

FabricGen: Microstructure-Aware Woven Fabric Generation

Yingjie Tang^{1†} Di Luo^{1†} Zixiong Wang¹ Xiaoli Ling² Jian Yang^{1,2} Beibei Wang^{2*}
¹Nankai University ²Nanjing University



Figure 1. We present FabricGen, an end-to-end material generation framework that generates woven fabric materials with fine details, enabling non-expert users to create diverse and highly detailed fabric materials for photorealistic rendering.

Abstract

Woven fabric materials are widely used in rendering applications, yet designing realistic examples typically involves multiple stages, requiring expertise in weaving principles and texture authoring. Recent advances have explored diffusion models to streamline this process. However, pre-trained diffusion models often struggle to generate intricate yarn-level details that conform to weaving rules. To address this, we present FabricGen, an end-to-end framework for generating high-quality woven fabric materials from textual descriptions. A key insight of our method is the decomposition of macro-scale textures and micro-scale weaving patterns. To generate macro-scale textures free from microstructures, we fine-tune pre-trained diffusion models on a collected dataset of microstructure-free fabrics. As for micro-scale weaving patterns, we develop an enhanced procedural geometric model capable of synthesizing natural yarn-level geometry with yarn sliding and flyaway fibers. The procedural model is driven by a specialized large lan-

guage model, WeavingLLM, which is fine-tuned on an annotated dataset of formatted weaving drafts, and prompted with domain-specific fabric expertise. Through fine-tuning and prompt tuning, WeavingLLM learns to design weaving drafts and fabric parameters from textual prompts, enabling the procedural model to produce diverse weaving patterns that stick to weaving principles. The generated macro-scale texture, along with the micro-scale geometry, can be used for fabric rendering. Consequently, our framework produces materials with significantly richer detail and realism compared to prior generative models.

1. Introduction

Woven fabric materials are commonly utilized in digital human modeling, interior design, and cinematic productions. Due to their unique weaving pattern and appearance, traditional fabric material design typically involves multiple stages, including weaving pattern creation (e.g., using tools like Substance Designer), texture authoring, and shading parameter tuning. The process is time-consuming, even for experienced artists. There is a growing demand for accessible tools that allow casual users to create photorealistic fabric materials from text, without any prior knowledge of fabrics or textiles.

* Corresponding author.

† These authors contribute equally.

¹ College of Computer Science, Nankai University, Tianjin, China

² School of Intelligence Science and Technology, Nanjing University, Suzhou, China

Diffusion models have shown impressive capabilities in generating images and even in creating materials [17, 21–23]. Recently, it has also been introduced for text/image-guided fabric material generation [4, 29]. While these models have significantly lowered the barrier to fabric material creation, the generated materials often produce artificial stripes, exhibit physically implausible patterns, or omit fabric microstructures, leading to unrealistic close-up renderings. The main challenges arise from the limited ability of diffusion models pretrained on natural images to generate microstructures, as well as the difficulty of applying constraints to the generation process.

In this paper, we introduce a novel framework, *FabricGen*, designed to generate detailed woven fabric materials from text. This framework produces highly detailed microstructures while sticking to weaving principles, without requiring complex manual design. The core of our framework lies in the decomposition of macro-scale textures and micro-scale weaving patterns. The macro-scale texture is generated by a fine-tuned diffusion model, while the micro-scale patterns are crafted by a large language model (LLM)-driven procedural geometric model.

Since macro-scale textures need to be free of microstructures, pre-trained diffusion models alone cannot adequately produce this type of fabric. To address this, we collect a dataset of microstructure-free fabrics to fine-tune a pre-trained diffusion model, allowing us to generate a macro-scale texture/albedo map. When it comes to micro-scale weaving patterns, two key challenges arise: first, existing procedural models struggle to create natural yarn-level microstructure due to missing crucial features such as random sliding and flyaway fibers. Second, even with procedural models, experienced artists are typically required to design the weaving drafts and corresponding parameters. To tackle these challenges, we first propose an advanced procedural model for woven fabric that enhances randomness and generality. Additionally, we customize a domain-specific LLM to design weaving draft and fabric parameters. The LLM is supervised fine-tuning (SFT) on a collected weaving draft dataset and further prompt-tuned with fabric expertise to learn the weaving rule and priors of woven fabrics, enabling an end-to-end generation without any manual design or prior knowledge of woven fabrics.

The albedo generated from the fine-tuned diffusion model, along with the microstructure data (e.g., normal, tangent) produced by the procedural model, can be used for fabric rendering using a fabric shading model. Consequently, our framework is capable of creating fine-grained materials with rich microstructures, guided by only text or together with images. This demonstrates a significant improvement in quality over previous generative models that were specialized for fabric materials or general materials.

To summarize, our main contributions include:

- a novel fabric generation framework—FabricGen, which decomposes macro-scale textures and micro-scale weaving patterns to enable the generation of highly detailed woven fabric materials,
- an LLM-driven procedural model for generating diverse micro-scale weaving patterns from text, along with multiply yarn geometry, yarn sliding, and flyaway fibers,
- a fine-tuned diffusion model for generating microstructure-free fabric textures from text/image.

2. Related Work

Material generation Material generation has emerged as an active research area, aiming to synthesize plausible textures from images or text prompts [1, 16, 21–23, 29, 32]. Early works primarily adopted generative adversarial networks (GANs) to model high-frequency details [1, 16, 32], e.g., TileGen [32] combined tileable StyleGAN2 variants [8] with conditional patterns to reconstruct artifact-free SVBRDF from single images under material category constraints. Recent approaches increasingly leverage diffusion models [11, 21–23, 26, 28]. ControlMat [22] formulates SVBRDF reconstruction as conditional synthesis, generating tileable high-resolution materials from uncontrolled photos. MatFuse [23] enables controllable generation and editing of 3D materials via multi-modal conditioning and latent manipulation. MaterialPicker [11] employs a Diffusion Transformer for text or photo-guided generation with distortion correction. ReflectanceFusion [26] adopts a tandem diffusion framework for text-driven editable SVBRDF. Generative neural materials [15] produce BTFs from text or images via a universal NeuMIP basis. Chord [28] utilizes two-stage diffusion and decomposition for high-quality SVBRDF generation. While these diffusion-based methods have significantly advanced general material generation, they still face common limitations: the inability to enforce domain-specific structural rules (e.g., weaving principles in fabrics), difficulty in producing realistic yarn-level microstructures, and resolution constraints that hinder fine-grained close-up rendering.

Fabric material generation Recently, there are some methods that focus on fabric generation [4, 29]. FabricDiffusion [29] formulates texture transfer as tileable material generation via diffusion, enabling distortion-free PBR textures to be generated from a single clothing image. DressCode [4] employs a fine-tuned Stable Diffusion model with three distinct decoders to synthesize tile-based PBR textures (diffuse, normal, and roughness maps) from text prompts.

While these methods are robust to surface variations and enable diverse outputs, their resolution limitations often lead to blurred details and structural artifacts. Different from the above works, we propose a new approach: the

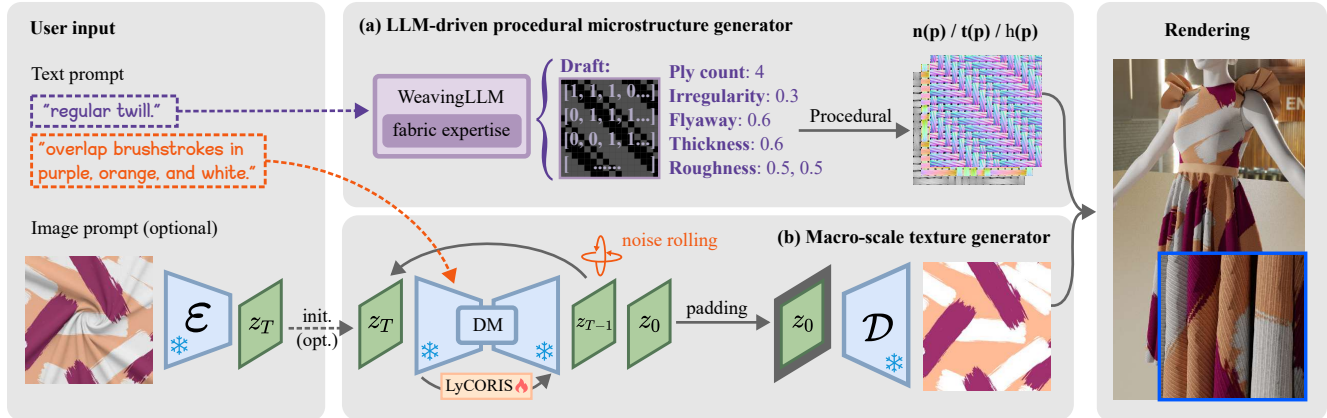


Figure 2. Overview of FabricGen. Our fabric material creation framework consists of two main components: (a) an LLM-driven procedural microstructure generator that generates yarn-level microstructures along with fabric parameters, and (b) a texture generator that generates tileable, macro-scale textures from a text prompt (and optional image). Note that the two components require distinct prompts.

weaving pattern is generated separately from the color pattern by an LLM-driven procedural model, which ensures both pattern diversity and fine fabric details. To the best of our knowledge, no existing work built for text-to-procedural microstructure generation.

Procedural fabric model Prior works have proposed various procedural geometric models for woven fabrics. One class of methods models fabrics at the yarn, ply, or even fiber level using explicit curves [10, 12, 18, 25, 31, 33]. These approaches achieve photorealistic results but rely on explicit geometric representations at the ply or even fiber level, incur computational and storage costs during both data generation and rendering. In contrast, another class of surface-based methods [6, 7] treats the fabric as a 2D sheet and procedurally generates spatially-varying normals, tangents, and heights for rendering. By avoiding explicit geometry construction and intersection tests, surface-based models are highly efficient and integrate seamlessly with traditional rendering pipelines.

However, existing surface-based models have notable limitations: they typically assume single-ply yarns and neglect global irregularities, such as yarn sliding and flyaway fibers that contribute to natural fabric irregularity. In this paper, we propose an enhanced procedural geometric model to handle these issues.

3. Background

Microstructure of woven fabric Woven fabrics are typically composed of horizontal and vertical yarns (namely, wefts and warps) interlaced in specific weaving patterns. The weaving pattern within a repeat unit can be represented as a binary matrix, referred to as a weaving draft. The diverse configurations of warp and weft yarns allow for the

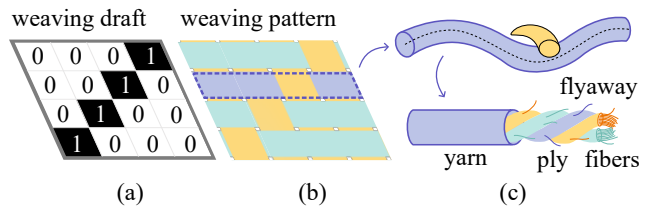


Figure 3. Woven fabrics are typically constructed by interlacing weft and warp yarns following specific patterns (b), which can be represented via a binary matrix, named weaving draft (a). Each yarn may consist of multiple plies, and each ply can be further divided into a number of fibers. Some fibers escape from the surface, resulting in irregular flyaway fibers (c).

creation of a wide variety of appearances [2], which are typically designed by skilled artists. Previous works [7, 19] typically focus on several classical patterns (twill, satin, and plain), ignoring the diversity implied in weaving patterns. In this paper, we first propose to generate weaving patterns directly from natural language descriptions.

As for yarn, each yarn may consist of a single strand or multiple strands (namely, plies), and each ply is composed of numerous fine fibers that are twisted together. Beyond these regular patterns, there are also some irregular features in the fabric, such as flyaway fibers escaping from the surface and yarns occasionally slipping out of place. We illustrate the basic composition of woven fabric in Fig. 3.

Procedural yarn representation Existing methods [6, 7] model yarns as curved cylinders, where the yarn centerline is represented as a circular arc parameterized by the maximum inclination angle u_{\max} , and the yarn cross-section is assumed to be circular, with individual fibers twisted around the yarn surface at a specified twist angle ψ .

The model analytically projects the 3D yarn geometry on the 2D UV domain. Formally, the procedural model \mathcal{F} can

be defined as a set of geometric functions over UV space:

$$\mathcal{F} = \{\mathbf{n}(\mathbf{p}), \mathbf{t}(\mathbf{p}), h(\mathbf{p})\}, \quad \mathbf{p} = (x, y) \in [0, 1]^2, \quad (1)$$

where \mathbf{p} denotes the UV coordinate, \mathbf{n} the surface normal, \mathbf{t} the tangent direction (or namely, orientation), and h the relative height field. Together, these functions define a parameterization of the yarn-level microstructure over the UV domain, enabling seamless integration with traditional rendering pipelines and allowing direct export of derived maps (e.g., normal, height, or tangent maps).

4. FabricGen: a two-scale fabric material generation framework

In this paper, we propose FabricGen, a novel fabric material generation framework, which generates high-quality fabric materials in an end-to-end manner given natural language prompts. At the core of the framework is the decoupling of the macro-scale fabric texture/albedo generation (Sec. 4.1) and the micro-scale fabric microstructure generation (Sec. 4.2), where the former is generated by a fine-tuned diffusion model and the latter is generated via an LLM-driven procedural geometric model. The diffusion model allows the diverse creation of macro-scale fabric textures, and the LLM-driven procedural model creates diverse and detailed microstructures. Finally, we employ fusion rendering using SpongeCake [24], a layered shading model. Rendering details are provided in the supplementary material. As a result, our framework produces materials that exhibit both visual diversity and fine-grained structural detail. An overview of our framework is shown in Fig. 2.

4.1. Macro-scale fabric texture generation

Previous works [4, 29] leverage diffusion models to generate fabric materials, by fine-tuning the pre-trained diffusion model with a collection of fabric materials. Their generated materials are supposed to include both the texture and the fabric microstructure together. Unfortunately, due to resolution limitations and the lack of structural constraints, the resulting materials often violate weaving rules or entirely omit microstructural details. To address these issues, we constrain the generative model to produce pure albedo maps that exclude microstructures. To this end, we collect a dataset of microstructure-free fabric textures and then fine-tune the pre-trained diffusion model with this dataset to generate albedo maps. In addition to text prompts, the generative model also supports optional fabric image conditions. Additionally, the generated materials should be seamless.

Diffusion model fine-tuning We collect a dataset of 600 microstructure-free fabric textures along with the corresponding description. Using the collected dataset, we fine-tune the FLUX.1-dev [9] model with LyCORIS [27], adapting its output to directly produce microstructure-free albedo

maps and converting a general-purpose image generator into a fabric-specific albedo generator. We incorporate noise rolling mechanism [22] to ensure tileability. Training details are shown in the supplementary.

Multimodal conditioning Our model also supports multimodal conditioning by inputting fabric images, either flat or with wrinkles. The input image is encoded into the latent space via a VAE decoder, and the resulting latent vector is used as an initialization condition for the diffusion model. This guides the generation process to suppress geometric wrinkles while preserving the original pattern style.

4.2. LLM-driven procedural microstructure generation

Besides the macro-scale texture, the key to fabrics is the microstructure, consisting of several levels (weaving pattern, yarn, ply, and fiber), as described in Sec. 3. Generating these hierarchical structures is challenging for diffusion models [4, 29] due to their limited output resolution. To this end, we resort to the procedural model for microstructure generation and leverage a LLM for weaving draft creation. Unfortunately, existing surface-based procedural approaches [6, 7] generally neglect global irregular features, and their single cylinder model does not support multi-ply configuration. To handle these issues, we propose a novel procedural geometric model with enhanced expressiveness and natural irregularity.

Procedural geometric model Our procedural model defines the yarn-level microstructures given a weaving pattern draft and a set of yarn parameters. We model plies in a yarn as curved helices, enabling multi-ply configuration, as shown in Fig.4. We also introduce some global irregular features, including yarn sliding and flyaway fibers. All these components constitute our final procedural geometric model, which produces spatially-varying geometric data, such as normal, orientation, and height (as visualized in Fig. 5). Notably, our model does not require explicit generation of these maps. Instead, it supports on-demand queries, enabling ultra-fine details without the need for pre-computing or storing high-resolution textures. Further details of our procedural model are provided in Sec. 5.

Weaving draft generation With the proposed fabric procedural model, it’s still not clear how to design the weaving pattern and geometric parameters automatically. Therefore, we propose WeavingLLM, a fine-tuned LLM that learns to generate formatted weaving drafts and corresponding parameters (e.g., roughness and ply counts) from natural language prompts, just like an experienced designer. The model is fine-tuned from the pre-trained Qwen2.5-14B-it [20] model using QLoRA [3] on a collected dataset of

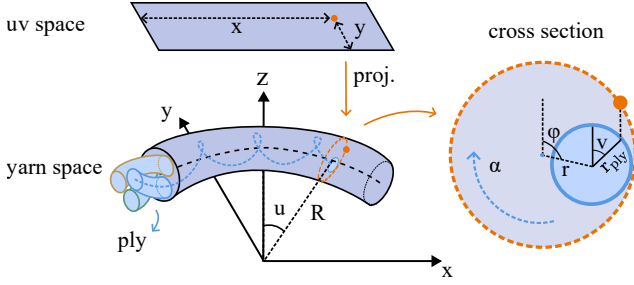


Figure 4. Curved helix model for multi-ply yarns. The model achieves analytical normal/orientation formulation, where the yarn space arc parameters u, v are linearly mapped from UV space.

1,142 annotated weaving drafts sourced from Handweaving.net [2]. All drafts are constrained to a maximum size of 16×16 . In addition to fine-tuning, we apply prompt tuning with domain-specific fabric expertise, enabling LLM to learn the prior knowledge of various types of fabrics.

Given a text prompt, WeavingLLM first generates a binary matrix as the weaving draft. The LoRA adapter is then disabled, and the base LLM predicts the fabric parameters following the fabric expertise. The weaving draft and parameters are subsequently used for procedural microstructure synthesizing and rendering. The LLM instructions, parameters list, and fine-tuning details are provided in the supplementary material.

Since we only generate the weaving pattern within a repeat unit, our model inherently adheres to fundamental weaving principles and produces rich details. The generated weaving drafts are constrained to a maximum size of 16×16 ; therefore, in theory, a loom with 16 shafts and treadles can physically realize any output of our model.

5. Procedural geometric model

In this section, we introduce our procedural geometric model with several novel components. We first introduce the curved helix model for multi-ply yarns. Then, we discuss how to model natural irregularities.

5.1. Curved helix model

Previous works [6, 7] model yarns as curved cylinders, which only support single-strand yarn configuration. Wu et al. [25] model plies as helix curves. However, it requires explicit storage and rendering of each curve. Instead, we propose a procedural curved helix yarn model to map ply-level geometry directly from UV coordinates, depicted in Fig. 4. We model yarn centerline as a circular arc, while individual plies form helices around the yarn centerline with rotation speed α . We parameterize the plies through yarn space arc coordinates u and v , which are linearly projected from surface UV coordinates x, y . Therefore, we derive the analytical normal \mathbf{n} , orientation \mathbf{t} and height h :

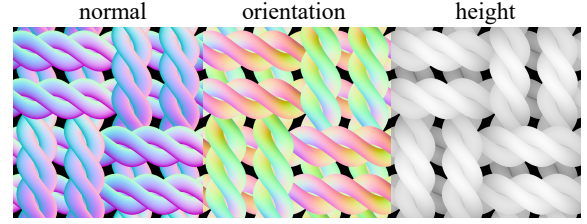


Figure 5. Visualization of the normal, orientation and height maps of a 3-ply basket weave. The parameters of this case are provided in the supplementary material.

$$\mathbf{n}(u, v) = \begin{bmatrix} \sin(u) \cos(v) \\ \sin(v) \\ \cos(u) \cos(v) \end{bmatrix}, \quad (2)$$

$$\mathbf{o}_{\text{ply}}(u) = \begin{bmatrix} \cos(u) \\ -r\alpha \cos(\varphi(u)) \\ -\sin(u) - r\alpha \sin(\varphi(u)) \end{bmatrix}, \quad (3)$$

$$\mathbf{t}(u, v) = \text{rotate}(\mathbf{o}_{\text{ply}}, \mathbf{n}, \psi) \quad (4)$$

$$h(u, v) = r \cos(\varphi(u)) + \cos(u)(R + r_{\text{ply}} \cos(v)), \quad (5)$$

$$\varphi(u) = \varphi(0) + uR\alpha. \quad (6)$$

where R is the radius of the yarn arc, r_{ply} is the ply radius, r is the distance from the ply center to the yarn center, $\varphi(u)$ indicates rotational phase of ply along the axis, $\varphi(0)$ is the initial phase of the ply. The fiber orientation \mathbf{t} comes from rotating the ply orientation \mathbf{o}_{ply} around the normal at a certain twist angle ψ . An visualization of these geometric functions is shown in Fig. 5. More derivation and configuration details are provided in the supplementary material.

5.2. Global irregular effects

To model natural irregularities of woven fabric, existing methods [6, 7] apply perturbations to the specular reflectance or the height field, thereby achieving a macroscopic irregular appearance. However, they neglect essential global irregular effects, such as yarns sliding out of place and fibers escaping from the surface. In this section, we discuss how to simulate these phenomena.

Yarn sliding The yarn arrangement in real fabrics is not always regular but exhibits inherent positional randomness, manifesting as yarn sliding. Thus, we model this effect by perturbing yarn positions using continuous procedural noise. Given a UV point $\mathbf{p} = (x, y)$ following the coordinate system in Fig. 4, we introduce axial (x -direction) 1D Perlin noise $P(x)$ on the yarn, then apply radial (y -direction, $y \in (0, 1)$) perturbations through:

$$y_s = 0.5 + (y - 0.5)(1 - k_{\text{sliding}}|P(x)|), \quad (7)$$

$$y_r = y_s^{e^{k_{\text{sliding}}P(x)}}, \quad (8)$$

where y_s denotes the scaled position, y_r is the final perturbed coordinate, and k_{sliding} is a user-defined parameter



Figure 6. **Comparison on text conditioning.** Given text prompt, DressCode [4] and MatFuse [23] generate PBR textures, which struggle to capture fine fabric details, resulting in blurring and distortion in close-up views. In contrast, our method generates procedural microstructure together with macroscale texture, ensuring a realistic appearance in both macro and close-up views.

Table 1. Quantitative comparisons between our method, MatFuse [23], and DressCode [4].

	MatFuse	DressCode	Ours
Text conditioning			
CLIP Score \uparrow	0.240	0.307	0.317
User study \uparrow	1.43%	16.02%	82.55%
Text and image conditioning			
CLIP-I Score \uparrow	0.722	N/A	0.827

that controls the sliding strength. The bijective mapping $f : y \rightarrow y_r$ enables practical implementation through inverse queries to retrieve regular coordinates from irregular space. This sliding mechanism reveals otherwise occluded lower-layer yarns, with visual plausibility guaranteed by our procedural model.

Flyaway fibers Flyaway refers to fibers escaping from the fabric surface. To model this, we propose to introduce an additional fiber layer with stochastic fiber orientations in the SpongeCake model, where the 3D fiber orientation field $\mathbf{o}_{\text{flyaway}}(\mathbf{p})$ is constructed from two 2D Perlin noise N_1, N_2 , where N_1 controls the fiber position and horizontal orientation, and N_2 controls the vertical orientation. Though carefully design, our stochastic orientation field exhibits a continuous fiber-like distribution. Detailed configuration is provided in the supplementary material.

6. Results

In this section, we first present our fabric material generation results and compare them with two representative text-to-material methods: DressCode [4], which targets fabric material generation, and MatFuse [23], which focuses on general materials. Then we present our results with image conditioning, and compare them with MatFuse. Finally, we perform ablation studies on several key components of our pipeline. All training, inference, and rendering are executed and evaluated on an NVIDIA RTX 4090 GPU.

6.1. Fabric material generation

Text conditioning We compare our method with DressCode and MatFuse in Fig. 6. Given text prompts, both prior methods generate PBR texture maps, while ours produces macro-scale textures and micro-scale weaving patterns (visualized as normal, orientation, and height map). By rendering the generated materials on a cloth mesh, existing methods yield plausible results at a distance but suffer from blurring and distortion under close-up inspection. In contrast, our method faithfully preserves fine-scale fabric details, delivering high visual fidelity at both the macro and micro levels. We quantitatively evaluate our results using the CLIP Score [5] and a user study. The CLIP Score is computed as the average over 100 generated materials from diverse text prompts, measured between each prompt and plane rendering. The user study includes 12 cases, in which 64 participants are asked to choose the result that best matches the prompt and appears most realistic. As shown in Tab 1,

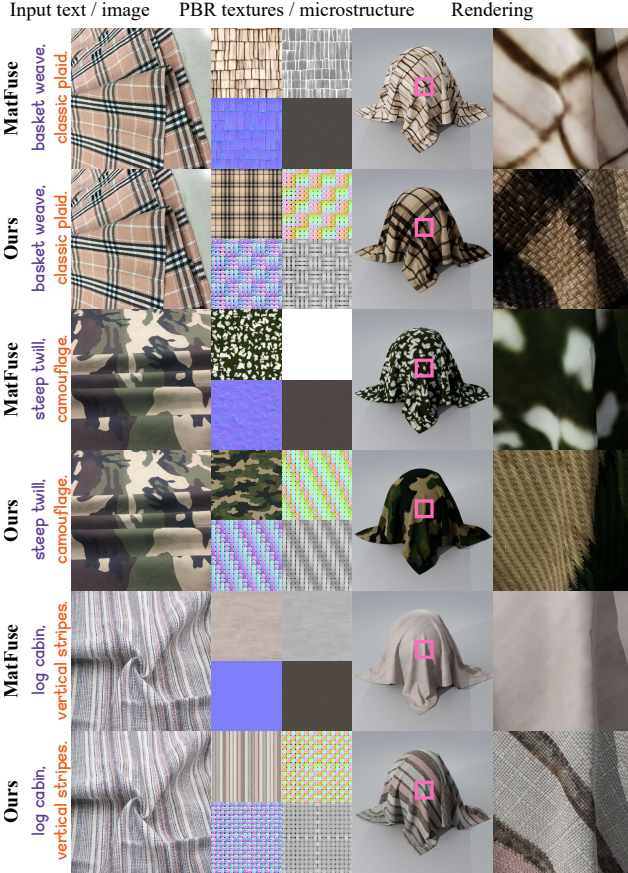


Figure 7. **Comparison on text and image conditioning.** Given both text and image prompts, our generated materials achieving better consistency and richer detail than MatFuse [17].

our method achieves higher CLIP Scores and receives better user preference, demonstrating better semantic alignment, perceptual quality, and visual realism. More results covering a wide range of patterns and examples of our user study are provided in the supplementary material.

Image conditioning In addition to text-based generation, our method also supports additional image conditioning, providing enhanced controllability over the generated materials. We compare our method with MatFuse [23] on the same text and image conditions, as shown in Fig. 7. To evaluate the results, we compute the average CLIP-I Score [14] between the rendering result and input image, as shown in Tab. 1. The results demonstrate that our method matches better with the input image. Although FabricDiffusion [29] is not included since it is not fully open-sourced, it shares a similar scheme (Diffusion-based generation) and limitation (lack of yarn-level details) with MatFuse. More results are provided in the supplementary material.

6.2. Ablation studies

Impact of WeavingLLM We first validate the necessity of fine-tuning a LLM in Fig. 8, by comparing WeavingLLM

	Qwen2.5-14B-it	GPT-5	Ours	Reference
spot bronson, lace				
	COSSIM↑: 0.35 PREF↑: 1.7%	COSSIM↑: 0.726 PREF↑: 18.6%	COSSIM↑: 0.912 PREF↑: 79.7%	
	LPIPS↓: 0.385	LPIPS↓: 0.19	LPIPS↓: 0.152	
herringbone twill				
	COSSIM↑: 0.583 PREF↑: 16.9%	COSSIM↑: 0.582 PREF↑: 11.9%	COSSIM↑: 0.701 PREF↑: 71.2%	
	LPIPS↓: 0.183	LPIPS↓: 0.245	LPIPS↓: 0.125	

Figure 8. **Comparison among WeavingLLM, Qwen2.5 [20], and GPT-5 [13].** COSSIM indicates the cosine similarity of Fourier spectra, and PERF indicates the user preferences ratio. All renderings use a pure gray albedo map. More reference drafts are provided in the supplementary material.

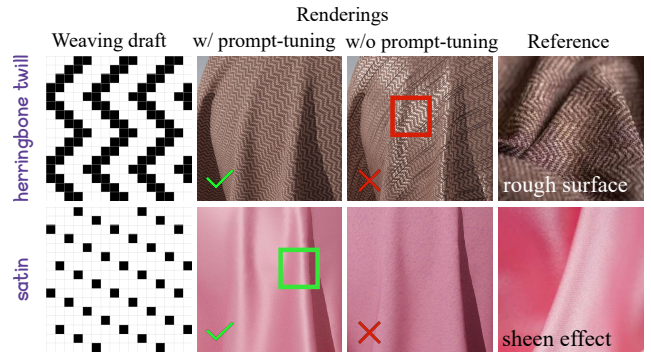


Figure 9. **Impact of prompt-tuning.** The parameters predicted by WeavingLLM successfully reproduce the rough appearance of herringbone twill and the anisotropic sheen of satin, whereas parameters predicted by the base LLM fail to reproduce this effect.

and our base model, Qwen2.5 [20], and GPT-5 [13]. To quantitatively evaluate the quality of the generated weaving pattern, we find several reference drafts in the dataset that match the description, and then compute the average cosine similarity of the Fourier spectra between the generated draft and all reference drafts. In addition, we conduct a user study, inviting 60 participants to select the best-matching pattern based on the given prompt and reference examples. Moreover, we compute the learned perceptual image patch similarity (LPIPS) [30] between the rendering of generated draft and reference draft. The results demonstrate that the drafts generated by WeavingLLM match better with the ref-

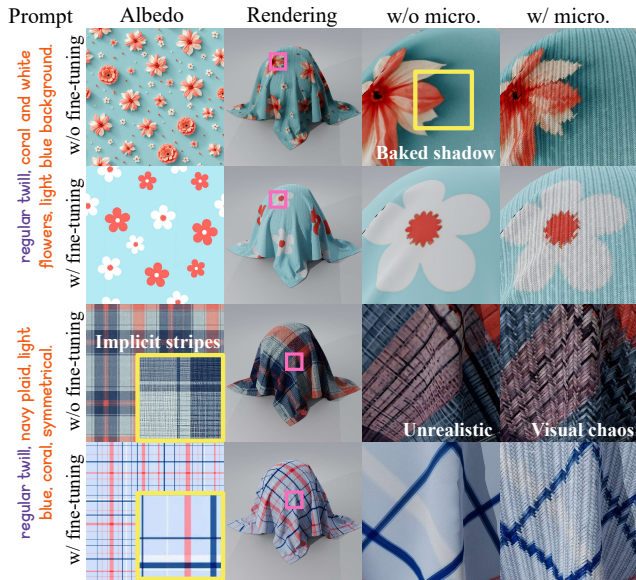


Figure 10. **Impact of macro-scale texture generator.** The diffusion model pre-trained on natural images occasionally produces unintended 3D artifacts (first row) or implicit stripes (third row), whereas our fine-tuned model is capable of generating microstructure-free albedo maps. The results in the fourth column use the regular twill pattern in Fig 2.

Ours	Ours	Jin et al.	Captured photo
w/o irregularity	w/ irregularity		
w/o sliding	w/ sliding		
LPIPS↓: 0.352	LPIPS↓: 0.310	LPIPS↓: 0.464	
w/o flyaway	w/ flyaway		
LPIPS↓: 0.499	LPIPS↓: 0.354	LPIPS↓: 0.471	

Figure 11. **Impact of irregular effects.** Irregular yarn displacements in real fabrics often result in strip-like gaps between yarns, and fibers escaping from the surface cause irregular highlights.

ferences, both in terms of quantitative analysis and user perception. More results cover a wide range of patterns are provided in the supplementary material.

Furthermore, we illustrate the necessity of prompt-tuning the LLM in Fig. 9. Since each type of pattern has a specific characteristic, parameters generated by an LLM without domain-specific priors may lead to an unrealistic appearance that contradicts daily experience. In contrast, with the given fabric expertise and priors, WeavingLLM is able to predict reasonable parameters according to the pattern characteristics.

Impact of macro-scale texture generator We illustrate the necessity of fine-tuning the diffusion model in Fig. 10. The original FLUX model occasionally generates 3D objects or implicit stripes, even if we have already instructed the model to output “no microstructure, no folds, no wrinkles” textures. These implicit details result in baked shadows and visually chaotic appearances. In contrast, our fine-tuned model produces microstructure-free fabric textures, along with the procedural microstructure, yielding cleaner and more stable rendering results.

Impact of our irregular effects We validate the influence of yarn sliding and flyaway in Fig. 11. By introducing yarn sliding effect, our rendering matches the strip-like gaps of real fabric. By introducing flyaway fibers, we reproduce the stochastic specular highlights captured in real photo. In contrast, the previous method [7] only applies simple perturbations on the color and height field, which are insufficient to capture these complex effects. To quantitatively validate our method, we compute the LPIPS between the renderings and captured photos, demonstrating that our method matches better with the irregularity of real fabrics.

6.3. Discussion and limitations

Our approach currently focuses on printed woven fabrics. However, there exist other types of fabrics, such as knitted fabrics, plastisol print, jacquard, and embroidery. Extending support to more diverse types is left as future work.

7. Conclusion

We introduce FabricGen, an end-to-end framework for generating high-fidelity woven fabric materials from text/image prompts. By decoupling macro-scale texture synthesis from micro-scale geometry modeling, our method produces photorealistic and semantically accurate fabric materials. At the macro level, we fine-tune a diffusion model to produce seamless, microstructure-free textures. At the micro level, a procedural geometry model guided by WeavingLLM, a domain-specific language model, synthesizes detailed yarn-level microstructures with natural irregularities. Our results demonstrate significantly improved detail, realism, and semantic alignment compared to state-of-the-art baselines, making it suitable for digital human clothing, interior design, and other applications. Quantitative metrics and user studies confirm the effectiveness of our approach. Extending our method to a broader range of textile types remains an open direction for future work.

Acknowledgment

We thank the reviewers for the valuable comments. This work has been partially supported by the National Natural Science Foundation of China under grant No. 62572230.

References

- [1] Urs Bergmann, Nikolay Jetchev, and Roland Vollgraf. Learning Texture Manifolds with the Periodic Spatial GAN, 2017. 2
- [2] Kris Bruland. Handweaving.net, 2025. <https://handweaving.net/>. 3, 5
- [3] Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. Qlora: Efficient finetuning of quantized llms, 2023. 4
- [4] Kai He, Kaixin Yao, Qixuan Zhang, Jingyi Yu, Lingjie Liu, and Lan Xu. Dresscode: Autoregressively sewing and generating garments from text guidance. *ACM Transactions on Graphics (TOG)*, 43(4):1–13, 2024. 2, 4, 6
- [5] Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. CLIPScore: A Reference-free Evaluation Metric for Image Captioning, 2022. 6
- [6] Piti Irawan and Steve Marschner. Specular reflection from woven cloth. *ACM Trans. Graph.*, 31(1):1–20, 2012. 3, 4, 5
- [7] Wenhua Jin, Beibei Wang, Miloš Hašan, Yu Guo, Steve Marschner, and Ling-Qi Yan. Woven fabric capture from a single photo. In *Proceedings of SIGGRAPH Asia 2022*, 2022. 3, 4, 5, 8
- [8] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and Improving the Image Quality of StyleGAN, 2020. 2
- [9] Black Forest Labs. Flux. <https://github.com/black-forest-labs/flux>, 2024. 4
- [10] Zixuan Li, Pengfei Shen, Hanxiao Sun, Zibo Zhang, Yu Guo, Ligang Liu, Ling-Qi Yan, Steve Marschner, Milos Hasan, and Beibei Wang. Fiber-level woven fabric capture from a single photo, 2024. 3
- [11] Xiaohe Ma, Valentin Deschaintre, Miloš Hašan, Fujun Luan, Kun Zhou, Hongzhi Wu, and Yiwei Hu. Materialpicker: Multi-modal dit-based material generation. *ACM Transactions on Graphics (TOG)*, 44(4):1–12, 2025. 2
- [12] Zahra Montazeri, Søren B. Gammelmark, Shuang Zhao, and Henrik Wann Jensen. A practical ply-based appearance model of woven fabrics. *ACM Trans. Graph.*, 39(6), 2020. 3
- [13] OpenAI. Introducing GPT-5. <https://openai.com/index/introducing-gpt-5/>, 2025. Accessed: YYYY-MM-DD. 7
- [14] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, 2021. 7
- [15] Nithin Raghavan, Krishna Mullia, Alexander Trevithick, Fujun Luan, Miloš Hašan, and Ravi Ramamoorthi. Generative neural materials. In *Proceedings of the Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers*, New York, NY, USA, 2025. Association for Computing Machinery. 2
- [16] Carlos Rodriguez-Pardo and Elena Garces. SeamlessGAN: Self-Supervised Synthesis of Tileable Texture Maps. *IEEE Transactions on Visualization and Computer Graphics*, 29(6):2914–2925, 2023. 2
- [17] Sam Sartor and Pieter Peers. MatFusion: A Generative Diffusion Model for SVBRDF Capture. In *SIGGRAPH Asia 2023 Conference Papers*, pages 1–10, 2023. 2, 7
- [18] Kai Schröder, Arno Zinke, and Reinhard Klein. Image-based reverse engineering and visual prototyping of woven cloth. *IEEE Transactions on Visualization and Computer Graphics*, 21(2):188–200, 2015. 3
- [19] Yingjie Tang, Zixuan Li, Milos Hasan, Jian Yang, and Beibei Wang. Woven fabric capture with a reflection-transmission photo pair. In *ACM SIGGRAPH 2024 Conference Papers*, New York, NY, USA, 2024. Association for Computing Machinery. 3
- [20] Qwen Team. Qwen2.5: A party of foundation models, 2024. 4, 7
- [21] Giuseppe Vecchio. Stablematerials: Enhancing diversity in material generation via semi-supervised learning. *arXiv preprint arXiv:2406.09293*, 2024. 2
- [22] Giuseppe Vecchio, Rosalie Martin, Arthur Roullier, Adrien Kaiser, Romain Rouffet, Valentin Deschaintre, and Tamy Boubekeur. Controlmat: a controlled generative approach to material capture. *ACM Transactions on Graphics*, 43(5):1–17, 2024. 2, 4
- [23] Giuseppe Vecchio, Renato Sortino, Simone Palazzo, and Concetto Spampinato. MatFuse: Controllable Material Generation with Diffusion Models. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4429–4438, 2024. 2, 6, 7
- [24] Beibei Wang, Wenhua Jin, Miloš Hašan, and Ling-Qi Yan. Spongecake: A layered microflake surface appearance model. *ACM Transactions on Graphics*, 42(1), 2022. 4
- [25] Kui Wu and Cem Yuksel. Real-time fiber-level cloth rendering. In *Proceedings of the 21st ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*, New York, NY, USA, 2017. Association for Computing Machinery. 3, 5
- [26] Bowen Xue, Claudio Guarnera, Shuang Zhao, and Zahra Montazeri. Reflectancefusion: Diffusion-based text to svbrdf generation. In *Eurographics Symposium on Rendering*. Eurographics Association, 2024. 2
- [27] Shih-Ying Yeh, Yu-Guan Hsieh, Zhidong Gao, Bernard BW Yang, Giyeong Oh, and Yanmin Gong. Navigating text-to-image customization: From lycoris fine-tuning to model evaluation. In *The Twelfth International Conference on Learning Representations*, 2023. 4
- [28] Zhi Ying, Boxiang Rong, Jingyu Wang, and Maoyuan Xu. Chord: Chain of rendering decomposition for pbr material estimation from generated texture images. *arXiv preprint arXiv:2509.09952*, 2025. 2
- [29] Cheng Zhang, Yuanhao Wang, Francisco Vicente Carrasco, Chenglei Wu, Jinlong Yang, Thabo Beeler, and Fernando De la Torre. FabricDiffusion: High-fidelity texture transfer for 3d garments generation from in-the-wild images. In *ACM SIGGRAPH Asia*, 2024. 2, 4, 7
- [30] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018. 7
- [31] Shuang Zhao, Fujun Luan, and Kavita Bala. Fitting procedural yarn models for realistic cloth rendering. 35(4), 2016. 3

- [32] Xilong Zhou, Miloš Hašan, Valentin Deschaintre, Paul Guerrero, Kalyan Sunkavalli, and Nima Kalantari. TileGen: Tileable, Controllable Material Generation and Capture, 2022. [2](#)
- [33] Junqiu Zhu, Zahra Montazeri, Julien Aubry, Ling-Qi Yan, and Andrea Weidlich. A practical and hierarchical yarn-based shading model for cloth. *Computer Graphics Forum*, 42(4):2–11, 2023. [3](#)