise map z_n and the upsampled output from the previous scale.

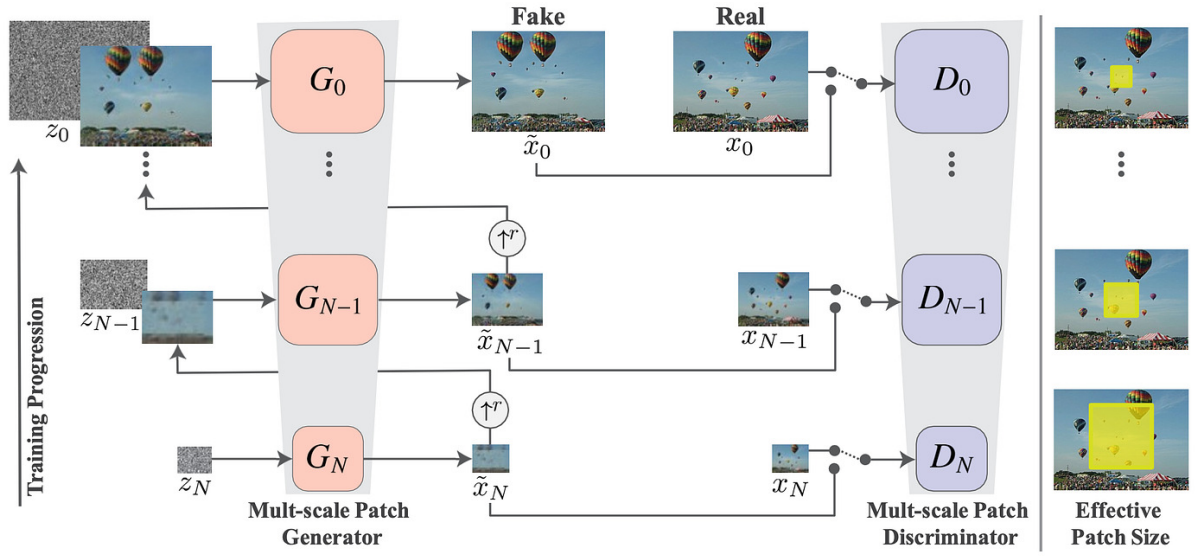


Figure 3.6: The SinGAN multi-scale pipeline. The architecture consists of a hierarchy of patch-based generators and discriminators that learn the internal statistics of a single image. The effective patch size (right) illustrates how the network captures larger structural dependencies at coarser scales before refining micro-details at the finest resolutions. Source: [32].

This multi-scale approach allows the system to exploit the patch recurrence typical of natural textures. As shown on the right side of Figure 3.6, the effective patch size varies relative to the image resolution at each scale: at coarser levels, the network captures the global arrangement and macro-structure, while at finer levels, it focuses on the synthesis of high-frequency details. This mechanism enables SinGAN to generate realistic variations without requiring massive training datasets, making it an ideal tool for texture expansion and the creation of diverse visual assets from a single tile.

The extension of neural synthesis to the third dimension has addressed the limitations of 2D projections on complex surfaces. However, adapting convolutional architectures designed for regular, Euclidean pixel grids to the irregular, non-Euclidean topology of 3D meshes presents a significant technical challenge. Deep Geometric Texture Synthesis [33] represents the state of the art in this domain, enabling the generation of geometric details directly on 3D meshes. Rather than producing standard displacement maps in UV space, this framework operates on the non-Euclidean geometry of the mesh, learning local statistics from neighborhoods of faces and vertices.

This model can deform vertices in any direction, both normal and tangential, allowing for the synthesis of complex structures like scales or thick textile fibers that maintain structural integrity regardless of mesh resolution. This approach is genus-oblivious, facilitating the transfer of textures between objects with different topologies without requiring consistent UV parameterization, thereby resolving a major bottleneck in 3D asset production.

In modern generative frameworks such as Stable Diffusion, a core framework of this work detailed in Section 4.1.1, the adversarial mechanism remains a core component, though its role has shifted from primary generator to a refinement tool within the VAE stage. While the diffusion process handles the global structure of the texture in latent space, the final translation to the pixel domain relies on an adversarial loss to recover high-frequency details.

As discussed in Section 2.2, VAEs are prone to producing blurry reconstructions due to the nature of pixel-wise loss functions. To mitigate this, architectures like the one utilized in LDMs incorporate a discriminator during the training of the decoder. This adversarial component ensures that the synthesized textile fibers remain sharp and perceptually consistent, preventing the regression to the mean that would otherwise degrade the photorealism of the final output.

However, despite these advancements, GAN-based synthesis and refinement face critical limitations in professional environments. The first is large-scale repetitiveness; while models excel at local variation, the limited receptive field of convolutional kernels often prevents them from capturing long-range dependencies, leading to visible tiling on expansive surfaces. Furthermore, despite their success in enforcing global structural constraints like periodicity, GANs often function as a black box regarding local semantic manipulation, offering minimal granular control. In professional workflows, artists require the ability to precisely place specific features like localized wear or material defects, a level of spatial intent that is difficult to achieve within an unconditioned latent space. The inherent lack of interpretability in the latent space, coupled with the training instabilities discussed in Section 2.3, has motivated the search for more controllable and stable generative paradigms, such as the latent diffusion models.

3.3 The Shift to Latent Generative AI

The integration of diffusion models with multimodal systems marks a transition from synthesis based on pixel regularities to generation guided by semantic intent. Modern latent frameworks can represent physical material properties described through natural language, moving texture generation from statistical reconstruction to conceptual synthesis, where the output is steered by linguistic and visual priors.

This shift to the latent domain is necessitated by the requirement to decouple the semantic structure of a texture from its spatial redundancies. While classical methodologies operate directly on the raw pixel grid, they encounter significant computational overhead due to the inherent dimensionality of image data. In contrast, Latent Diffusion Models (LDMs) perform the generative process within a compressed manifold, providing a more efficient environment for synthesis. By utilizing a Variational Autoencoder (VAE), the architecture filters out spatial redundancies while preserving data integrity. This abstraction allows the denoising backbone to prioritize global structural coherence over stochastic pixel noise.

Furthermore, the latent space offers a theoretical advantage over the discrete sampling of patch-based methods. While classical patching is restricted by a limited receptive field, the compressed nature of the latent grid allows convolutional kernels to perceive a larger portion of the global context. This enables the synthesis of non-repeating variations, where the model can generate novel fiber configurations and structural transitions consistent with the underlying data manifold.

3.3.1 Multimodal Alignment

The capacity to control the generative process through semantic descriptors is fundamental to the practical deployment of latent diffusion models. Unlike unconditional synthesis, which relies solely on learned distributional priors, multimodal conditioning enables the model to navigate the latent manifold toward specific targets defined through alternative modalities, most commonly, natural language. This paradigm shift transforms the generative process from stochastic exploration into directed synthesis, where the output is actively steered by external semantic signals.

The conditioning mechanism operates by injecting information from a separate modality into the denoising backbone at strategic points in the architecture. In practice, this is typically achieved through cross-attention layers, where the conditioning signal acts as a contextual key-value store that modulates the self attention operations of the U-Net. The model learns to attend selectively to regions of the conditioning embedding that are semantically relevant to the content being synthesized at each spatial location and denoising timestep. This allows the network to incorporate high-level conceptual guidance while maintaining the flexibility to resolve fine-grained details autonomously.

The advantages of multimodal conditioning extend beyond controllability. By anchoring the generative trajectory to an explicit semantic representation, the model gains access to a richer space of visual priors. Rather than relying exclusively on distributional statistics learned from pixel co-occurrences, the system can leverage conceptual relationships encoded in the conditioning modality. This is particularly valuable for texture synthesis, where material properties are more naturally expressed through linguistic descriptors than through raw pixel patterns.

Contrastive Language-Image Pre-training

A prominent instantiation of this multimodal framework is the CLIP text encoder [34]. CLIP aligns visual and linguistic representations in a shared latent space through a contrastive objective, enabling zero-shot transfer across modalities. In the context of LDMs, the encoder maps prompt tokens into continuous embeddings, which are injected into the U-Net via the cross-attention mechanism described above. This ensures that the denoising process is steered toward generating content contextually aligned with the textual description.

By acting as a conceptual translator, CLIP enables the system to retrieve associated visual priors to guide the denoising trajectory. This ensures that synthesis becomes a directed execution of intent, where the model incorporates the physical nature of the material rather than being limited to superficial color arrangements. This alignment provides the predictability and semantic consistency required for professional asset production. Without this semantic anchor, the process would risk domain mismatch, leading synthesized regions to diverge from the original material's perceptual and structural coherence.

The efficacy of this framework is rooted in its capacity for zero-shot generalization, enabling the generation to transcend specialized datasets. The model interprets linguistic cues to represent the geometric nuances of a specific textile, allowing for the simulation of the material hand, which encompasses the tactile and structural qualities suggested by a fabric's surface appearance. Consequently, synthesis is no longer restricted to the replication of known samples but becomes a negotiation between human intent and a learned visual-semantic prior.

3.3.2 Generative Expansion

In the current paradigm, inpainting and outpainting have evolved from restorative utilities into methodologies for asset creation, shifting from local texture extension to global semantic extrapolation. While traditional example-based synthesis relies on rearranging existing pixels, a process constrained by the source sample's statistical limits, latent outpainting leverages the model's prior knowledge to interpret a material's underlying logic. This allows for the synthesis of unique manifestations of a material identity, such as realistic fluctuations in fiber distribution or the progression of localized weathering.

This capability addresses the historical trade-off between structural fidelity and visual variety. In classical frameworks, the limited receptive field often results in the wallpaper effect, where the inability to perceive long-range dependencies leads to visible tiling artifacts. Conversely, by operating within the compressed latent manifold, the diffusion process maintains a broader conceptual overhead, ensuring synthesized regions evolve logically across the canvas.

Furthermore, the transition from stochastic noise to informed latent initialization ensures that expansions remain anchored to the original textile structure, preventing the structural drift seen in iterative pixel-based methods. This workflow transforms texture generation into a contextualized act of creation, providing a solution to procedural monotony and conferring a sense of realism impossible to achieve through standard mathematical noise.

Despite the expressive power of latent diffusion, standard architectures cannot natively generate perfectly repeatable textures. This is a structural consequence of the convolutional kernels, which treat image edges as absolute boundaries due to the zero-padding applied during training. In a standard pass, zero-padding introduces implicit spatial anchoring, effectively creating a boundary bias that prevents the visual signal from flowing across margins. To achieve the continuity required for professional production, it is necessary to implement a toroidal topology within the latent space, transforming the grid into a continuous, self-referential manifold.

This shift requires a re-evaluation of how the model perceives spatial coordinates. Without forcing a periodic state, the process cannot guarantee that complex structures, such as interlacing yarn or geometric motifs, maintain their integrity during tiling. A theoretical hurdle is the statistical discrepancy that arises when the model's learned expectations, conditioned on edges, are violated by forced periodicity. Because internal activation statistics are optimized for bounded representations, a naive imposition of circularity can lead to a loss of structural definition or chromatic inconsistencies.

The resolution of this challenge involves a balancing of the model's learned distribution and a seamless toroidal flow. By treating the latent field as a closed manifold, the synthesis ensures every region is processed with consistent context. This requirement for global spatial synchronization establishes the technical foundation for a textile generation toolkit capable of eliminating the perceptual grids typical of tiled digital assets.

Chapter 4

A diffusion-based model for textile generation

4.1 The toolkit

The framework proposed in this thesis draws inspiration from the work of Riso et al. [35] in the field of structured pattern expansion via diffusion models. The authors address the synthesis of stationary patterns characterized by sharp geometries and flat colors, highlighting how general-purpose architectures, such as Stable Diffusion, struggle to preserve edge precision and structural regularity without specific adaptation. To overcome these limitations, Riso et al. propose a workflow based on the integration of an outpainting pipeline and the use of Low-Rank Adaptation (LoRA) [36]. The necessity for LoRA fine-tuning arises from the domain gap between natural images and the graphic style of hand-drawn patterns: by injecting low-rank matrices into the transformer blocks of the U-Net, the authors are able to instill the model with the geometric rules of a custom procedural dataset without losing the network's original semantic knowledge.

In the domain of photorealistic textile generation addressed in this thesis, however, specific fine-tuning for every fiber or weave type is not strictly required to achieve high-quality results. Building upon the methodological foundations established by Riso et al. regarding the spatial management of latents, this work recontextualizes their workflow for the textile industry. Specifically, while the framework was adapted to accommodate the photorealistic requirements of regular textile geometries and it has been further extended to encompass textures with irregular patterns. By doing so, this methodology enables the generation of seamless and coherent results, leveraging outpainting techniques to handle stochastic variations in the textile surface.

4.1.1 Stable Diffusion

This work leverages Stable Diffusion, an implementation of the LDM framework discussed in Section 2.4. It is a text-to-image diffusion model conditioned on the text embeddings provided by a Contrastive Language-Image Pre-training (CLIP) text encoder. Specifically, this work employs Stable Diffusion v1.5, which includes native inpainting capabilities guided by binary masks. In the proposed workflow, the input texture is positioned at the center of the canvas and the model's inpainting functionality is utilized to coherently synthesize the surrounding masked regions.

The model was originally trained on the LAION-5B dataset, a large-scale open-source

corpus containing over five billion image-text pairs scraped from the web. Its architecture consists of three primary modules coordinated by a scheduler, which manages the denoising trajectory across the inference steps: a Variational Autoencoder (VAE), a U-Net backbone and a CLIP encoder, whose functions were already discussed in Section 3.3.1.

Specifically, the model employs a VAE architecture with a downsampling factor of $f = 8$ and $c = 4$ latent channels, producing 64×64 latents from the standard 512×512 input resolution. In this work, the Fine-Tuned Mean Squared Error (*ft-mse*) version of the VAE is employed. This specific module, released by Stability AI [37], is an improved version of the original KL-regularized autoencoder. It is chosen for its superior ability to reduce compression artifacts and maintain the fidelity of high-frequency details, a requirement that is essential for seamless and high-quality texture synthesis.

Unlike standard text-to-image models that operate on a 4-channel latent input, this specific configuration utilizes an expanded architectural design to handle the complexities of image restoration and expansion. Following the concatenation-based conditioning approach introduced by Rombach et al. [17], the U-Net's input layer is expanded to nine channels. This modification allows the model to process a concatenated tensor composed of three distinct elements: the 4-channel noisy latent representation (z_t), the 4-channel latent representation of the masked image context (z_{masked}) and a single-channel binary mask (m).

While the 9-channel input provides the necessary spatial constraints by making the model aware of the unmasked surrounding context, the Cross-Attention mechanism continues to act as the primary bridge for semantic guidance. The CLIP text encoder provides the K and V matrices that instruct the U-Net on how to fill the masked regions. This dual-conditioning strategy ensures that the generated inpainting results are not only spatially consistent with the original texture but also semantically aligned with the user's stylistic requirements, effectively bridging the gap between local continuity and global conceptual coherence.

4.1.2 IP-Adapter

Stable Diffusion relies on the CLIP text encoder, but this approach is constrained by the prompt engineering bottleneck. Users must articulate visual concepts through a linguistic medium lacking granularity. In textile synthesis, describing weave frequency, fiber distribution, or wear through text is inefficient and prone to interpretative variance.

To address these limitations, this work integrates the IP-Adapter architecture [38]. Unlike methods requiring fine-tuning, the IP-Adapter preserves the frozen weights of the U-Net while introducing a parallel pathway for visual information. This modularity ensures that generative capabilities remain intact while guided by a reference image.

The innovation lies in the decoupled cross-attention strategy, which modifies the mechanism described in Section 2.4. In standard diffusion, external features are integrated through a shared attention layer, but concatenation often leads to modal imbalance. The IP-Adapter circumvents this by adding separate cross-attention layers for the image prompt.

Formally, the IP-Adapter introduces projection matrices for image keys (K') and values (V'). While queries (Q) are derived from the U-Net feature maps, the textual and visual attention are computed independently. The final output is obtained by merging the contributions:

$$\text{Output} = \text{softmax}\left(\frac{QK^T}{\sqrt{d}}\right)V + \lambda \cdot \text{softmax}\left(\frac{Q(K')^T}{\sqrt{d}}\right)V' \tag{4.1}$$

The first term represents textual conditioning, while the second accounts for visual guidance. The factor λ controls the influence of the image prompt. This separation allows the model to learn textile characteristics without interfering with the prompt, ensuring a synthesis that respects both modalities.

Visual information is processed through a pipeline designed to translate pixels into a format compatible with the attention blocks. A CLIP Vision Model extracts features from the reference image, capturing attributes such as color palette, density and patterns.

Since the dimensionality of CLIP Vision embeddings differs from the U-Net hidden dimensions, a projection module is employed. This module, consisting of a linear layer and layer normalization, maps features into the decoupled cross-attention space. This design prevents modal interference, allowing the system to leverage the guidance of the source textile while maintaining flexibility through the text encoder.

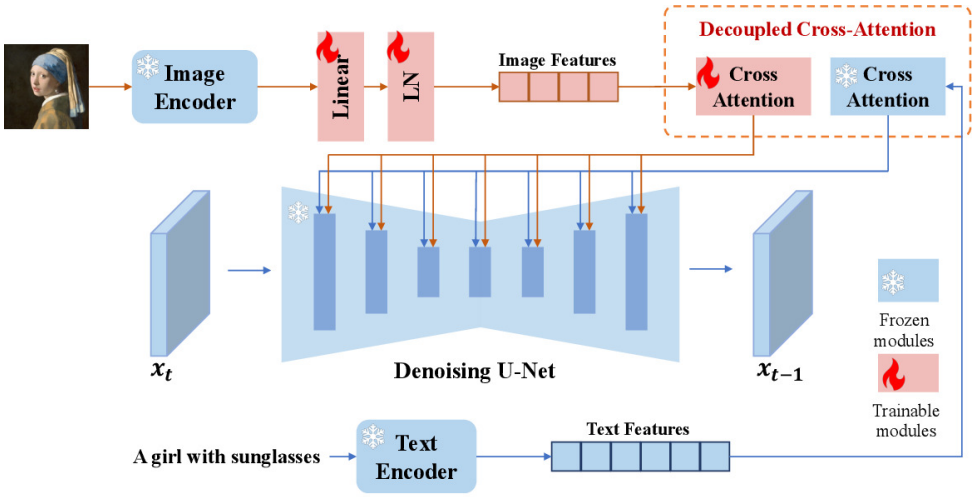


Figure 4.1: IP-Adapter architecture and decoupled cross-attention mechanism. Source: [38]

4.1.3 Noise Rolling

Achieving seamless tiling in diffusion-based generation necessitates the ability to repeat an image pattern without visible discontinuities at the boundaries. Standard Stable Diffusion models typically exhibit limitations in this task because the convolutional kernels within the U-Net backbone treat image edges as absolute boundaries. This is primarily due to the zero-padding applied during the inference phase, which prevents the kernels from perceiving the periodic nature of a tiled signal. To overcome this, this work adopts a technique known as noise rolling [17].

Noise rolling consists of applying a cyclic spatial shift to the latent tensor z_t before each denoising step. Formally, considering the latent tensor as a spatial function $z(h, w)$, the rolling operation applies a cyclic translation $(\Delta h, \Delta w)$ defined as:

$$z_{rolled}(h, w) = z((h + \Delta h) \pmod H, (w + \Delta w) \pmod W) \tag{4.2}$$

After the U-Net predicts the noise for the shifted latent, the resulting noise map is "unrolled" by applying the inverse translation before proceeding to the next inference step.

This mechanism ensures that every region of the latent space is processed as a central area rather than a boundary at different stages of the diffusion process, effectively neutralizing edge artifacts. In this implementation, the rolling is disabled after 90% of the inference steps. This allows the model to refine fine-grained details without the influence of the cyclic shift, which can otherwise introduce minor blurring during the final convergence.

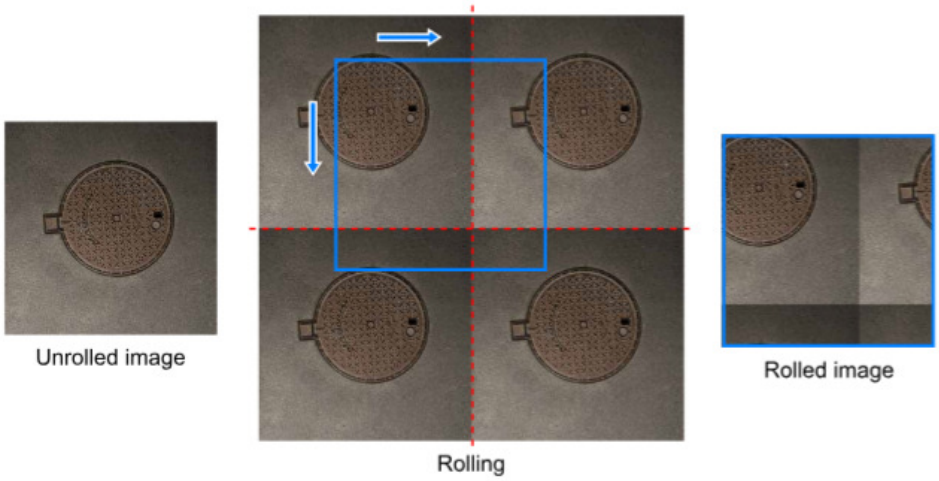


Figure 4.2: Mechanism of noise rolling for seamless latent generation. Source: [39]

However, noise rolling alone is often insufficient to guarantee perfect tileability. Even with a shifted latent, a convolutional filter processing a pixel at the boundary may still lack access to the adjacent information from the opposite side of the tile. To address this structural limitation, the U-Net architecture can be modified to incorporate circular padding instead of standard zero-padding. Circular padding allows kernels to wrap around the edges, effectively treating the latent grid as a continuous torus.

Despite its theoretical benefits, prevalent methodologies that advocate for global circular padding often result in a significant degradation of generative quality, as reflected by poorer CLIP scores. This occurs because the model was originally trained with zero-padding; forcing circular padding throughout the entire process shifts the internal activation statistics away from the learned distribution.

To mitigate this performance drop while preserving tileability, this work adopts a hybrid strategy: noise rolling is utilized to establish global coherence, while circular padding is activated exclusively after around 40-45% of the inference process. This optimization ensures that the generated textures are seamlessly tileable without compromising the high-fidelity details and semantic alignment of the final output.

4.1.4 Latent Replication

A central challenge in high resolution texture synthesis is the transition from a standard inference resolution to an expanded canvas. In the latent space, this corresponds to upscaling the tensor from 64×64 to 128×128 dimensions. To facilitate this transition, this work employs a technique known as Latent Replication [40].

Instead of initializing the newly expanded regions with pure Gaussian noise,which would require the model to generate content from scratch without a structural reference, the system utilizes the current state of the latents to perform an "informed initialization."

This process involves replicating the denoised latent tensor into a 2×2 grid to fill the expanded 128×128 area.

However, a naive replication would result in a repetitive grid pattern with visible discontinuities at the quadrant boundaries. To mitigate this, the toolkit implements a circular shift (rolling) during the replication phase. Specifically:

- The original 64×64 latent remains anchored at the center or is distributed across the quadrants.
- Each replicated tile is subjected to a spatial offset of half its dimension (32 pixels).

Upon completing 60% of the total inference steps, the spatial expansion is triggered. At this stage, Latent Replication provides the U-Net with a structured starting point for the remaining 40% of the denoising process. During these final steps, the model does not need to redefine the core structural features of the texture. Instead, it concentrates on blending the boundaries of the replicated quadrants, effectively transforming four individual blocks into a single seamless textile pattern.

Furthermore, this strategy significantly mitigates the risk of computational drift, a phenomenon in which the generative process gradually diverges from the original stylistic and structural parameters when expanding into new regions. By utilizing existing latent information rather than stochastic noise for the initialization of the expanded canvas, the system ensures that the outpainted areas remain strictly anchored to the original structure of the source textile, preventing the emergence of inconsistent patterns in the final high resolution output.

4.2 Schema and workflow

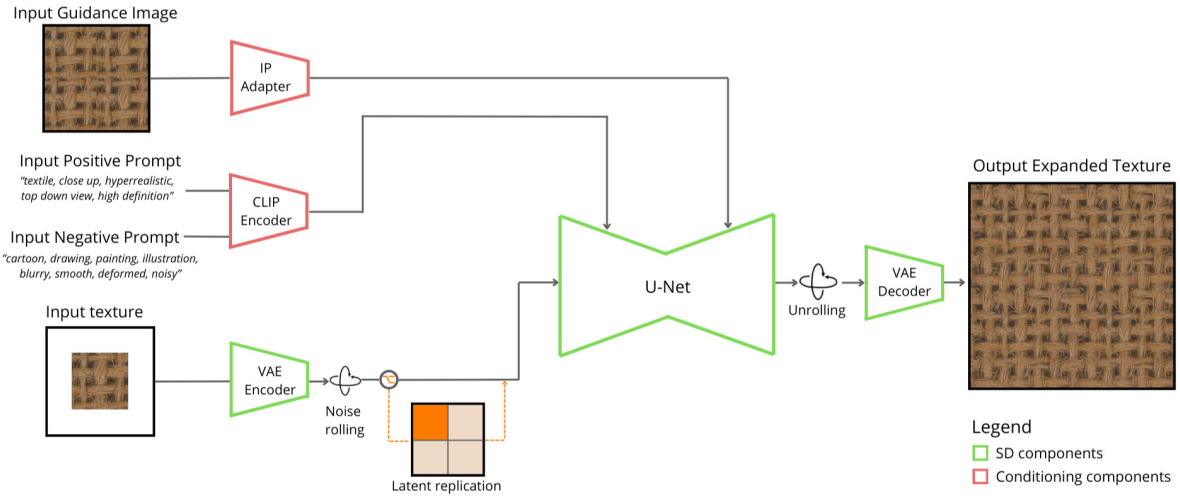


Figure 4.3: Project schema. Adapted from [35]

The integration of the previously described components results in a pipeline designed for iterative textile expansion. The operational logic is summarized in the following sections and illustrated in Figure 4.3.

The operational pipeline coordinates the interaction between the IP-Adapter, the inpainting U-Net and the custom callback function. The process follows a synchronized sequence of operations:

1. Input Preparation: The system receives the base 256×256 texture on a white 512×512 canvas (the Input Texture), the textual prompts (Positive and Negative) for the CLIP Encoder and the Guidance Image for the IP-Adapter (as shown on the left side of Figure 4.3).

2. Latent Encoding: The VAE encoder maps the Input Texture into the latent space.

3. Multimodal Conditioning: The CLIP Text Encoder and the IP-Adapter's Vision Encoder generate dual conditioning signals from the prompts and the Guidance Image to guide the U-Net.

4. Diffusion Loop and Callback Control: The U-Net performs the iterative denoising process. During this phase, a custom callback function (described in Section 4.1.3) intervenes at each step to apply the Noise Rolling and handle the Latent Replication once the 60% threshold is reached.

5. Final Reconstruction: Once the denoising process is complete, the VAE decoder projects the refined latents back into the pixel space to produce the final 1024×1024 expanded output.

4.2.1 Multimodal conditioning

The core of the generative workflow lies in the simultaneous processing of multimodal inputs. While the previous section detailed the IP-Adapter's architecture, its operational role within the workflow is to guide the U-Net alongside the textual prompt.

During the iterative denoising process, the U-Net operates as a synthesis engine where two conditioning streams converge:

- Semantic Stream (Text): The CLIP text embeddings provide high-level instructions regarding the material and color.

- Structural Stream (Image): The IP-Adapter provides low-level topological features extracted from the reference image, such as the specific weave density, thread orientation and micro-shadows of the fabric.

The cooperation between these two streams is particularly crucial during the first half of the inference steps. During this phase, the IP-Adapter ensures that the generated content in the masked regions is structurally indistinguishable from the original unmasked texture.

4.2.2 Inference and Custom Callback

The core of the texture generation process is governed by a custom callback function that executes after each denoising step. This function ensures spatial coherence and facilitates seamless texture expansion by managing three primary operations: latent replication, rolling and unrolling.

At the beginning of each callback cycle, the system performs an unrolling operation to return the tensors to their original coordinate frame if a spatial shift was applied in the previous iteration. The subsequent shift, defined by $(\Delta h, \Delta w)$, follows a scheduled distribution:

- Early stages: The shift amplitude is large to establish global structural coherence across the texture.
- Mid stages: The amplitude is gradually reduced to perform fine-scale calibrations.
- Final stages: The shift is set to zero to refine high-resolution details without spatial distortion.

4.2.3 Spatial Expansion and Latent Replication

A pivotal moment in the workflow occurs upon reaching 60% of the total inference steps. At this stage, the system executes a spatial expansion, quadrupling the latent tensor area (scaling from 64×64 grid to 128×128 is equivalent to a jump from a 512×512 pixel equivalent to 1024×1024 pixels). This allows the model to first establish the primary characteristics of the texture in a lower-dimensional space before committing to the full resolution.

To populate the expanded area, a Latent Replication strategy is implemented. Instead of a simple grid duplication, which would create visible seams, the replication is achieved through a circular shift mechanism. By shifting the quadrants by one-quarter of the latent dimensions, the system creates spatially offset overlapping regions. This ensures that the denoising process treats the replicated patterns as a unique, coherent structure rather than a repetitive grid of identical copies.

4.2.4 Synchronized Transformation

A critical technical requirement for the callback is the synchronized transformation of three specific tensors: the *latents*, the *mask* and the *masked_image_latents*. By rolling all three inputs simultaneously, the system maintains spatial alignment between the noise and the context. If this synchronization were broken, the model would attempt to inpaint inconsistent regions at each step, leading to failure of the generative process.

Finally, once the denoising loop is complete, the refined latent representation is processed by the VAE decoder (the *ft-mse* variant mentioned in Section 4.1.1) to produce the final high-resolution output.

Chapter 5

Results and discussion

This chapter presents the experimental validation of the proposed workflow for tileable textile synthesis. The evaluation encompasses a comprehensive analysis of the model's components and their impact on the quality of generated assets, examining how different conditioning strategies, architectural interventions and timing mechanisms contribute to the overall synthesis process.

The testing phase covers a diverse range of textile materials, organized into three structural categories: geometric meso-structures with periodic symmetry, irregular patterns exhibiting macro-scale stochastic variations and high-frequency microstructures featuring fine fiber details. This classification enables a precise assessment of the workflow's capacity to handle different scales of complexity and structural hierarchies.

Section 5.2 analyzes the interplay between guidance mechanisms, visual conditioning and architectural modifications such as circular padding and latent replication. Section 5.3 discusses the technical boundaries identified during experimentation, particularly concerning VAE compression artifacts and the trade-offs between different expansion strategies. Finally, Section 5.4 synthesizes the findings and demonstrates the practical applicability of the toolkit for professional production environments.

All experiments were conducted on the following hardware configuration:

GPU specs:

- GPU model: NVIDIA Quadro P4000
- Architecture: Pascal (16nm FinFET)
- VRAM: 8 GB GDDR5
- CUDA CORES: 1792 CUDA Cores
- Performance FP32: 5.3 TFLOPS (Peak Single Precision)

CPU specs:

- Operating system: Ubuntu 24.04.3 LTS (Noble Numbat)
- Kernel version: 6.8.0-100-generic
- NVIDIA driver version: 535.288.01
- CUDA version: CUDA 12.2

5.1 Samples description

The selection of samples for the testing phase covers the full spectrum of textile materials used in AAA production. In computer graphics, textiles are viewed as hierarchical structures rather than simple surfaces. This hierarchy requires a differentiated approach based on the nature of the material, which can be either stationary or variable. To validate the objectives of this work, samples are divided into three categories based on the scale of their components, as illustrated in Figure 5.1. This classification allows for a precise evaluation of denoising and expansion processes against specific structural challenges.

(a) Geometric meso-structure (wool knit) showcasing periodic symmetry.

(b) Irregular pattern (leather) with macro-scale stochastic variations.

(c) High-frequency micro-structure (denim) featuring fine fiber details.

Figure 5.1: Classification of textile samples according to their structural hierarchy: (a) structured weaves, (b) organic/irregular motifs and (c) high-resolution micro-fibers.

The first group includes geometric textures, such as wool knit (Figure 5.1a) and twill patterns. These meso-structures possess symmetrical repetition and sharp boundaries. They serve to validate the precision of circular padding and latent replication. The main challenge for these textures is preserving thread alignment across the tile margins. In this context, the geometric accuracy of the padding ensures that meso-structures remain consistent across the expanded canvas without yarn interruptions. This category confirms that the mathematical transition to a toroidal topology preserves the continuity of structured weaves.

The second group focuses on irregular patterns that lack a regular structure and exhibit local variations. This category includes textiles such as leopard motifs and hybrid textures like natural leather (Figure 5.1b), which presents macro-scale chromatic irregularities. This case study tests the effectiveness of noise rolling and outpainting in preventing the wallpaper effect. By synthesizing variations, the model fills the expanded regions while maintaining motif density. This process avoids pattern drift, ensuring that the distribution of variations remains natural. These samples validate the model's ability to generate synthesized content that remains semantically aligned with the source without creating detectable repetitions.

Finally, high-frequency samples, including denim (Figure 5.1c) and jersey fibers, are used to evaluate the micro-structure of the fabric. These materials contain transparencies and fine connections that challenge VAE compression. This group is essential to assess the decoder's ability to reconstruct details without excessive blurring. Specifically, the ft-mse decoder is tested to ensure that interconnections between fibers remain sharp. This group

validates the workflow's capacity to maintain the visual clarity required for production assets, ensuring the final output satisfies high-resolution rendering standards.

5.2 Analysis of the model components

5.2.1 Guidance and conditioning balance

The effectiveness of the synthesis depends on the equilibrium between three forms of conditioning: the positive prompt, the negative prompt and the visual reference. Within this framework, each conditioning component performs a specific function for the material reconstruction.

The integration of these signals requires a specific weighting mechanism to prevent semantic conflicts and ensure structural fidelity. This regulatory role is performed by Classifier-Free Guidance (CFG) [41]. This technique allows the generative network, specifically the U-Net of the Stable Diffusion model, to regulate the influence of the input data during the synthesis.

Instead of utilizing an external network to evaluate the results, the system performs two simultaneous predictions at every denoising step. This process is governed by the following equation:

$$\epsilon_{cfg} = \epsilon_{uncond} + w \times (\epsilon_{cond} - \epsilon_{uncond}) \tag{5.1}$$

The variable ϵ_{cfg} represents the final noise used for the denoising process. This value stems from a linear extrapolation between two internal states. The term ϵ_{uncond} defines the unconditional prediction. In this state, the model generates noise estimates by ignoring the prompts and the visual adapter, using a null token as input. This represents the statistical memory of the model, indicating how a generic image would look without specific instructions.

Conversely, ϵ_{cond} is the conditional prediction that incorporates the influence of the text and the visual features. These features are processed via the decoupled cross-attention described in Section 4.1.2. This term represents the specific target, such as the weave of a fabric or the density of a fiber.

The subtraction $(\epsilon_{cond} - \epsilon_{uncond})$ establishes a direction within the latent space. This vector isolates the semantic signal by removing the background noise from the intended features. By subtracting the generic guess from the guided one, the model identifies the pure essence of the requested textile. The guidance scale, denoted as w, dictates the amplification factor of this direction. By applying this weight, the model enhances the intended characteristics and filters out generic patterns.

This process ensures that the synthesis remains faithful to the textile while maintaining visual clarity. However, the selection of w requires precision. An excessive scale forces the model toward extreme embeddings, leading to chromatic saturation and the loss of micro-texture.

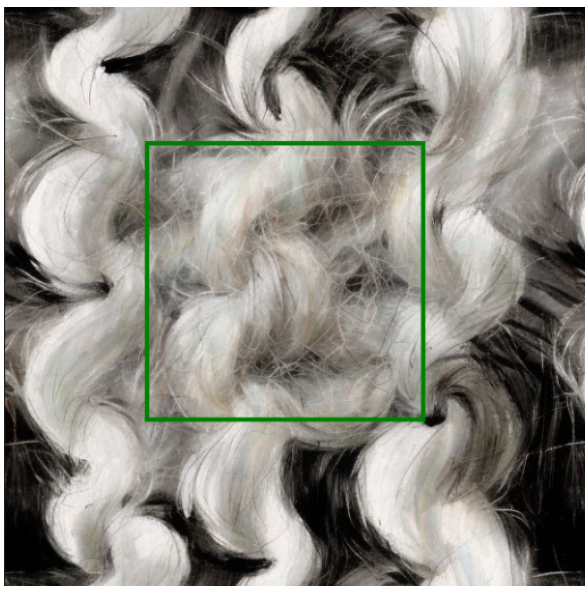
Positive Prompting

The positive prompt is a conditional input consisting of linguistic tokens that define the desired attributes of the generated image. It serves as the semantic anchor for the denoising process. While the IP-Adapter provides structural guidance based on the reference image, the textual prompt maintains a high priority in defining the global style.

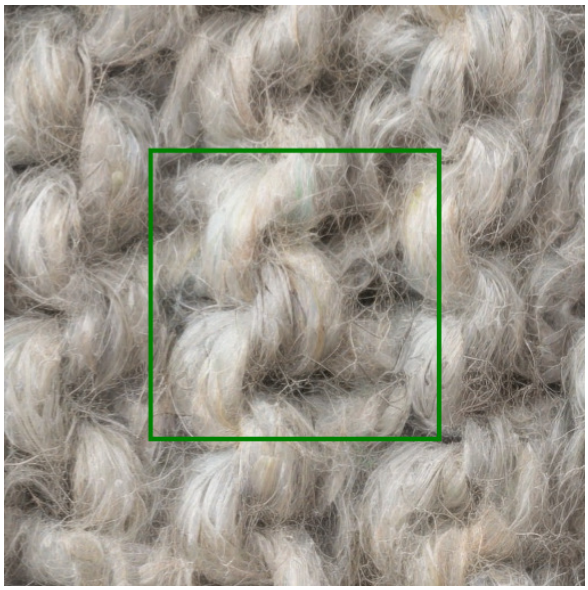
The selection of the specific prompt "textile, close up, hyperrealistic, top down view, high definition", utilized for the synthesis in Figure 5.2b, follows a precise logic designed to align the latent distribution with professional standards. Each token acts as a generative constraint that counteracts the natural bias of the model toward general photography. The term "textile" establishes the primary semantic anchor, ensuring that the U-Net prioritizes weights associated with fibers rather than other types of materials.

The scale of the synthesis is managed by the "close up" descriptor. Without this instruction, the model might interpret the reference as a distant garment or a furniture piece, leading to a loss of the microscopic weave identity. By forcing a macro perspective, the network concentrates its predictive capacity on the interlacing of the yarn. This is complemented by the "top down view" token, which is important for the creation of tileable assets. This instruction establishes an orthographic perspective that eliminates vanishing points and perspective distortions. This geometric alignment ensures that the generated texture remains flat, facilitating its application within PBR shaders and digital environments.

Finally, the descriptors hyperrealistic and high definition act as effective boosters within the latent manifold. By emphasizing the physical accuracy of the surface, they instruct the network to preserve the internal shadows between fibers. This maintenance of visual weight ensures a tactile suggestion that remains consistent with the fabric identity. In summary, this combination of tokens provides a stable frame that ensures structural fidelity and perspective consistency.



(a) Synthesis with a generic prompt.



(b) Synthesis with the technical prompt.

Figure 5.2: Impact of stylistic tokens on material identity. The use of a generic prompt like "cloth soft fabric, simple texture" in (a) forces the model to ignore the realistic details of the seed, leading to a simplified interpretation. Conversely, the adoption of technical descriptors in (b) ensures that the network maintains the microscopic complexity of the reference.

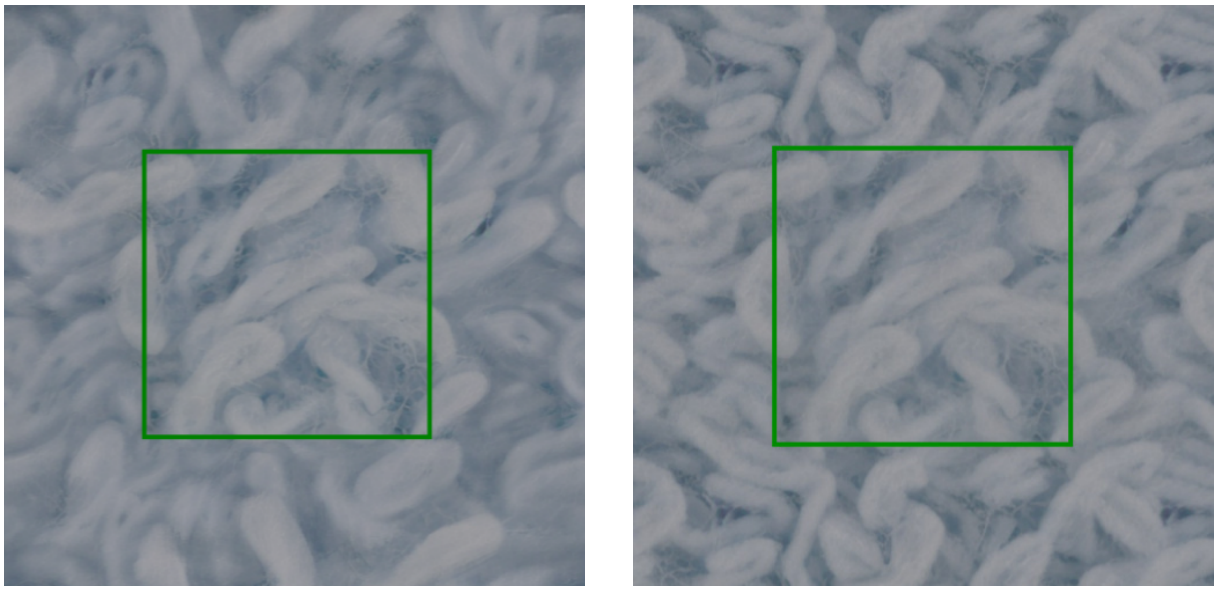
Negative Prompting

In this work, the negative prompt utilized is: "cartoon, drawing, painting, illustration, blurry, smooth, deformed, noisy". This specific conditioning, applied in the synthesis

shown in Figure 5.3b, operates through neutralization within the latent space. By defining what the model must avoid, the trajectory of denoising moves away from common artifacts. This list directly mirrors the problematic descriptors that compromise the material identity when present in the positive conditioning. By negating these specific terms, the system achieves a symmetrical balance.

An inadequate prompt creates a semantic conflict within the latent space. When descriptors such as "cartoon", "drawing", or "illustration" are utilized, the model receives a stylistic command that overrides the physical nature of the material. The U-Net interprets these tokens as instructions to simplify the visual forms. Instead of synthesizing the microscopic complexity of the fibers, the network begins to group pixels into flat regions of color. This phenomenon occurs because the weights of the model associated with the word "cartoon" are linked to datasets of drawings where microscopic texture is absent.

The resulting surface respects the geometry of the fabric but loses its material consistency. Furthermore, the inclusion of terms like "blurry" or "smooth" acts as a filter. These descriptors instruct the network to ignore the brightness gradients that define the relief of the yarn. This leads to the loss of the material hand, which represents the tactile sensation suggested by the visual appearance. Without the internal shadows of the weave, the material loses its visual weight. A heavy wool conditioned by an illustrative prompt appears as a light surface.



(a) Synthesis without negative constraints. **(b)** Synthesis with the negative prompt.

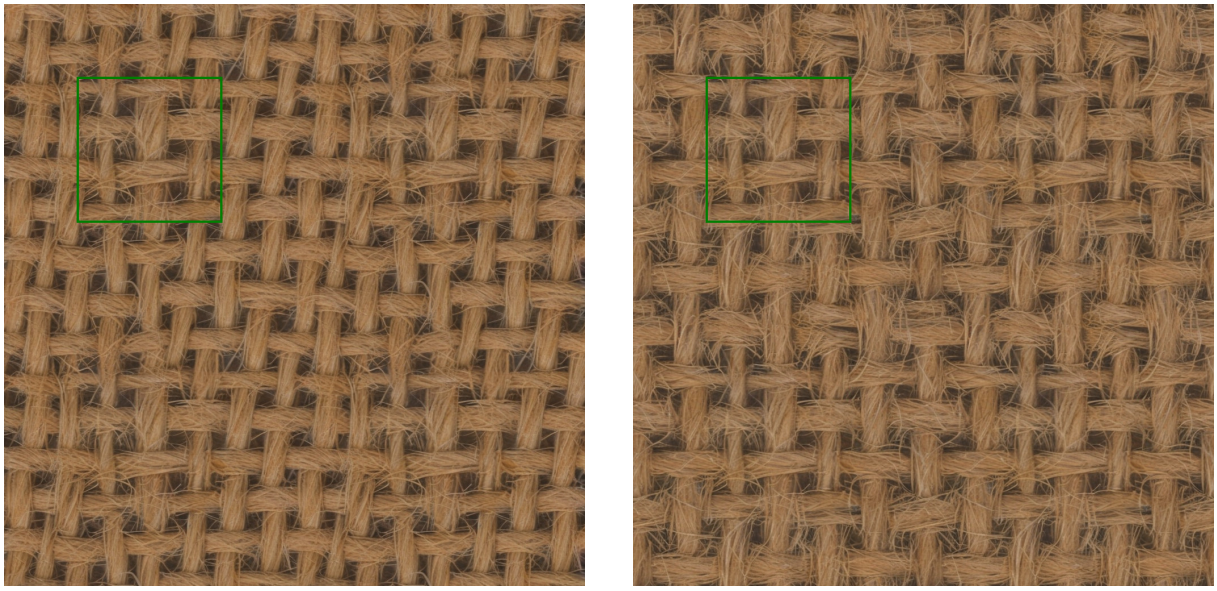
Figure 5.3: Visual precision and material realism through negative prompting. Figure (a) illustrates synthesis degradation due to artifacts and blur. Figure (b) demonstrates how the specific negative prompt prevents this simplified interpretation, ensuring surface clarity and microscopic detail during the outpainting phase.

During the expansion of the texture, the model must generate new pixels that remain coherent with the original seed. If a domain mismatch occurs, the external regions diverge from the center. This creates a visible junction caused by conceptual inconsistency. Instructions such as "deformed" or "noisy" further degrade the output by introducing statistical variations that the VAE decoder interprets as compression artifacts. By placing these terms in the negative conditioning, the U-Net is instructed to minimize their influence. This active exclusion preserves the material consistency and counteracts the

natural bias of the model toward centered compositions. As shown in Figure 5.3, these constraints prevent blurry regions and visible seams, ensuring the high precision required for professional workflows.

Visual Reference

The visual reference acts as a structural foundation and it's introduced through the IP-Adapter architecture.



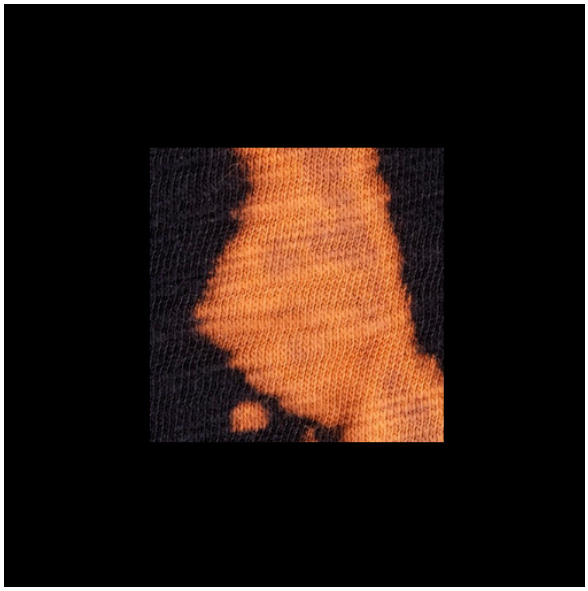
(a) Synthesis without visual guidance. **(b)** Synthesis with visual guidance.

Figure 5.4: Impact of visual conditioning on structural fidelity. The absence of an image reference in (a) causes geometric distortion within the expanded regions. The integration of the IP Adapter in (b) maintains the structural alignment of the fibers throughout the process.

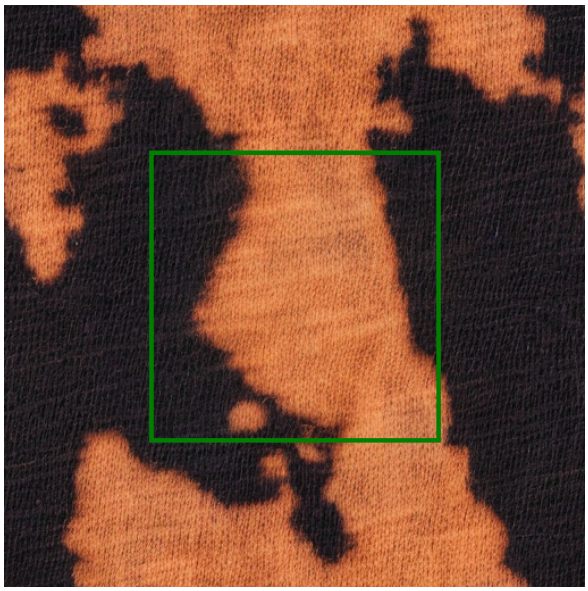
(a) Visual reference for the IP Adapter. **(b)** Synthesis with a high scale.

Figure 5.5: Impact of the adapter scale on structural fidelity. An excessive value of $\lambda = 0.9$ in (b) causes the replication of local noise and compression artifacts from the source image in (a).

The factor λ modulates the contribution of the image prompt within the attention layers. The absence of this conditioning causes a structural deviation. Without the visual guidance, the model must invent the geometry of the weave based solely on textual data. Figure 5.4 illustrates this limitation, which is critical for geometric textures where the alignment of fibers is fundamental.

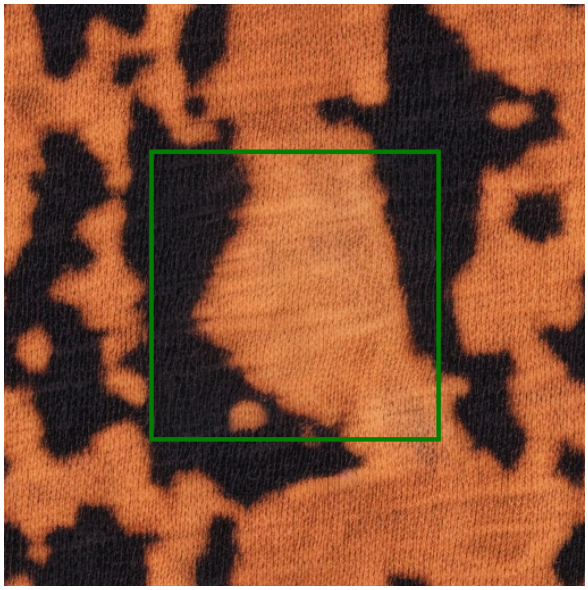


(a) Guidance image with a single seed.



(b) Synthesis with sparse pattern distribution.

(c) Guidance image with a tiled grid.



(d) Synthesis with dense pattern distribution.

Figure 5.6: Influence of IP-Adapter conditioning on irregular textures. The isolated seed in (a) produces a minimal distribution in (b). The structural replication in (c) facilitates the generation of a copious pattern in (d), demonstrating the role of the visual reference in modulating spatial density.

The intensity of the visual influence depends on the calibration of λ. A scale between 0.6 and 0.8 allows the model to inherit the weave structure while maintaining the flexibility to synthesize new regions. If the scale is too low, the expanded areas begin to hallucinate patterns that do not match the source. Conversely, an excessive scale causes structural

rigidity. When λ exceeds 0.9, the model replicates the specific noise and compression artifacts of the source instead of its textile logic. This excessive fidelity results in visible boundaries and chromatic distortions. Figure 5.5b demonstrates an instance where a high scale forces the model to burn the texture, destroying the midtones that are necessary. The final output tends to respect more the structure, but loses variety and realism. The equilibrium between the adapter scale λ and the global scale w is therefore vital to create tileable materials that are faithful to the physical sample and visually continuous.

Furthermore, the IP-Adapter represents a significant component in the modulation of the outpainting process. Its influence is evident in the control of pattern density and material color. Within the category of irregular motifs, the system leverages the IP-Adapter through a distinct methodology compared to regular textiles. By modifying the reference image provided to the adapter, the workflow actively conditions the generative behavior of the model. This strategy allows for the specific adjustment of the distribution, as shown in Figure 5.6 and orientation of the pattern during the expansion phase.

5.2.2 Circular padding activation

Once the identity of the textile is established through conditioning signals, the activation of circular padding represents the transition from structural composition to periodic refinement. While noise rolling manages the global alignment of the texture, circular padding ensures that neural filters maintain local continuity across boundaries.

The selection of the activation threshold is determined by the internal mechanics of the U-Net. During the initial 40% of the denoising process, the model relies on zero-padding as a form of implicit positional encoding. This spatial anchoring allows the convolutions to orient the layout within the latent grid. If circular padding is activated prematurely, as shown in Figure 5.7a, the model loses these spatial references. This results in a degradation of generative quality where the material fails to define a coherent structure.

(a) Circular padding activated from the beginning. Loss of details.

(b) Circular padding activated at 40% of the denoising steps.

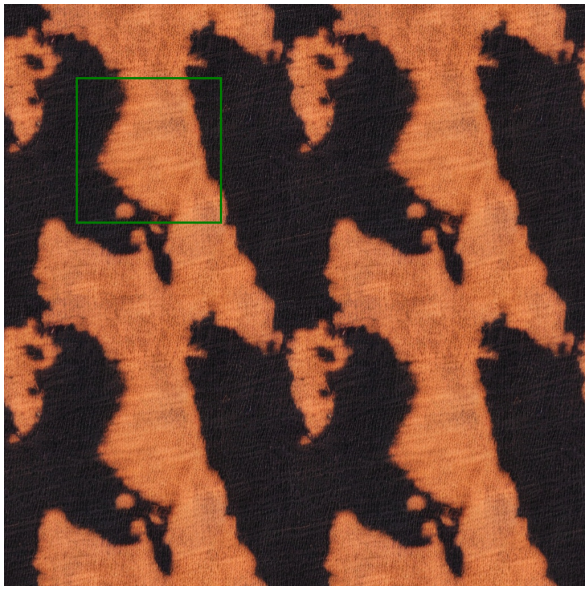
Figure 5.7: Visual impact of padding activation timing.

Beyond the 40% mark, the global layout is typically consolidated. At this stage,

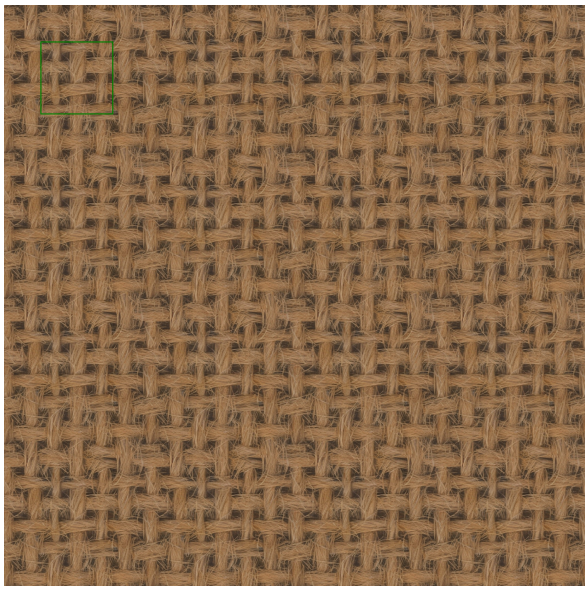
the priority shifts toward the synthesis of microscopic details and fiber continuity. By replacing standard convolutions with circular ones, the system transforms the latent plane into a periodic surface. When a kernel moves beyond a boundary, it samples pixels from the opposite edge. This mechanism facilitates a seamless flow of gradients, ensuring that the weave of the fabric continues without interruptions. Figure 5.7b illustrates how this delayed activation preserves the material integrity while imposing perfect tileability.

The intervention must extend to the VAE to ensure a successful output. Standard decoders often introduce micro-seams during the final reconstruction in the pixel space. These artifacts emerge because the convolutional layers of the VAE utilize zero-padding during the inference. By applying circular decoding, the transition from the compressed latent space to the final image maintains geometric continuity. This final step prevents the appearance of edge artifacts that would otherwise compromise the utility of the asset for rendering.

The coordination between these techniques facilitates the achievement of perfect native tileability across diverse textile categories and with both workflows.



(a) Visual verification of perfect seamless tiling for an aperiodic textile output within a 2048×2048 expanded grid.



(b) Visual verification of perfect seamless tiling for a highly structured regular weave output within a 2048×2048 expanded grid.

Figure 5.8: Comprehensive visual evaluation of native tileability for diverse synthesized textile patterns.

5.2.3 The latent replication activation

The selection of the 60% threshold for the replication process described in Section 4.1.4 corresponds to a strategic window between mode separation and the concentration phase. This timing is critical for the structural success of the expansion. The experimental results in Figure 5.9 illustrate the consequences of improper activation.

Activating the process before the 60% mark is premature. During these early stages, the model performs mode separation to establish the global identity and the semantic layout of the texture. If the expansion occurs at 10%, the model lacks a consolidated structure to replicate. As shown in Figure 5.9a, this instability deforms the geometry

and introduces incoherent details. Since the latent representation is not yet stable, the U-Net synthesizes divergent patterns in each quadrant. This leads to a loss of structural continuity where the fabric weave appears fractured.

(a) Replication at 10%: geometric deformations.

(b) Replication at 60%: optimal balance.

Figure 5.9: Visual comparison of activation strategies for latent replication.

The 60% mark represents a state of optimal malleability. At this interval, the semantic structure is already crystallized, ensuring that the replicas share a common identity. Figure 5.9b demonstrates how this balance maintains the integrity of the latent manifold. The noise level remains high enough to allow the U-Net to blend the quadrants without structural discontinuities. This approach justifies the choice of a structured initialization over Gaussian noise to prevent the emergence of inconsistent patterns.

5.2.4 Deactivation of noise rolling in the final steps

The implementation of the rolling offset after the replication manages the boundaries of the original quadrants. By shifting the junctions to the center of the receptive field, the model perceives the edges as informative context, ensuring that the textile weave continues without interruptions. The U-Net treats the seams as missing regions, utilizing the surrounding data to synthesize a coherent transition.

Disabling the rolling offset beyond the 90% threshold serves to stop the spatial drift of the latents. This stability is required to eliminate blur. If the displacement continued until the final step, small variations in the latent positions would result in a soft appearance. Stopping the rotations allows the model to concentrate the predictive capacity on microscopic fibers and specular reflections.

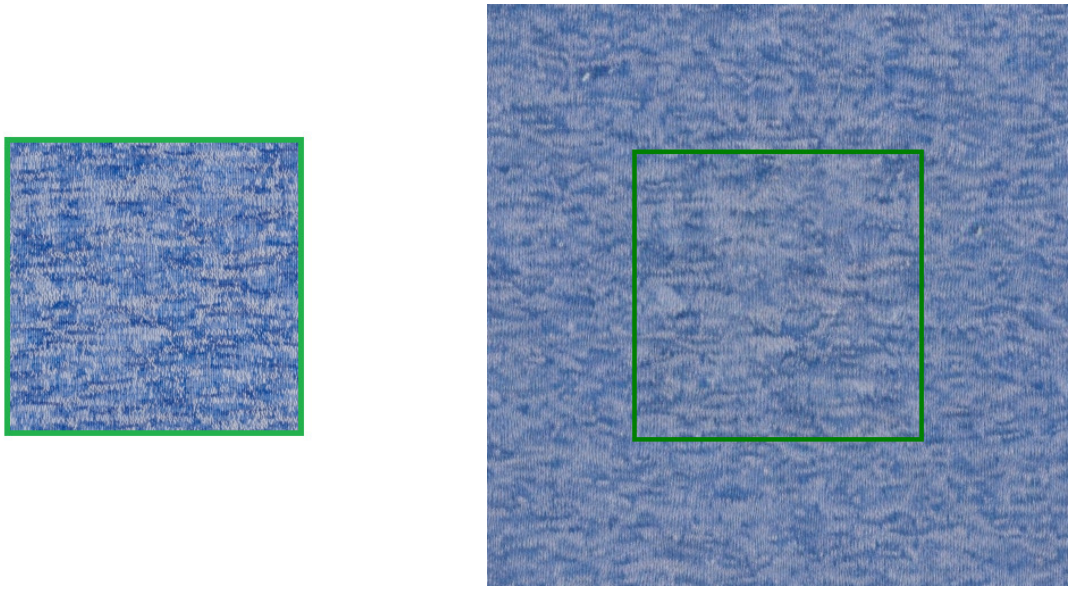
5.3 Limits

5.3.1 Loss of details

Despite the positive outcomes, the experimentation identifies technical boundaries within this framework. These constraints define the applicability of the model in production.

The primary limitation involves the VAE compression, which represents a fundamental bottleneck for high-frequency detail.

Since the diffusion process occurs in a latent space reduced by a factor of eight, microscopic information is often lost. An image of 512 pixels becomes a matrix of 64 units, causing a smoothing effect on textures with dense spatial data. As shown in Figure 5.10b, materials like jersey suffer from this loss of definition. Even if the model respects the global coherence of the seed, it fails to represent the high-frequency detail of the fibers. This limitation results in a blurred output and chromatic inaccuracies that degrade the material identity.



(a) Jersey seed **(b)** Result of the outpainting from jersey seed.

Figure 5.10: Comparison between input seed and outpainting result.

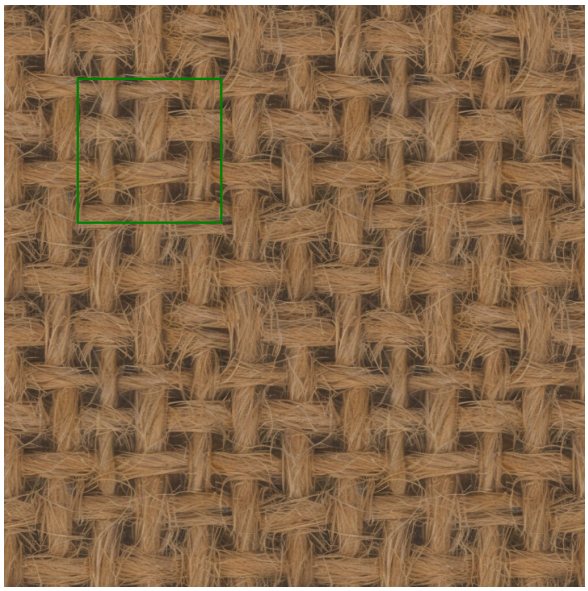
A partial mitigation is provided by the *ft-mse* variant of the VAE decoder. This version improves the reconstruction of edges and reduces the appearance of artifacts. However, the softening of textures remains a consequence of computational efficiency. The latent bottleneck is designed to prioritize semantic structure over microscopic sharpness.

Within the video game industry, this limitation can be managed through the detail mapping strategy described in Section 1.2.1. The image produced by the diffusion model acts as a structural macro-map. This map delineates the forms, seams and color variations of the textile. Since game engines restore surface grains through secondary micro-maps, the loss of micro-scale information in the base texture is often acceptable. A secondary tiling map adds the necessary detail to the macro-structure, resolving the lack of sharpness. This layered approach allows the generative model to provide the structural foundation while specialized shaders maintain visual clarity. Consequently, the diffusion model remains a powerful tool for the synthesis of large-scale features that avoid the repetitive nature of procedural noise.

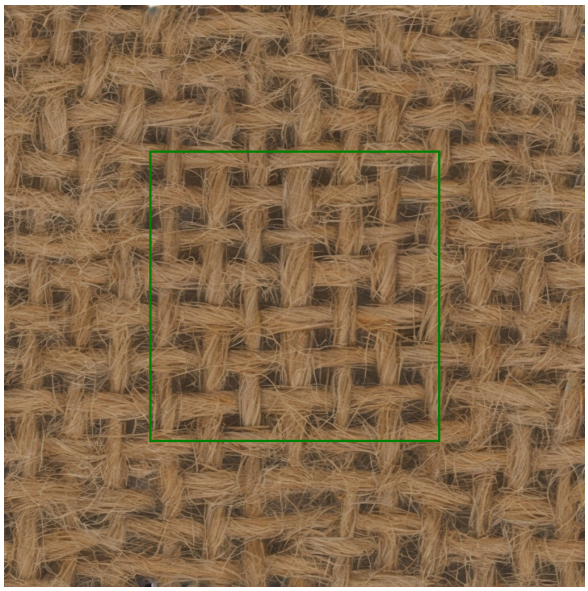
5.3.2 Limitations of the latent replication approach

Another constraint involves the latent replication strategy utilized in this work. While this approach provides control over the material structure, it introduces specific constraints regarding visual variety. In this workflow, the model performs an initial outpainting from

a 256×256 seed to a 512×512 latent tensor. Subsequently, the system expands the canvas to 1024×1024 through the replication process described in Section 4.1.4.



(a) Final result with outpainting from 256 to 512 and latent replication.



(b) Final result with native outpainting from 512 to 1024.

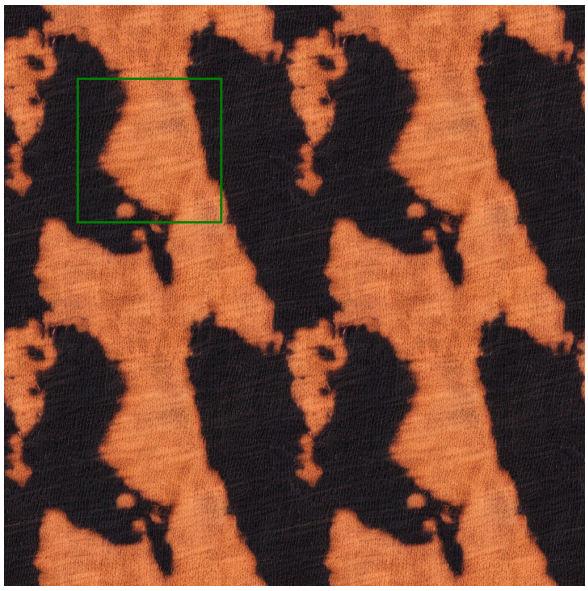
Figure 5.11: Different methods for the synthesis of regular textiles like jute.

The choice between latent replication and native outpainting involves a trade-off between geometric stability and visual variety. The replication strategy is ideal for regular textures, such as the jute shown in Figure 5.11a. The cloning of the seed across the quadrants guarantees perfect tileability and prevents geometric distortions. However, this strategy introduces a periodic bias. The primary consequence of this replication is the emergence of feature cloning. Every distinctive element within the seed appears four times in a rigid grid. This redundancy signals a synthetic origin to the human observer, especially in organic textures where the repetition of a specific stain or fiber breaks the visual randomness.

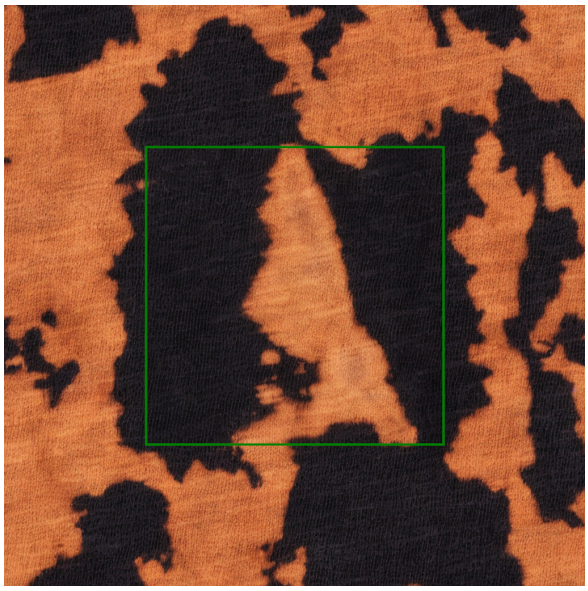
Conversely, native outpainting from 512 to 1024 pixels avoids the repetition of features. This method allows the model to synthesize a continuous motif without cloning the mask. While this is beneficial for irregular patterns, it results in a loss of structural control. For geometric textiles, the absence of a replicated grid causes the weave to drift or deform across the expanded area. This phenomenon is linked to the receptive field of the U-Net architecture.

The convolutional kernels of Stable Diffusion v1.5 are optimized for a spatial scale of 512×512 pixels. When the model operates directly at 1024×1024, the relative coverage of each kernel decreases. The network captures a smaller portion of the global context, leading to semantic fragmentation. Without a structured latent prior, the model cannot maintain distant correlations across the canvas. This leads to the structural drift observed in Figure 5.11b, where the weave loses its linear orientation.

Furthermore, native expansion often fails to achieve perfect tileability despite the use of circular padding. This limitation stems from the training distribution of the base model. Operating at expanded resolutions shifts the internal statistics of the activations. This shift causes a degradation in the periodic coherence of the boundary gradients. Circular padding cannot compensate for this statistical divergence because the model lacks the



(a) Final result with outpainting from 256 to 512 and latent replication.



(b) Final result with native outpainting from 512 to 1024.

Figure 5.12: Different methods for the synthesis of irregular patterns.

learned experience to process large toroidal surfaces.

Consequently, the latent replication remains the superior choice for technical assets. Although it suffers from periodic repetition, it ensures the geometric precision required for professional tiling. The loss of variety is a functional cost to preserve the integrity of the material identity. In production environments, this limitation is often mitigated by the detail mapping strategy, which masks the periodic bias through the addition of secondary textures.

5.4 Results discussion

The experimental results demonstrate the effectiveness of the proposed diffusion workflow for generating tileable textile textures at a resolution of 1024×1024 pixels. The outputs show a total absence of seams at the boundaries, confirming the utility of the toroidal topology for synthesizing expansive surfaces from limited source samples. The integration of circular padding within the denoising process ensures that transitions between adjacent tiles remain mathematically and perceptually continuous, meeting the requirements for high fidelity digital environments.

5.4.1 Native tileability and seamless synthesis

The primary technical contribution of this work is the achievement of native tileability without post-processing. Unlike conventional approaches that require manual seam correction or blending operations, the proposed method generates seamless textures directly through the synthesis process. This is accomplished through the coordinated activation of circular padding and noise rolling mechanisms at specific stages of denoising.

During the outpainting phase, the model synthesizes new regions that maintain consistency with the seed context without producing repetitive artifacts. By utilizing the IP

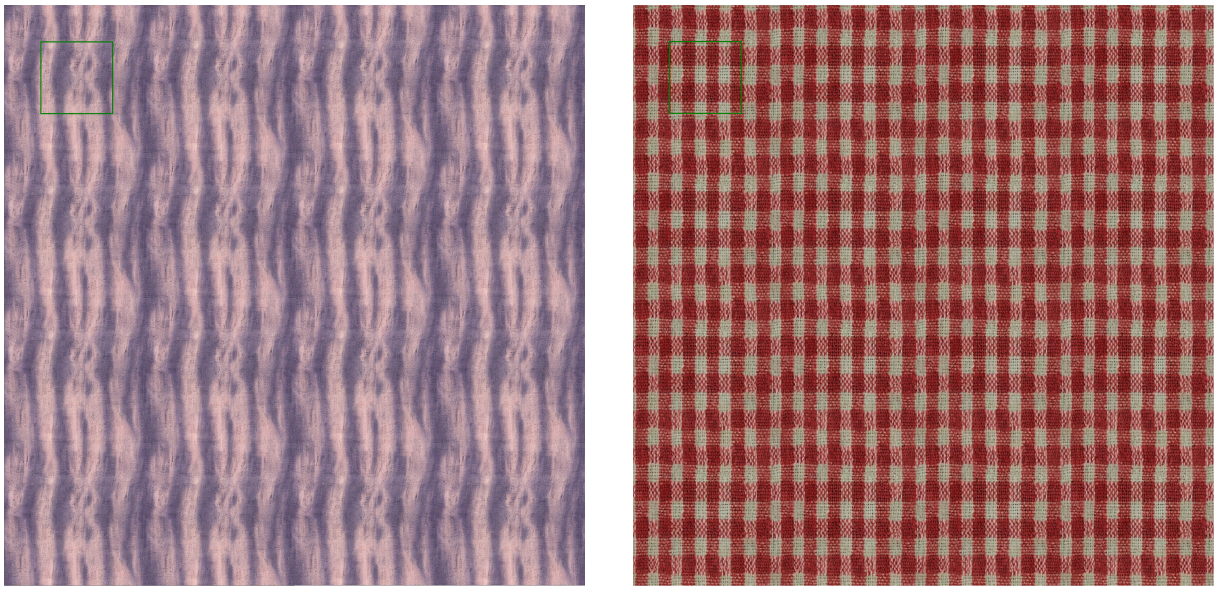


Figure 5.13: Visual verification of perfect tiling of the output within a 2048×2048 expanded grid.

Adapter for structural guidance, the algorithm analyzes the features of the seed to fill the expanded canvas while preserving the original weave density and fiber frequency. This balance between structural fidelity and stylistic variation is necessary for materials where motif distribution must appear organic.

The verification of tileability across diverse textile categories (as shown in 5.13) confirms the robustness of the approach. For regular weaves such as jute and structured fabrics, the model maintains geometric periodicity while ensuring thread alignment across tile margins. For irregular patterns such as leather and organic motifs, the system preserves stochastic distribution without introducing visible junctions. High-frequency microstructures including denim and jersey demonstrate that fine fiber details remain sharp and continuous across expanded grids.

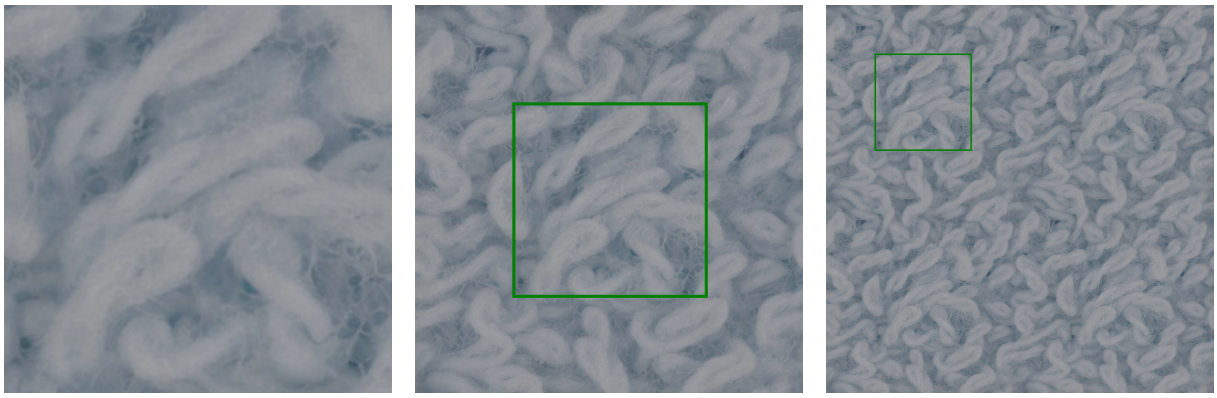
5.4.2 Semantic expansion and scale preservation

The workflow performs semantic expansion from 256×256 to 1024×1024 pixels, representing a 16-fold increase in spatial resolution. This expansion maintains the physical properties and perceived scale of the source textile rather than applying naive interpolation. The U-Net architecture preserves yarn thickness, fiber orientation and weave density across the canvas, ensuring that the generated textures remain faithful to the reference material.

Chromatic and lighting consistency are maintained throughout the spatial expansion process. The model preserves the original lighting characteristics and color values of the seed, preventing color drift or the emergence of artificial shadows.

The dual-pathway methodology addresses the distinct requirements of regular and irregular textile structures. For structured weaves, the latent replication pathway ensures geometric precision and pattern continuity.

For organic materials, the outpainting pathway expands the texture by generating believable pattern instances that naturally incorporate chromatic and structural variations



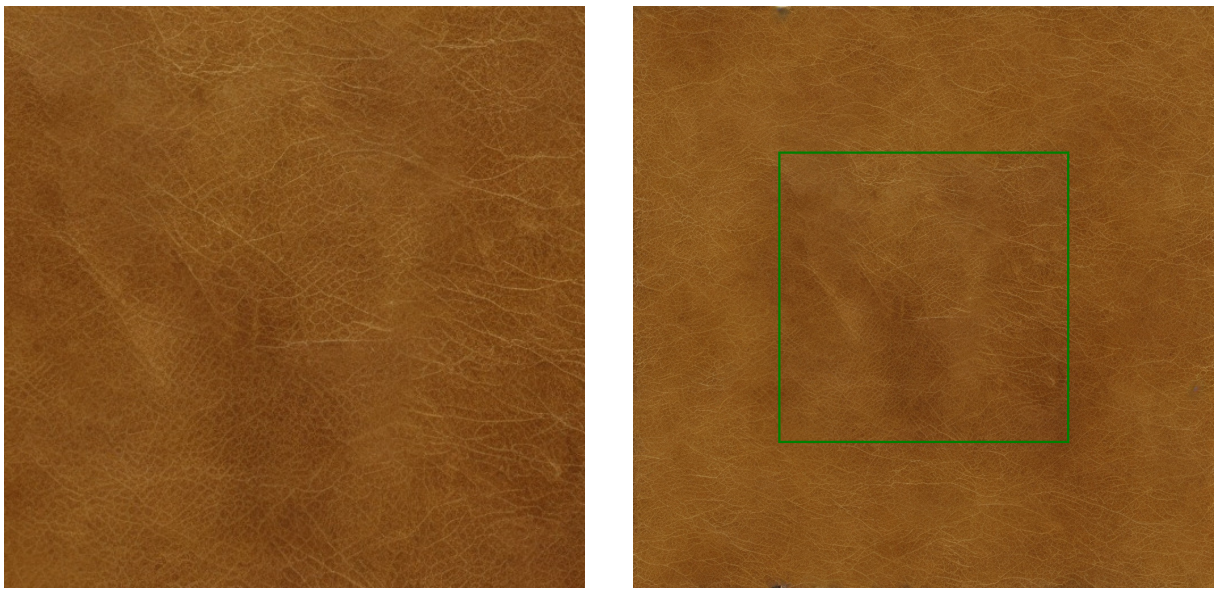
(a) Visual verification of perfect seamless tiling for an aperiodic textile output within a 2048 × 2048 expanded grid.

(b) Visual verification of perfect seamless tiling for a highly structured regular weave output within a 2048 × 2048 expanded grid.

(c) Visual verification of perfect seamless tiling for a highly structured regular weave output within a 2048 × 2048 expanded grid.

Figure 5.14: Comprehensive visual evaluation of native tileability for diverse synthesized textile patterns.

consistent with the source material. This adaptive strategy demonstrates that pattern topology fundamentally determines optimal synthesis configuration.



(a) Final synthesis of a leather textile.

(b) Final synthesis of a leather textile.

5.4.3 Production integration and workflow automation

The practical significance of this work extends beyond technical validation to address production realities in professional environments. The toolkit integrates with standard production tools without requiring workflow redesign. Artists maintain control through visual reference selection, ensuring that automated generation aligns with creative intent rather than producing arbitrary results. The preservation of creative control ensures that efficiency gains do not compromise artistic vision, addressing a fundamental concern in production automation.

Furthermore, the U-Net architecture preserves the physical properties and perceived scale of the source textile. By maintaining consistent yarn thickness and fiber orientation across the canvas, the model ensures that the generated textures remain faithful to the reference. These high resolution outputs facilitate the extraction of surface maps. As the junctions are mathematically seamless, the resulting normal and roughness maps do not exhibit specular artifacts or lighting discontinuities at the boundaries.

The primary technical advantage of this workflow lies in the tiling achieved through noise rolling and circular padding. Unlike standard outpainting methods that treat edges as absolute boundaries, the toroidal topology treats the canvas as a continuous loop, enabling infinite repetition on 3D models without visible grid patterns. The noise rolling technique further mitigates perceptual patterns by distributing generative variance across the latent space. In a professional production pipeline, this automation reduces the requirement for manual refinement while maintaining the quality standards necessary for modern material synthesis.

Conclusion

This thesis has investigated the challenges of automated textile synthesis within the constraints of AAA game development. The primary objective was to resolve the persistent conflict between generative flexibility and the structural requirements of textiles. By shifting the synthesis process from the pixel domain to the latent manifold, this work has demonstrated that it is possible to generate high-quality textile assets that are natively tileable and structurally coherent without the need for manual post-processing.

The experimental results confirm that the toolkit successfully produces professional assets capable of infinite repetition without perceptual artifacts. For structured weaves, the model maintained strict geometric periodicity, while for irregular materials the system preserved the stochastic distribution of features.

This research establishes several key findings regarding diffusion-based textile synthesis. The work demonstrates that pattern topology fundamentally determines optimal synthesis strategy, challenging the assumption that a single unified approach can adequately serve the full spectrum of textile materials. The validation of temporally coordinated interventions reveals that precise staging of architectural modifications significantly impacts output quality, with specific timing windows yielding superior results. Furthermore, the research confirms the effectiveness of example-based visual guidance for controlling material characteristics that resist adequate description through language alone. The investigation validates that structured initialization strategies outperform stochastic approaches when preserving geometric properties during spatial expansion.

The practical significance of this work extends beyond technical validation to address production realities in professional environments. The dual-pathway approach enables generation across different textile types while maintaining quality standards. The toolkit integrates with standard production tools without requiring workflow redesign. Artistic control through visual reference selection ensures automated generation aligns with creative intent rather than producing arbitrary results. Critically, the preservation of creative control ensures that efficiency gains do not compromise artistic vision, addressing a fundamental concern in production automation.

Several constraints bound the current scope. The methodology addresses diffuse map generation exclusively, leaving complete multi-channel material synthesis for future investigation. Pattern classification remains a manual decision rather than an automated process. The specific parameter values governing intervention timing were determined empirically and may require adjustment for different model configurations or material categories. Output quality inherits the characteristics of input samples, with source limitations propagating through the generation process.

Future Developments and Improvements

The current methodology provides a solid foundation for the synthesis of diffuse maps, yet several directions for future research remain open.

A primary evolution involves the direct generation of a complete PBR set. While the present framework produces precise color data, the simultaneous synthesis of normal, roughness and height maps within the same latent space would eliminate the requirement for external tools. This parallel synthesis would automate the elaborate pipeline required for PBR materials, effectively minimizing the human interaction currently needed for

digital assets. A multi-channel diffusion model would ensure that every microscopic detail in the diffuse map corresponds to a physical relief in the displacement map, enhancing the material realism and lighting accuracy.

Another development concerns the integration of specialized targeted tuning. Although the base model possesses an inherent capacity for textiles, a dedicated phase on specific macro datasets would address a primary limitation of the current workflow. At present, textile textures with dense details are not perfectly captured by the generative process due to the compression limits of the latent space. By optimizing the U-Net architecture on high-resolution textile samples, the network could learn the complex interlacing of diverse fabrics. This refinement would allow the model to synthesize microscopic structures that are currently blurred during the reconstruction phase, ensuring that even the smallest fibers maintain their visual identity in the final output.

Finally, the generation of procedural variations represents a significant advancement. While the present system produces a single seamless tile, a future expansion could involve the synthesis of multiple unique samples that maintain boundary continuity with each other. This methodology would prevent the repetitive appearance of textures on massive surfaces in virtual environments. By exploring the latent space around a material identity, the model could produce a diverse collection of compatible tiles. This evolution would facilitate the implementation of stochastic tiling within game engines, ensuring that every region of a digital world remains unique while preserving the physical properties of the source material.

Bibliography

[1] M. Ashikhmin and P. Shirley, "An anisotropic phong brdf model," *Journal of Graphics Tools*, vol. 5, Jan. 2001. DOI: 10.1080/10867651.2000.10487522

[2] Unity, *Mask and detail maps in hdrp*, Accessed: 2026-03-04, 2025. [Online]. Available: https://docs.unity3d.com/Packages/com.unity.render-pipelines.high-definition@17.5/manual/Mask-Map-and-Detail-Map.html

[3] Epic Games, *Detail textures in unreal engine 5*, Accessed: 2026-03-04, 2023. [Online]. Available: https://dev.epicgames.com/documentation/en-us/unreal-engine/adding-detail-textures-to-unreal-engine-materials

[4] NaughtyDog, *Parallelizing the Naughty Dog Engine*, Accessed: 2026-03-05, 2015.

[5] J. Crossland, *The art of characters in alan wake 2: Character outfit creation process*, GDC Vault, Game Developers Conference (GDC), 2024.

[6] K. Vaidyanathan, M. Salvi, B. Wronski, T. Akenine-Möller, P. Ebelin, and A. Lefohn, "Random-Access Neural Compression of Material Textures," in *Proceedings of SIGGRAPH*, 2023.

[7] Texturebook, *Geometry and image textures*, Accessed: 2026-03-03, 2022. [Online]. Available: https://www.letourneaudesign.com/texture-guidebook

[8] W. P.-B. Rendering, "Physically-based rendering," *Procedia IUTAM*, vol. 13, no. 127-137, p. 3, 2015.

[9] S. Hill et al., "Physically based shading in theory and practice," in *ACM SIGGRAPH 2020 Courses*, 2020, pp. 1–12.

[10] Lightbeans, *Pbr textures*, Accessed: 2026-03-03, 2024. [Online]. Available: https://lightbeans.com/en/blog/tutorials/pbr-physically-based-rendering-explained-a-quick-guide

[11] Forum, *Techniques to reduce terrain texture repating effect*, Accessed: 2026-03-04, 2024. [Online]. Available: https://hub.jmonkeyengine.org/t/techniques-to-reduce-terrain-texture-repeating-effect/45623/8

[12] Y. LeCun, S. Chopra, R. Hadsell, M. Ranzato, F. Huang, et al., "A tutorial on energy-based learning," *Predicting structured data*, vol. 1, no. 0, 2006.

[13] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[14] S. Hussain. "Reparameterization-trick in vaes explained." Accessed: 2026-02-27. [Online]. Available: https://snawarhussain.com/blog/genrative%20models/python/vae/tutorial/machine%20learning/Reparameterization-trick-in-VAEs-explained/

[15] I. Goodfellow et al., "Generative adversarial networks," *Communications of the ACM*, vol. 63, no. 11, pp. 139–144, 2020.

[16] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," *Advances in neural information processing systems*, vol. 33, pp. 6840–6851, 2020.

[17] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 10 684–10 695.

[18] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*, Springer, 2015, pp. 234–241.

[19] K. Perlin, "An image synthesizer," in *ACM SIGGRAPH Computer Graphics*, ACM, vol. 19, 1985, pp. 287–296.

[20] Ken Perlin, *Improved noise reference implementation*, Accessed: 2026-03-10. [Online]. Available: `https://mrl.cs.nyu.edu/~perlin/noise/`

[21] A. A. Efros and T. K. Leung, "Texture synthesis by non-parametric sampling," in *Proceedings of the seventh IEEE international conference on computer vision*, IEEE, vol. 2, 1999, pp. 1033–1038.

[22] S. Geman and D. Geman, "Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images," *IEEE Transactions on pattern analysis and machine intelligence*, no. 6, pp. 721–741, 1984.

[23] A. A. Efros and W. T. Freeman, "Image quilting for texture synthesis and transfer," 2001.

[24] L. Gatys, A. S. Ecker, and M. Bethge, "Texture synthesis using convolutional neural networks," *Advances in neural information processing systems*, vol. 28, 2015.

[25] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv: 1409.1556*, 2014.

[26] D. C. Liu and J. Nocedal, "On the limited memory bfgs method for large scale optimization," *Mathematical programming*, vol. 45, no. 1, pp. 503–528, 1989.

[27] E. Risser, P. Wilmot, and C. Barnes, "Stable and controllable neural texture synthesis and style transfer using histogram losses," *arXiv preprint arXiv: 1701.08893*, 2017.

[28] D. Ulyanov, V. Lebedev, A. Vedaldi, and V. Lempitsky, "Texture networks: Feed-forward synthesis of textures and stylized images," in *International Conference on Machine Learning (ICML)*, 2016, pp. 1349–1357.

[29] D. Ulyanov, A. Vedaldi, and V. Lempitsky, "Instance normalization: The missing ingredient for fast stylization," *arXiv preprint arXiv: 1607.08022*, 2016.

[30] N. Jetchev, U. Bergmann, and R. Vollgraf, "Texture synthesis with spatial generative adversarial networks," *arXiv preprint arXiv: 1611.08207*, 2016.

[31] U. Bergmann, N. Jetchev, and R. Vollgraf, "Learning texture manifolds with the periodic spatial gan," in *International Conference on Machine Learning (ICML)*, 2017, pp. 469–477.

[32] T. R. Shaham, T. Dekel, and T. Michaeli, "Singan: Learning a generative model from a single natural image," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019, pp. 4570–4580.

[33] A. Hertz, R. Hanocka, R. Raja, and D. Cohen-Or, "Deep geometric texture synthesis," *ACM Transactions on Graphics (TOG)*, vol. 39, no. 4, pp. 108–1, 2020.

[34] A. Radford et al., "Learning transferable visual models from natural language supervision," in *International conference on machine learning*, PmLR, 2021, pp. 8748–8763.

[35] M. Riso, G. Vecchio, and F. Pellacini, "Structured pattern expansion with diffusion models," *arXiv preprint arXiv:2411.08930*, 2024.

[36] E. J. Hu et al., *Lora: Low-rank adaptation of large language models*, 2021. arXiv: 2106.09685 [cs.CL]. [Online]. Available: https://arxiv.org/abs/2106.09685

[37] S. AI, *Stable diffusion vae fine-tuned models*, https://huggingface.co/stabilityai/sd-vae-ft-mse, 2022.

[38] H. Ye, J. Zhang, S. Liu, X. Han, and W. Yang, "Ip-adapter: Text compatible image prompt adapter for text-to-image diffusion models," *arXiv preprint arXiv: 2308.06721*, 2023.

[39] G. Vecchio et al., "Controlmat: A controlled generative approach to material capture," *ACM Transactions on Graphics*, vol. 43, no. 5, pp. 1–17, Sep. 2024, ISSN: 1557-7368. DOI: 10.1145/3688830 [Online]. Available: http://dx.doi.org/10.1145/3688830

[40] O. Bar-Tal, L. Yariv, Y. Lipman, and T. Dekel, "Multidiffusion: Fusing diffusion paths for real-time image generation," *arXiv preprint arXiv: 2302.08113*, 2023.

[41] J. Ho and T. Salimans, "Classifier-free diffusion guidance," *arXiv preprint arXiv: 2207.12598*, 2022.

Ringraziamenti

Desidero esprimere la più sincera e profonda gratitudine alla mia relatrice Prof.ssa Serena Morigi per avermi guidato e supportato in questo lavoro con preziosi consigli e costante disponibilità. Rivolgo un sincero ringraziamento al tutor Paolo Zuzolo per il continuo confronto e la dedizione con la quale ha seguito ogni fase della ricerca.

Un ringraziamento immenso va ai miei genitori per tutto l'amore e i sacrifici fatti per permettermi di raggiungere questo traguardo. Grazie per essere stati la mia prima e più grande fonte di ispirazione: guardando voi, ho imparato il valore della perseveranza e l'importanza di affrontare la vita con coraggio e umiltà. Grazie per il vostro sostegno incondizionato e per aver creduto in me anche quando io stessa faticavo a farlo. Siete stati la mia roccia: sapere di poter contare su di voi ha reso ogni ostacolo più leggero. Se oggi sono qui, è soprattutto merito della forza e della fiducia che avete saputo infondermi.

A mia sorella Roberta, il mio porto sicuro e la mia certezza più grande. Grazie per avermi sostenuta in ogni occasione, per tutto questo tempo. Sei sempre stata la prima a credere in me, in ogni mia scelta e in ogni mio passo. Grazie per essermi stata accanto nei momenti di sconforto e per aver gioito dei miei successi con la stessa intensità con cui li ho vissuti io. Non riuscirei a immaginare questo percorso senza di te.

A Filippo. Sei stato la mia luce nei momenti più bui, la mia ispirazione quando le idee mancavano e il mio sostegno più solido. Ti bastava uno sguardo per capire esattamente come mi sentissi, trovando sempre il modo di ridarmi la forza per non mollare. Grazie per aver reso questo viaggio un'esperienza meravigliosa da vivere insieme.

A Flaminia, un'amica rara e preziosa. Grazie per avermi ascoltata con una dolcezza che porterò sempre nel cuore, per aver saputo quando spronarmi e quando distrarmi, ricordandomi l'importanza di restare me stessa oltre i libri.

A Emanuele, per la sua presenza speciale e il legame profondo che ci unisce. Grazie per essere stato fonte di ispirazione e per avermi permesso di evadere dalla realtà nei momenti più difficili. Ritrovare la giusta leggerezza è stato fondamentale per ricaricare le energie e andare avanti.

A Mariano, compagno di mille giornate tra i libri e i banchi dell'università. Grazie per il confronto, per le risate e per aver reso gli anni dell'università un'avventura indimenticabile.

Un pensiero va anche a tutta la mia famiglia, per il grande affetto che mi ha dimostrato e per aver sempre fatto il tifo per me.

Grazie a tutti i miei amici, sia quelli storici che mi conoscono da sempre, sia quelli incontrati più recentemente: ognuno di voi ha aggiunto un tassello prezioso alla mia vita, rendendomi la persona che sono oggi.

Infine, un ringraziamento speciale al piccolo Francis, per la sua presenza rassicurante durante le lunghe ore passate a scrivere. La sua compagnia è stata la migliore terapia contro ogni stress.