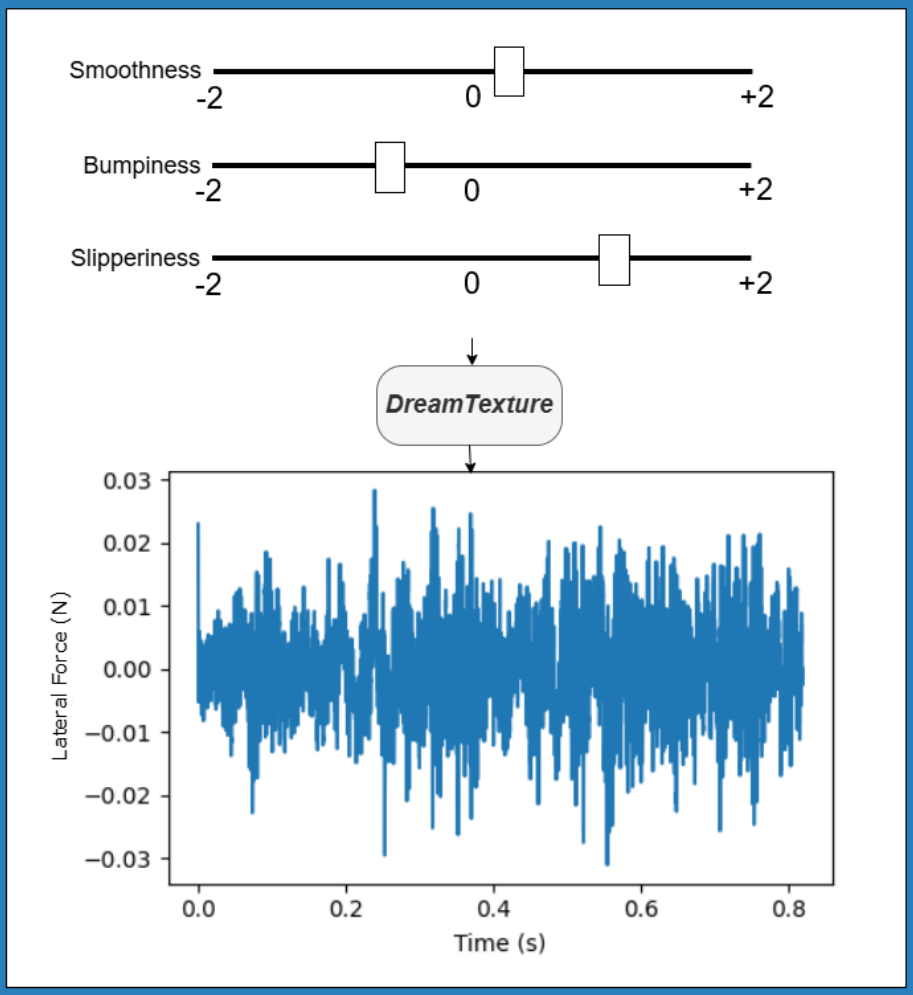
DreamTexture: Latent Diffusion Model for Psychophysical Feature-to-Texture Generation

Quinn Begelinger

Delft University of Technology



TU Delft

DreamTexture: Latent Diffusion Model for Psychophysical Feature-to-Texture Generation

Master's Thesis Report

by

Quinn Begelinger

To obtain the degree of:
Master of Science in Robotics
at the Delft University of Technology.
To be defended publicly on:
Thursday July 3, 2025 at 14:00 PM

Student Number:	4962338	
Supervisor:	Dr. Y. Vardar	
Project Duration:	October 2024 - June 2025	
Thesis Committee:	Dr. Y. Vardar,	TU Delft, Chair and Supervisor
	Dr. C. Pek	TU Delft, External member

TU Delft

Preface

The closest people to me know that when I enjoy doing something, I pour my heart and soul into doing it in a way that I can be proud of. A project of this scale was new to me, and the challenge of staying engaged would have been much more difficult if not for the people around me, who inspired and motivated me throughout. Even when nothing seemed to go to plan, I could always count on you.

First of all, a heartfelt thank you to Yasemin, who was understanding throughout the entire process and offered the crucial pieces of feedback at the moments I needed it. Thank you for helping me stay on track and being such a reliable source of common sense, expertise, and reassurance.

Secondly, thank you to my friends with whom I've shared laughter, tears, and countless hours in and out of university: Ruben, Laura, Roel, Per, Thijn, and Paula. Without you all, I may have gone mad. Here's to many more years of friendship.

Third, I want to thank my girlfriend and my family. Your love and support kept me grounded and confident through the hardest parts. Thank you for always believing in me. You are my heroes.

Finally, special thanks to Jagan, Maxime, Koen, Tim, my high school friends, and the other members of HITLab, all of whom helped in unique and meaningful ways.

Quinn Begelinger
Delft, June 2025

Contents

1 Paper **1**

A Detailed Model Architecture **18**

B (Hyper)parameter selection **20**

C Distribution of SENS3 Database **21**

D More Inference Results **24**

References **27**

1

Paper

DreamTexture: Latent Diffusion Model for Psychophysical Feature-to-Texture Generation

Quinn Begelinger (4962338)*
TU Delft
Delft, Netherlands
qbegelinger@tudelft.nl
*Supervisor: Dr. Yasemin Vardar, TU Delft

***Abstract*—Generative AI has revolutionized domains such as language, vision, and audio; yet, its application to the field of haptics, specifically signals for friction modulation devices, remains barely explored. A generative model could alleviate the issues associated with recording friction-based texture signals, such as the expenses of recording equipment and the limitation to lab environments, which significantly constrain the diversity of texture signals that can be rendered on friction modulation haptic devices. We propose a generative latent diffusion model called *DreamTexture*. The model is conditioned on a feature vector derived from a psychophysical perceptual space, where each dimension corresponds to an adjective pair (e.g., Rough–Smooth, Sticky–Slippery). We investigate whether *DreamTexture* can synthesize friction signals that align with users' perceptual expectations, despite the subjective nature of tactile experiences, influenced by individual skin properties and linguistic interpretation. Moreover, *DreamTexture* is optimized for real-time inference on commercially available hardware, making haptic content creation more scalable and accessible. Our findings indicate that the diffusion process lends itself well to the efficient generation of one-dimensional friction signals and produces realistic signals, but it exhibits limitations in fully capturing the variability inherent in the input space.**

I. Introduction

In the modern digital landscape, screens and speakers dominate our interactions, providing rich visual and auditory experiences. Yet, the rich and nuanced sense of touch, an important aspect of human perception, remains largely unaddressed in most digital media. However, research has shown that incorporating haptic feedback into media significantly enhances users' emotional responses, leading to stronger engagement, higher product ratings, and more memorable experiences [1].

Consequently, physical haptic devices are a popular subject of research [2], [3], [4]. Friction modulation devices, for example, the electrovibration display [5], already achieve effective rendering and have potential for widespread integration in products such as smartwatches and smartphones. However, acquiring the proper input signals is not trivial, often demanding complex laboratory equipment and specialized expertise to record finger-surface interactions. This limitation currently impedes the widespread adoption of such devices. In this work, we propose an alternative approach: leveraging diffusion models to *generate* friction signals. Inspired by their success in synthesizing high-dimensional data across vision and audio domains, we explore their capacity to synthesize realistic texture signals for haptics. We aim to validate this research direction and identify the core challenges in applying generative models to haptic signal generation.

Designing tactile feedback is inherently complex, as it depends on how humans perceive textures through touch. When a finger slides across a surface, the tactile experience arises from mechanical cues such as softness, thermal properties, and, central to this study, surface texture. Micro- and mesoscale variations cause the skin to deform, leading to lateral friction forces that vary depending on the surface and interaction parameters. This interaction can be modeled as a one-dimensional temporal signal (friction over time), but the resulting signal is highly sensitive to applied pressure and finger velocity [6].

The lateral friction force during touch can be described as:

$$F_l = \mu F_n \tag{1}$$

where F_l is the lateral force, μ the friction coefficient, and F_n the normal (downward) force. Importantly, μ varies with surface properties, and as faster movement results in sharper changes, making velocity a critical parameter alongside F_n for determining F_l when capturing or generating texture signals.

As a result, recording haptic textures requires accurate tracking of both the lateral force *and* the interaction dynamics over time. This necessitates specialized lab equipment, as used in the SENS3 dataset [7], and makes it difficult to collect data from immovable surfaces or through non-expert participants, significantly limiting scalability and accessibility.

We are not the first to employ generative AI models to alleviate this issue; prior work has explored the use of Variational Autoencoders (VAEs) [8] and Generative Adversarial Networks (GANs) [9] to synthesize haptic signals. However, diffusion models have recently demonstrated superior performance across various domains, particularly in generating high-quality and diverse samples of images [10] and music [11]. Despite this, they remain underexplored in the domain of haptic signal generation. Moreover, existing models typically rely on image inputs (photographs of surfaces) for texture generation [7], [12], [13]. While intuitive, this has two key limitations: image data does not reliably contain textural information, and it complicates the generation of novel or fictional textures that lack visual references.

Instead, we propose conditioning on perceptual ratings from the SENS3 dataset [7], which contains both high-quality friction signals and subjective ratings of 50 real textures. These ratings were collected from multiple participants using adjective-based scales like Rough–Smooth, Flat–Bumpy, and Sticky–Slippery. Compared to images, these descriptors provide a more direct and human-centered way to represent texture.

By using these perceptual ratings as conditioning inputs, we allow users to generate texture signals based on how they want the surface to feel, rather than how it looks. This approach makes it easier to create both realistic and imaginary textures, and it opens the door to more accessible, scalable tools for haptic content creation without requiring complex lab setups or expert hardware.

To implement this, we introduce *DreamTexture*, a psychophysical feature-to-texture diffusion model that synthesizes tactile signals using a two-stage latent diffusion framework [14], [15]. The first stage generates a compressed latent representation of the texture, while the second stage reconstructs the final friction signal conditioned on this latent representation. This architecture splits the computational load, enabling real-time inference on consumer-grade hardware.

Our model achieves a 512× compression ratio, allowing us to generate 1.64-second texture signal fragments in 10 diffusion steps, completing in 0.75 seconds. Using v-objective diffusion, we attain peak performance with an RMSE of $0.0051N$. However, we find that the perceptual conditioning has limited influence on the generated outputs, likely due to contradictory or noisy data in the dataset.

Our contributions are as follows:

1) We propose the first psychophysical feature-to-texture diffusion model, incorporating a two-stage latent architecture for scalable training and inference.
2) We demonstrate real-time generation on commercially available hardware with only 12 hours of training time.
3) We surpass state-of-the-art image-conditioned GAN models in RMSE-based signal reconstruction.
4) We identify critical limitations in current haptic datasets that impede effective conditioning on perceptual ratings.

II. Background

Diffusion models are a relatively recent development in generative AI, first introduced in 2015 [16] and gaining prominence after critical breakthroughs in 2020 [17]. Diffusion models operate using two processes that are inverses of each other: a "forward" diffusion process and a "reverse" diffusion process. The forward process systematically destroys the structure of input data by gradually adding noise in small steps. The reverse process then seeks to restore the structure, starting from a noise distribution and subtracting fractions of noise iteratively. This iterative approach makes the reverse process flexible and computationally feasible, resulting in enhanced performance.

The forward process is defined by a function $q(x_t|x_{t-1})$, which takes input data x and adds noise at each step. It is defined as:

$$q(x_t|x_{t-1}) = \mathcal{N}(x_t; \sqrt{1-\beta_t}x_{t-1}, \beta_t\mathbf{I}), \tag{2}$$

Where β_t controls the variance of the added noise.

The reverse process is not analytically tractable, so a neural network is trained to approximate the denoising steps by predicting the added noise. The goal is to minimize the loss function $-\log(p_\theta(x_0))$, which maximizes the likelihood of generating the original sample x_0. Since x_0 is dependent on all states of x, we can not calculate this loss function directly and the model predicts the noise $\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)$ added in the forward process, from which the mean of the reverse distribution is derived as:

$$\boldsymbol{\mu}_\theta\left(\mathbf{x}_t, t\right) = \frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}_\theta\left(\mathbf{x}_t, t\right)\right) \tag{3}$$

Where $\alpha_t = \beta_t - 1$. During training, the neural network is tasked with predicting the noise present in each noisy data version, with the loss function being the Root Mean Square Error (RMSE) between the predicted noise and the actual noise added in the forward process.

Some recent improvements, such as v-objective training and DDIM sampling [18], enable faster inference and better stability. These techniques allow diffusion models to approach high generation quality in significantly fewer steps, critical for real-time applications.

With DreamTexture, we aim to use the powerful iterative generation of this technology for texture signal synthesis. We extend its existing architecture by introducing an efficient, custom training and inference procedure, along with a novel conditioning mechanism, to fully realize its potential.

III. Methods

A. Database Preparation

For the training of our model, we use the SENS3 Dataset [7]. This dataset consists of several different types of recorded interactions with 50 different textures. The textures are chosen to cover a broad range of tactile sensations, and can be divided into 10 categories. The categories are not balanced, with "Fabric" having 12 different textures (the most), while "Rubber" only has 1 texture (the least).

We use two subsets of the SENS3 dataset [7] to train the DreamTexture architecture.

1) Friction Signals: The first subset consists of finger-on-texture sliding interaction data recorded during free exploration of various surfaces by 2 participants. This includes recordings of the lateral friction forces, as well as the normal force and finger velocity at all times. Not all recordings are of equal length, varying between 2 to 6 minutes per texture, producing a potential data imbalance.

We apply a bandpass Butterworth filter [19], removing all frequencies below 20 Hz that fall in the same range as the finger's motions and above 1 kHz that are outside of the perceivable range [20]. Figure 1 shows an example texture from

the database and its spectrogram before and after filtering. The removal of low-frequency components significantly changes the signal's shape and amplitude, indicating that a significant portion of the original signal was attributable to finger motion. After filtering, the signal is centered around 0 N, rather than exhibiting a constant positive offset.

Due to the unconstrained nature of the recordings, both normal force and finger velocity vary widely across each session. Ideally, these parameters would be provided as conditioning inputs during training, enabling the model to generate context-aware signals. However, due to the limited size of the dataset, we omit this conditioning, which could result in generations that do not reflect specific force or velocity profiles and therefore have unpredictable sensations.

To alleviate this, the training data was filtered based on specified speed and force ranges. As a result, users must interact with the output signal within the same range, so the values that were most comfortable for users to maintain during interactions were chosen. A force range of 0.4–0.6 N and a speed range of 66–99 mm/s were chosen, based on [21]. A segmentation algorithm was designed to obtain a subset of the data that includes only data recorded within the specified ranges. The algorithm is defined as follows:

$$\begin{aligned} x = \bigoplus_{i=1}^{N} x_i, \quad \text{where } \forall i \in \{1,\dots,N\},\ \forall t \in [t_i, T_i]: \\ v(t) \in [v_{\min}, v_{\max}],\ f(t) \in [f_{\min}, f_{\max}], t_{i+1} - T_i < \delta \\ T_N - t_0 > \psi \end{aligned} \tag{4}$$

Here, x is the concatenated output signal composed of valid segments x_i, each spanning a time interval $[t_i, T_i]$. The velocity $v(t)$ and force $f(t)$ at every moment within each segment must lie within the specified bounds. The parameter $\delta = 0.1\,\text{s}$ allows for short interruptions where the values may briefly fall outside the target ranges, maintaining continuity. The total duration of the final signal must exceed a minimum length $\psi = 1.64\,\text{s}$ to ensure that each segment contains at least one valid training sample.

Choosing $\delta = 0.1\,\text{s}$ preserves enough samples while maintaining high data quality. Larger values of δ run the risk of introducing inconsistent data, while smaller values reduce the dataset size to a point where training becomes ineffective.

Finally, the extracted data segments are divided into shorter sections matching the target signal length used during training.

The start time of signals within each texture recording is arbitrary because it was recorded as one long interaction; therefore, we can apply a 50% overlap between consecutive segments along the time axis. This overlap increases the effective size of the dataset and ensures that temporal features that might otherwise be truncated at the edges of a segment are captured more fully, appearing centrally in other segments. After slicing and overlapping, we obtain approximately 55,000 texture samples in the unfiltered dataset and 22,000 in the filtered dataset for training.

2) Psychophysical ratings: The second subset consists of subjective ratings given on 8 different axes by 12 different participants for each of the 50 textures. Each axis is represented and rated independently using a 9-point scale (0–8).

To reduce input dimensionality and avoid sparsity in the psychophysical space, we selected a minimal set of adjectives that efficiently describe textures without introducing redundancy. Prior studies have shown that increasing the number of adjectives only marginally improves the explained variance in texture perception, as many descriptors are used interchangeably by participants, even if not strictly synonymous. For example, "jagged," "sharp," and "hard" were found to provide little additional information beyond "rough," while "wet," "damp," and "greasy" often overlapped with "slippery" [22]. Based on these insights, we selected the following six perceptual axes as essential for spanning the texture space:

- Roughness
- Hardness
- Warmness
- Slipperiness
- Bumpiness
- Evenness

Since most friction-modulation devices cannot render thermal or compliance-related properties, we discard *warmness* and *hardness*. *Evenness* is discarded because we believed it to be the most ambiguous, and we retain **roughness**, **slipperiness**, and **bumpiness** as our final set of dimensions. This reduces the input space to a manageable three-dimensional psychophysical representation.

The validity of this reduced axis set is supported by [23], which demonstrated that novel textures can be synthesized by linearly interpolating between textures that are close in a similar perceptual space.

The rating vectors were collected from 12 participants, one for each texture. During the training, when we associate texture ratings with texture signals, ratings are reused across all corresponding texture signal segments, introducing a potential bottleneck due to limited variation in conditioning data.

Moreover, since participants may interpret the 0–8 scale differently (i.e., one participant's "4" may reflect a different intensity than another's), we normalize each participant's ratings independently to follow a standard normal distribution, across each of the rating dimensions separately:

$$X \sim \mathcal{N}(0, 1)$$

This normalization ensures that each participant contributes equally to the learning process, mitigating bias from individual rating tendencies. As a result, the average rating becomes 0, and most values fall within the range of -2 to 2. We calculate the average normalized ratings for all three dimensions for each texture to visualize the average position of each texture in perceptual space in Figure 2.

B. Designing DreamTexture

DreamTexture is comprised of two diffusion models working in series with each other:

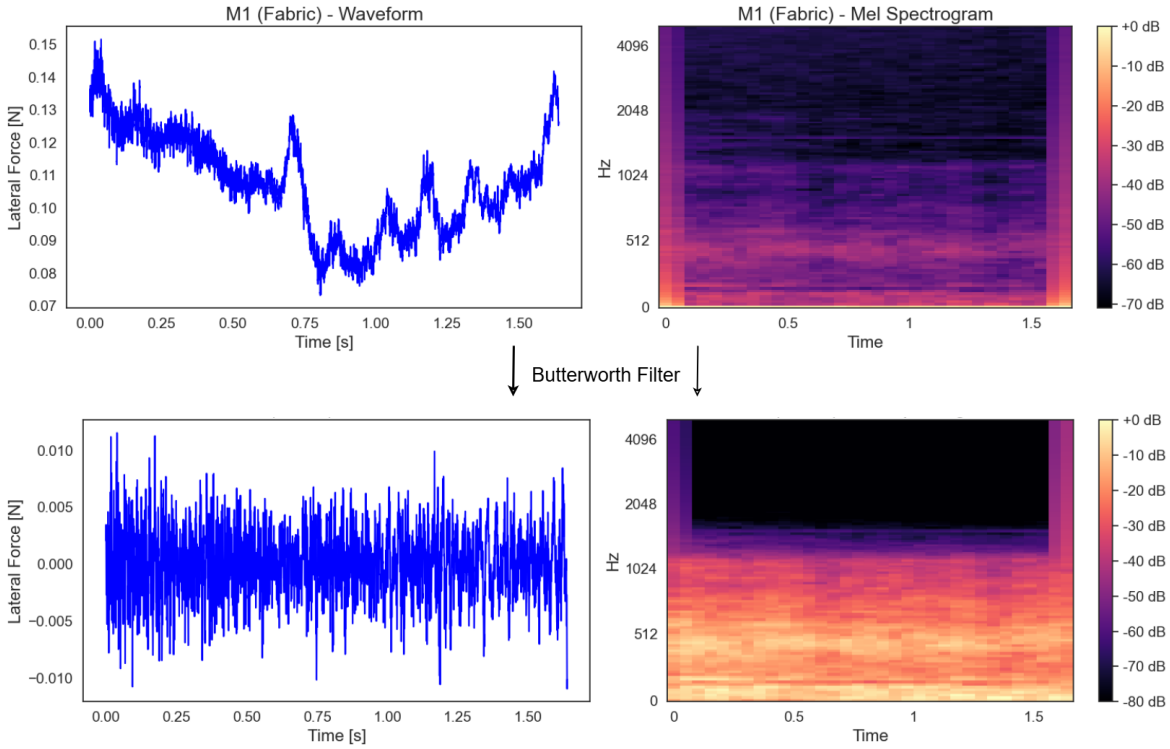


Fig. 1: A 1.64-second section of a friction signal recorded from a piece of fabric is depicted next to its mel-spectrogram. The top row is directly from the database, and the bottom row is the result from applying a bandpass filter between 20 and 1000 Hz. As a result of the filter, the waveform signal is now centered around 0 N, showing that a large portion of the signal's magnitude was made up of the frequencies outside of the bandpass.

1) **DALE**: Diffusion Adjective-to-Latent Encoder. Maps a psychophysical feature vector to a latent representation of texture characteristics.
2) **LCTG**: Latent-Conditioned Texture Generator. Converts the latent representation into a time-domain texture signal suitable for rendering on friction modulation devices.

This division allows for heavy computation to be handled upfront (in the DALE), keeping the LCTG lightweight for real-time inference. Figure 3 illustrates this inference pipeline.

Our database provides the texture signals with the input rating labels we need to connect the input to the output. However, since we are proposing the use of a latent representation with an unknown relation to the input, we adopt a two-phase training strategy:

- Train the LCTG: Learn to reconstruct texture signals and simultaneously learn a latent representation encoding from real texture data to facilitate the reconstruction.
- Train the DALE: Mimic the behavior of the LCTG's encoder by learning to map ratings to the same latent space.

In the first training stage, the encoder of the Latent-Conditioned Texture Generator (LCTG) learns to produce a latent representation that maximizes the information's usefulness to its paired decoder. While this latent space does not carry explicit semantic meaning, it is derived from spectrograms of real texture signals. Although we could hypothesize what features might be important to encode, training the encoder jointly with the diffusion-based decoder enables the use of backpropagation to *learn* an optimal representation.

In the second stage, the Diffusion Adjective-to-Latent Encoder (DALE) is trained to replicate the behavior of the LCTG encoder. By leveraging the dataset linking texture signals to psychophysical ratings, DALE learns to map directly from ratings to latent representations. This two-stage training process is illustrated in Figure 4.

1) Part 1: Latent Conditioned Texture Generator: The LCTG consists of three essential components:

- A Mel-spectrogram conversion
- An encoder that compresses the spectrogram signals into a latent vector.
- A diffusion-based decoder conditioned on the latent vector, used to iteratively denoise a signal.

Ultimately, these components work as a type of autoencoder, compressing the signal through the mel-spectrogram and encoder and then reconstructing it using the decoder. During training or inference without the DALE, the LCTG's sole purpose is to *reconstruct* given textures.

Mel-Spectrogram Conversion To begin the encoding process, each texture signal is transformed into a Mel-spectrogram. This involves computing the Short-Time Fourier

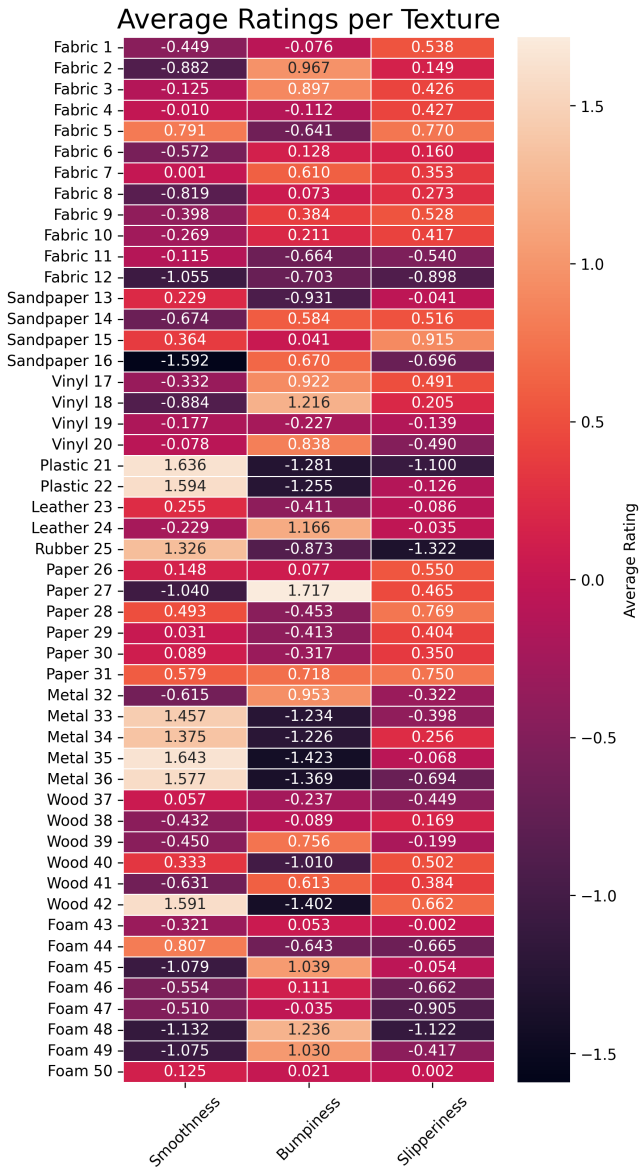


Fig. 2: Heatmap showing the average normalized rating given by the 12 participants to each of the 50 textures for the adjectives: Smoothness, Bumpiness, and Slipperiness. Values are annotated; color bar included for quick pattern reference. Inter-category variety exists, as for example, out of the Fabrics, Fabric 2 has the highest bumpiness of 0.97 and 12 the lowest of -0.703, making for a spread of 1.70 standard deviations.

Transform (STFT), which decomposes the signal into its frequency components over time, resulting in a two-dimensional representation of frequency magnitudes across successive time frames. The frequency axis is then converted to the Mel scale. This is a perceptual scale that more closely aligns with human sensitivity to pitch changes by decreasing the resolution as frequencies get higher. Given demonstrated similarities between tactile and auditory perception [24], the Mel-spectrogram serves as an effective representation. It emphasizes low-

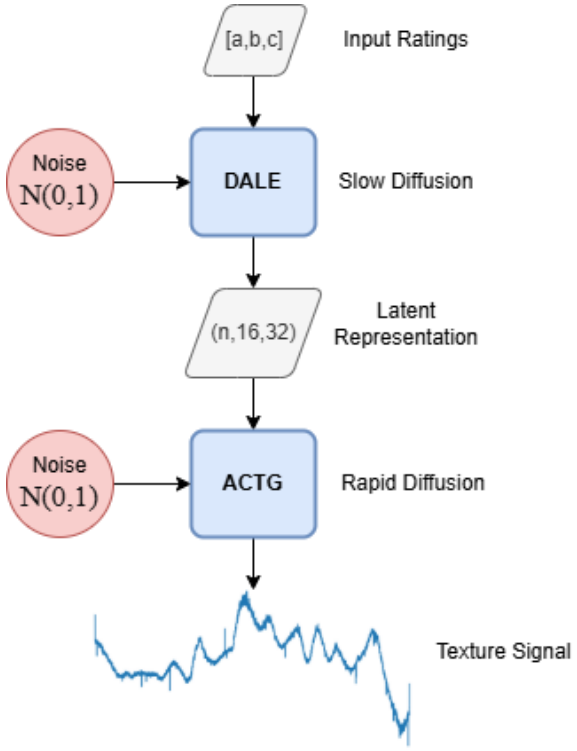


Fig. 3: The two-step inference process for which our model is built. Both models use noise as an input, and the only user input is the ratings at the top (into the DALE). The output comes out of the LCTG. In practice, these two models are integrated together.

frequency components, which are particularly noticeable in human touch perception.

1D-Spectrogram Encoder The Mel-spectrogram is processed by a 1D convolutional encoder consisting of 10 layers with varying kernel sizes and strides. These layers operate along the temporal axis, treating each Mel-frequency vector as a feature descriptor at each time step.

The use of multiple kernel sizes enables the encoder to capture both short-term temporal details and long-range structural patterns, creating a complete understanding of the signal. Furthermore, the choice of 1D convolutions, rather than 2D, offers a large computational advantage to perform real-time inference. By focusing exclusively on the temporal dimension, the model reduces both parameter count and memory use while retaining the essential temporal dynamics necessary for downstream tasks. This makes the encoder particularly suitable for efficient training and deployment at scale.

The number of channels is reduced through a linear projection: each Mel-frequency bin, initially treated as a separate input channel, is projected into a fixed-size representation of 16 channels.

The Mel-Spectrogram conversion and the encoding result in an overall temporal compression factor of 512×. The combination of channel reduction and temporal subsampling leads to a substantial decrease in the number of parameters and computational complexity.

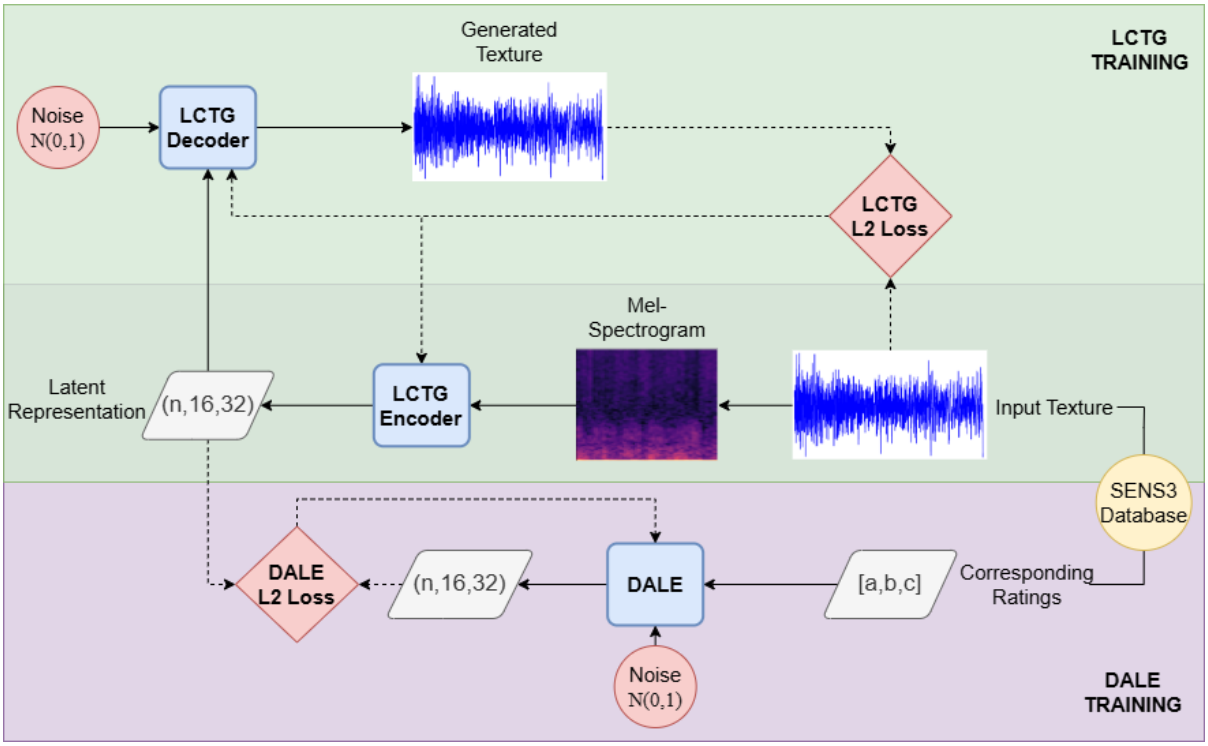


Fig. 4: The flow of in- and output data to and from the models during training is visualized with solid arrows. The flow of data for loss computation and backpropagation is visualized with dotted arrows. The LCTG's training is contained in the green area. The DALE's training is contained in the purple area, which overlaps with the green area. In the overlapping area, the LCTG's encoder is trained during the LCTG's training and consequently used to compute the Latent Representations for training the DALE. n represents the batch size and a, b, c the values of the Roughness, Bumpiness, and Slipperiness ratings. It is important to note that the Noise nodes are composed of the input data with added noise, rather than pure noise.

This latent vector serves as a compressed yet expressive summary of the original texture signal, optimized for use in the generative decoding stage.

U-Net Decoder The decoder is implemented as a 1D U-Net [25], consisting of seven layers in both the downsampling and upsampling paths. The number of channels increases progressively from 8 to 1024 as the temporal resolution is reduced, enabling the network to extract increasingly abstract and high-level features. Skip connections between corresponding downsampling and upsampling layers facilitate the transfer of fine-grained temporal information that might otherwise be lost during downsampling. A symbolic representation of the data flow through our U-Net is shown in Figure 5.

The U-Net architecture is particularly well-suited for generative tasks, as it allows for both global context modeling and the preservation of local structure [25]. The hierarchical encoding path captures coarse-level temporal dependencies, while the decoding path reconstructs fine details with the help of the skip connections. This design improves the fidelity of the reconstructed signal by allowing low-level features to bypass the compression bottleneck and be directly integrated into the reconstruction.

At each layer of the U-Net, two types of conditioning are applied:

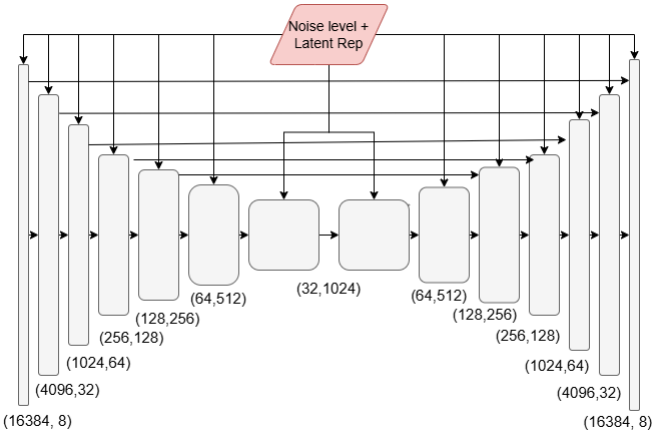


Fig. 5: Symbolic representation of the data flow through the U-Net. Data at different levels of the U-Net travels through the arrows to other levels. At each level, noise modulation and conditioning on the latent representation are applied. The spatial (temporal) dimensions decrease as depth increases, while the number of channels increases accordingly.

- **Noise Modulation:** The model conditions on the noise level using a scale-and-shift operation, where the parameters are functions of the current noise level and learned

during training. This mechanism allows the network to adapt its behavior across denoising timesteps in the diffusion process.

- **Latent Conditioning:** The latent representation is tiled along the temporal axis to match the resolution of the current layer and concatenated along the channel dimension. This enables the decoder to integrate global information from the encoded texture representation throughout the denoising trajectory.

v-objective Diffusion Model

We adopt *v-objective diffusion*, introduced by Salimans and Ho [26], which modifies the diffusion loss to predict a weighted combination of the noise and the original data, referred to as the *velocity*. This objective improves stability in later timesteps and enables high-quality generation in fewer steps compared to standard noise-prediction objectives used in diffusion models.

The loss is defined as:

$$\mathcal{L}_\theta = \mathbb{E}_{x,\varepsilon,t}\left[\|v_t - \hat{v}_\theta(z_t, c, t)\|^2\right] \tag{5}$$

Here, $v_t = \alpha_t \varepsilon - \sigma_t x$ is the velocity target, and $\hat{v}_\theta(z_t, c, t)$ is the model's prediction. The model receives:

- z_t: the noisy signal at timestep t,
- c: a conditioning vector from the latent space, and
- t: the current timestep, used to determine noise weights.

The noisy input is constructed as:

$$z_t = \alpha_t x + \beta_t \varepsilon \tag{6}$$

$$\alpha_t = \cos\left(\tfrac{\pi}{2}\sigma_t\right), \quad \beta_t = \sin\left(\tfrac{\pi}{2}\sigma_t\right) \tag{7}$$

We use a *linear noise schedule*, where $\sigma_t \in [0, 1]$ increases linearly with t. Training across a broad range of σ_t values teaches the model to progressively remove noise through denoising iterations.

Once trained, the model can infer $\hat{v}_\theta$ at any noise level, providing flexibility in the number of sampling steps. Fewer steps enable faster inference, while more steps allow for finer noise removal.

Sampling with DDIM

For inference, we employ *Denoising Diffusion Implicit Models (DDIM)* [18], a fast, non-Markovian sampling strategy compatible with v-objective training. Unlike standard DDPM sampling, which requires many stochastic steps, DDIM enables deterministic or semi-deterministic sampling with significantly fewer iterations.

At each timestep, the following quantities are computed:

$$z_0 = \alpha_t z_t - \beta_t \hat{v}_\theta(z_t, c, t) \tag{8}$$

$$\varepsilon_t = \beta_t z_t + \alpha_t \hat{v}_\theta(z_t, c, t) \tag{9}$$

$$z_{t-1} = \alpha_{t-1} z_0 + \beta_t \varepsilon_t \tag{10}$$

This process allows iterative refinement of the signal while maintaining consistency with the denoising dynamics learned during training.

DDIM sampling is **10–50× faster** than standard DDPM sampling while retaining high sample quality. This makes it particularly well-suited for real-time or resource-constrained generation settings.

Frequency-Aware Loss Component

Inspired by [21], we explore an optional loss term that incorporates frequency-domain information. Specifically, we add a mean-squared error term between the Fourier transforms of the predicted and target velocity signals. This stems from the hypothesis that frequency content plays a central role in texture perception. While this frequency-aware loss may not necessarily improve reconstruction in terms of RMSE, it has the potential to yield outputs that are perceptually more faithful.

The modified loss function is defined as:

$$\mathcal{L}_\theta = \mathbb{E}_{x,\varepsilon,t}\left[\|v_t - \hat{v}_\theta(z_t, c, t)\|^2 + \lambda \left\|\mathcal{F}\{v_t\} - \mathcal{F}\{\hat{v}_\theta(z_t, c, t)\}\right\|^2\right] \tag{11}$$

Here, λ is a weighting factor that controls the influence of the frequency-domain component, and $\mathcal{F}\{\cdot\}$ denotes the Fourier transform.

Together, these design choices form a modular and efficient autoencoding architecture that captures and reconstructs texture signals with temporal and perceptual fidelity, laying the foundation for the 2-step latent generation using our DALE model.

2) Part 2: Diffusion Adjective-to-Latent Encoder: We train the second model, the Diffusion Adjective-to-Latent Encoder (DALE), to approximate the output of the LCTG encoder while conditioning on the psychophysical feature space rather than raw texture signals. This enables the model to generate novel texture representations, rather than merely reconstructing existing ones.

We employ the same backbone for this model, using 1-dimensional UNets, v-objective diffusion, and DDIM sampling.

Dataset Generation

To generate the training data for DALE, we first use the trained encoder of the LCTG model to encode the segmented texture signals from the SENS3 database. We do not run the decoder; only the latent representations are saved. Since each texture in the SENS3 dataset is segmented into 1.64-second fragments, this yields over 200 latent vectors per texture.

Each of these latent vectors is then paired with one of the 12 available sets of psychophysical ratings for the corresponding texture. Rather than using only the average rating, we randomly assign one of the 12 individual rating sets to each latent representation. This strategy increases the diversity of the input space and helps prevent sparsity in the training data.

As a result, we construct a dataset consisting of latent texture representations paired with diverse psychophysical feature vectors, which we use to train the DALE model.

Rating Embedding

The injection of the ratings is done through a cross-attention mechanism at every level of the UNet. Before that, we first

embed the ratings into a higher-dimensional representation using sinusoidal functions to allow the model to find non-linear and periodic patterns:

$$\mathbf{r}_{\text{weighted}} = \mathbf{r} \cdot \mathbf{w}^\top \cdot 2\pi \tag{12}$$

$$\mathbf{r}_{\text{freqs}} = \left[\sin(\mathbf{r}_{\text{weighted}}) \mid \cos(\mathbf{r}_{\text{weighted}})\right] \tag{13}$$

$$\mathbf{r}_{\text{out}} = \left[\mathbf{r} \mid \mathbf{r}_{\text{freqs}}\right] \tag{14}$$

Here, $\mathbf{r}$ is the original rating vector and $\mathbf{w}$ is a learned linear projection that scales the ratings according to their importance. The resulting $\mathbf{r}_{\text{out}}$ vector is a concatenation of the original ratings and their sinusoidal embeddings, and serves as the conditioning input to the cross-attention layers.

The cross-attention block enables the model to determine which aspects of the latent representation should be influenced by which ratings. Different parts of the latent are modulated differently based on psychophysical importance, enabling high-precision, context-dependent modulation.

Context Scaling in Cross-Attention

To further enhance the model's ability to control the influence of the conditioning input, we introduce a learnable scalar parameter that modulates the context vector before it is used to compute the keys and values in the cross-attention mechanism. Specifically, before computing the key–value pairs from the context, we apply a learned multiplicative scale:

$$\mathbf{k}, \mathbf{v} = \texttt{Linear}(\mathbf{c} \cdot \gamma) \tag{15}$$

Where $\mathbf{c}$ denotes the (normalized) context vector derived from the conditioning input, and γ is a learnable scalar initialized to a relatively high value (e.g., 10.0). This allows the model to dynamically adjust the strength of the conditioning during training, enabling more flexible alignment between the latent features and the conditioning signal.

In preliminary experiments, we found this to improve stability during early training and help prevent the attention mechanism from prematurely ignoring the ratings, especially in cases where the conditioning signal was initially noisy or weakly informative.

Conditioning Usage Loss

Due to the subjective nature of the psychophysical ratings, contradictions between ratings and textures are inevitable. As a result, the model may learn to ignore the conditioning input and instead model an *unconditional* distribution over the output textures, effectively collapsing to $p(x \mid z)$, where the clean data x only depends on the noisy data z, rather than the intended conditional distribution $p(x \mid z, r)$, that incorporates rating input r.

This lack of correlation between the ratings and the generated textures would severely impair the downstream performance of our pipeline. To mitigate this, we introduce an auxiliary loss term that explicitly encourages the model to make use of the conditioning input.

$$\mathcal{L}_\theta,\ v_1 = w(z, r, t) \tag{16}$$

$$\_,\ v_2 = w(z, r_{\text{shuffled}}, t) \tag{17}$$

$$\mathcal{L}_{\text{div}} = -\cos(v_1, v_2) \tag{18}$$

$$\mathcal{L}_{\text{total}} = \mathcal{L}_\theta + \lambda_{\text{div}} \cdot \mathcal{L}_{\text{div}} \tag{19}$$

The model is represented by w and the timestep by t. v is the predicted velocity following v-objective diffusion, which is calculated twice, the second time using shuffled ratings. The negative cosine similarity between the two predictions is added as a loss component and weighted by λ_{div}.

Together, these components form the architecture of the DALE. For more details about the exact number of layers and hyperparameters, see Appendix A.

C. Training of DreamTexture

Both models undergo training on an NVIDIA GeForce RTX 4060 Ti. This section underlines how the general model structure was optimized and subsequently trained.

1) Training the LCTG: The Latent-Conditioned Texture Generator (LCTG) is initially trained on the full dataset for 40 epochs. This pretraining phase enables the model to learn generalizable frequency structures across a diverse set of texture signals, thereby capturing broad latent representations that form the foundation for subsequent specialization.

Following this initial stage, the model is fine-tuned on a filtered subset of the data, limited to signals recorded under controlled finger speed and force conditions. This subset is selected based on predefined thresholds to reduce variability in signal characteristics and emphasize more consistent patterns. Fine-tuning in this constrained domain allows the model to specialize its learned representations, enhancing its ability to capture nuanced features and improving performance on data within the defined parameter space.

Hyperparameters were selected through a trial-and-error approach. Due to time constraints, an exhaustive Optuna-based search was not feasible. Instead, ten experimental runs were conducted with varied parameter settings to approximate optimal values (see Appendix B for details). Parameters excluded from this search were chosen based on prior work in music generation [26] or set to commonly used defaults in related architectures.

2) Training the DALE: The Diffusion Adjective-to-Latent Encoder (DALE) is trained directly on the same filtered subset used during the LCTG fine-tuning stage. For each training sample, a latent representation is first extracted using the encoder component of the pretrained LCTG. This latent vector is then paired with a corresponding psychophysical rating, forming the input-output pair for supervised training.

DALE is trained for 40 epochs using this fixed dataset. The hyperparameter selection process mirrors that of the LCTG: multiple models were trained with slight variations in key parameters to empirically identify effective configurations.

IV. Experimental Results and Analysis

1) Generating Textures: The number of diffusion steps used during inference can be freely selected, with higher step counts generally leading to improved generation quality. However, due to the efficiency of the v-objective formulation, high-quality outputs can be achieved even with relatively few steps. Figure 6 illustrates the trade-off between generation quality and computation time across varying diffusion step counts. On an NVIDIA RTX 4060, it is feasible to perform over 50 diffusion steps in real time, achieving an RMSE as low as 0.0060. The optimal balance between performance and efficiency, quantified as the product of RMSE and inference time ($\text{RMSE} \times t$), is at ten diffusion steps, as shown in Table I. This configuration is therefore adopted for generating the results presented in this work.

Steps	Efficiency Score (RMSE × Time)
5	0.002936
10	0.002879
20	0.003718
40	0.006457
60	0.008664
80	0.010333
100	0.013104

TABLE I: Performance-Efficiency Trade-off across different step counts

2) Criteria Overview: Evaluating texture signals is challenging due to their subjective nature and the lack of a precise mathematical mapping to the psychophysical dimensions used for conditioning. We use a combination of quantitative and qualitative metrics to assess model performance. These include RMSE for both waveform and spectral representations of LCTG reconstructions, as well as qualitative evaluations of their perceptual fidelity. DALE performance is assessed through reconstruction RMSE and by measuring latent shift magnitudes in response to varying input ratings. For the complete DALE–LCTG pipeline, we compute RMSE between generated and original textures, and compare average spectra for generations at extreme adjective values to evaluate semantic responsiveness. Inference time is also measured to assess computational efficiency.

Human experiments, which would offer crucial validation of perceptual alignment, were omitted. Due to the limited magnitude of latent shifts and inconsistencies within the dataset, both discussed in later sections, we had concerns about the reliability of the results and ultimately decided to exclude the human evaluation component.

3) Evaluation of LCTG Reconstructions: We begin by evaluating the reconstruction performance of the Latent-Conditioned Texture Generator (LCTG), a critical step to verify how well the latent space encodes perceptually and physically relevant features of the input texture signals. To this end, we reconstruct each texture in the validation set and present four randomly selected examples in Figure 7.

Qualitatively, the reconstructed waveforms closely match the original signals in magnitude and overall shape. Key global features, such as prominent low-frequency components, are generally preserved. However, inspection of the Fourier spectra reveals that some low-frequency content is underrepresented in the reconstructions, and the amplitude of these components tends to be lower than in the original signals. Conversely, the reconstructions sometimes exhibit higher energy in frequencies above 1 kHz, frequencies that lie beyond the typical human tactile perception range and were not present in the training data. These extraneous high-frequency components will be filtered out during rendering, potentially degrading the perceived texture quality and leading to weaker sensations unless compensated by amplification.

Metric	Mean	Median	Max	Min
RMSE	0.011433	0.010723	0.018123	0.006164
FT_RMSE	0.903572	0.903394	1.290457	0.517045
Power Difference	0.000050	0.000048	0.000100	0.000005
SNR (dB)	-1.866411	-1.951638	-1.206888	-2.355481

TABLE II: Primary Quantitative Metrics for the LCTG's reconstruction listed with their ranges. These are obtained from reconstructing a part of the dataset that was kept separately as a validation set.

To complement these qualitative observations, Table II summarizes quantitative reconstruction metrics. The RMSE (Root Mean Squared Error) values are low (mean ≈ 0.0114), indicating close similarity between the original and reconstructed signals in the time domain. The FT_RMSE (Fourier Transform RMSE) is higher (mean ≈ 0.9), reflecting noticeable differences in frequency content, consistent with the spectral observations. The Power Difference is very small (mean $\approx 510^{-5}$), showing that the overall signal energy (power) is well preserved during reconstruction. The SNR (Signal-to-Noise Ratio) averages around -1.87 dB, which implies that reconstruction noise or errors are slightly larger than the original signal power, highlighting areas where reconstruction quality can be improved.

Overall, these results suggest that while the LCTG effectively reconstructs the main temporal structure and power of texture signals, discrepancies remain in spectral fidelity, particularly in the low-frequency range and the presence of unexpected high-frequency content. Addressing these limitations could further enhance the perceptual realism of synthesized textures and their suitability for tactile rendering applications.

4) DALE Faithfulness Evaluation: To evaluate the faithfulness of the DALE model to its conditioning input, we analyze the *latent shift magnitude*, which is defined as the L2 norm between latent vectors generated from inputs with varying rating values. This measure captures the extent to which changes in the conditioning signal (i.e., psychophysical ratings) result in meaningful shifts in the latent space.

To isolate the effect of the conditioning and control for stochastic variation, we fix the noise vector across conditions. For each adjective (Roughness, Bumpiness, Slipperiness), we vary its rating value while holding the other two fixed at zero. Rating values are sampled symmetrically around the zero mean in increments of 0.2. For each input rating value, we generate 30 latent vectors using distinct noise samples and compute the average pairwise L2 distance between latents generated at the

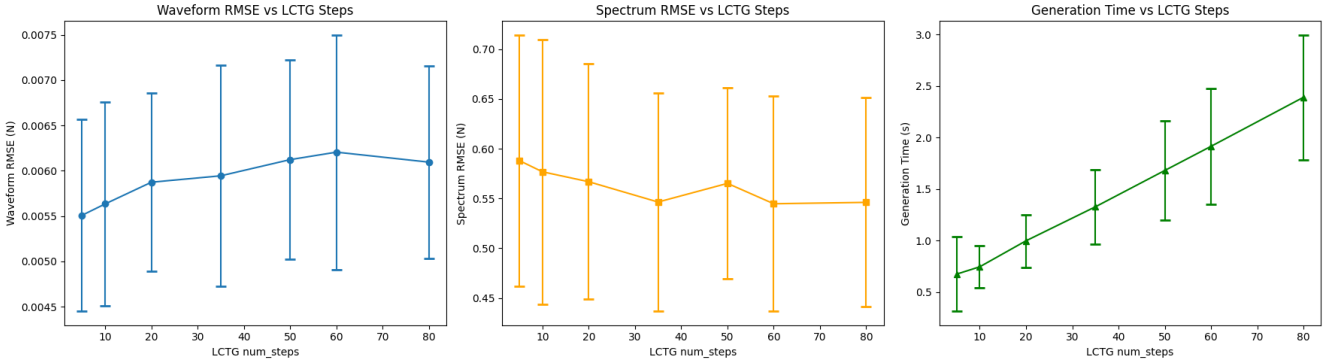


Fig. 6: The full pipeline was run with different amounts of diffusion steps. The left plot shows the waveform RMSE, the middle plot shows the Fourier RMSE, and the right plot shows the generation time, with each plot having the number of steps on the x-axis. The error bars show large deviations in performance and small deviations in generation time.

chosen input value with the latents generated with an input value of 0.

The resulting *latent shift magnitudes* are normalized by the overall magnitude of the latent representations and plotted in Figure 8. The y-axis indicates the normalized shift magnitude, and the x-axis reflects the rating difference from the baseline value. Each curve shows the response of the latent space to variation along one adjective dimension.

The observed shifts are relatively small, with most values falling within a 0–2% range of the latent magnitude. Roughness and Bumpiness exhibit weak but consistent increases in shift magnitude as the input rating diverges from the mean, suggesting that DALE modestly incorporates conditioning information for these dimensions. In contrast, Slipperiness displays no discernible trend, indicating limited responsiveness of the latent representation to this rating dimension under the current configuration.

5) Full Inference Results: To evaluate the reconstructive capabilities of the complete pipeline, we perform the following procedure:

1) Randomly select a texture segment from the validation dataset.
2) Load a corresponding set of ratings provided by a randomly chosen participant.
3) Use the DALE model to generate a latent representation from the ratings.
4) Generate the texture signal using the LCTG.
5) Compare the generated texture signal to the originally selected segment.

An example is shown in Figure 9, where a randomly selected texture signal (blue) is reconstructed three times using three distinct sets of participant ratings. Each reconstruction (orange) reflects the variation in the corresponding rating vector, resulting in noticeable differences in the generated signals, including in their magnitudes.

We repeat this process across 20 randomly selected distinct texture segments, with 5 repetitions per segment, each using a different randomly sampled set of ratings. This evaluation yields a mean waveform RMSE of 0.005569 ± 0.001088 and a spectral RMSE of 0.572766 ± 0.131719. For comparison, the GAN-based image-to-friction model proposed by Cai et al. [32] reports higher signal RMSEs ranging from 0.016 to 0.051, while achieving similar performance on the Fourier spectrum (RMSE between 0.07 and 0.91).

To assess the influence of individual perceptual rating dimensions on generated textures within the full inference pipeline, we systematically varied one rating dimension at a time over a range of –2 to +2 standard deviations (in increments of 0.5), while independently sampling the remaining two dimensions from a standard normal distribution. For each fixed value of the target rating, 50 texture signals were generated using the complete DALE–ACTG pipeline. The average frequency spectrum of the resulting signals was computed using the Fast Fourier Transform (FFT) to identify systematic trends in the spectral characteristics. A full set of results is provided in Appendix D, while Figure 10 compares the average spectra at the extreme values (–2.0 and +2.0) for each rating dimension.

The mean L1 distance between the spectra for extreme values was 0.13 for Smoothness (opposite of Roughness), 0.07 for Bumpiness, and 0.10 for Slipperiness. Across all conditions, the frequency peaks remained largely stationary, with the primary variation manifesting as amplitude scaling rather than spectral shape deformation. Notably, both Smoothness and Slipperiness exhibited gain changes predominantly in the low-frequency range ($<$ 200 Hz), whereas Smoothness showed the most pronounced effects in the high-frequency domain ($>$ 200 Hz). Despite these variations, a qualitative analysis of the figures reveals that all three rating dimensions exert a comparable influence on the output, and the averaged spectral profiles across generated textures suggest limited frequency diversity.

The Frechet Audio Distance (FAD) [30] is a widely used metric for evaluating the similarity between the distributions of real and generated one-dimensional signals, typically in the context of audio. However, it is equally applicable to our use case involving texture signals. To compute the FAD, we generated 500 texture samples using randomly sampled ratings

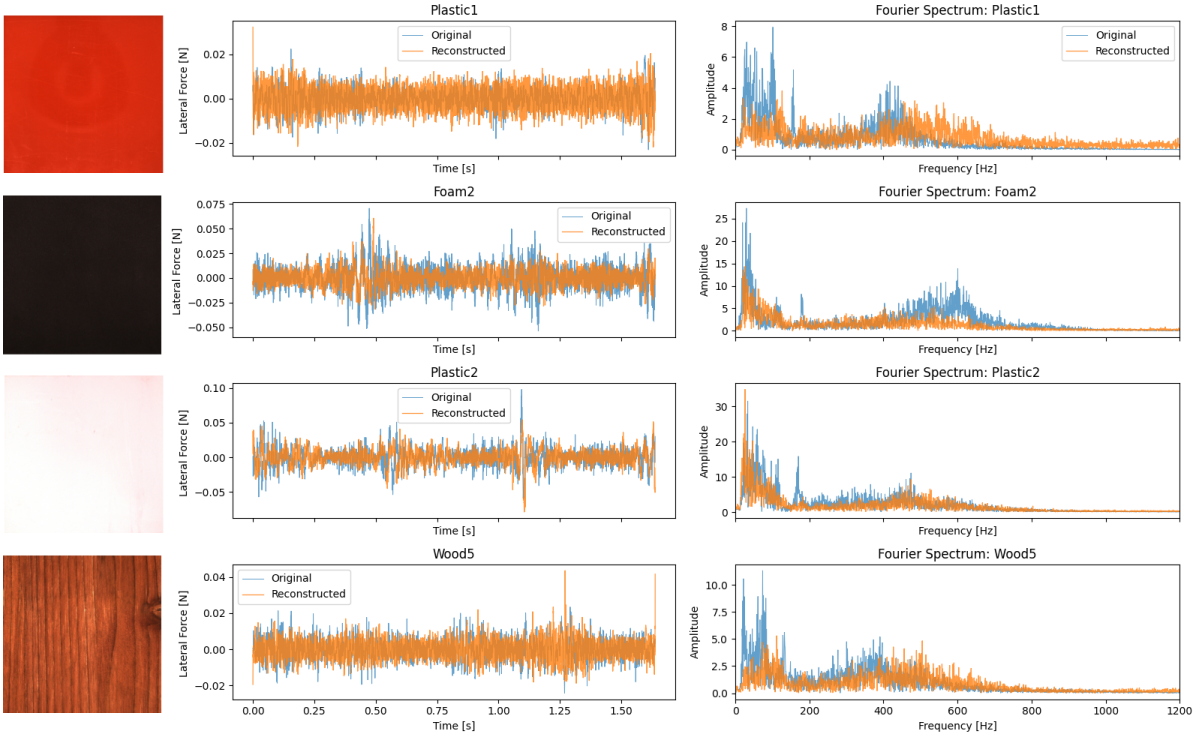


Fig. 7: Reconstruction results for four texture signals from the SENS3 database [7] using the trained LCTG. Each row corresponds to a different texture; the left image shows a photo of the material, while the right plots display its signal waveform and Fourier spectrum. In both plots, the original signal is shown in blue and the reconstructed signal is overlaid in orange. Note that the y-axis scale varies between signals. The Fourier spectrum is truncated at 1200Hz to focus on frequencies relevant to human tactile perception, which generally do not exceed 1000Hz.

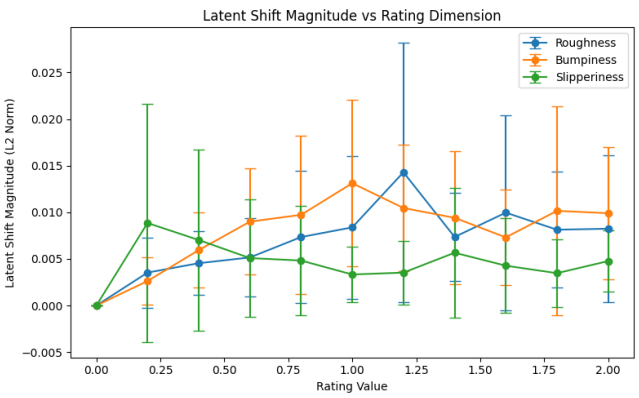


Fig. 8: Latent shift magnitude (y-axis) as a function of change in the input rating (x-axis). Each curve corresponds to one rating dimension, with others held constant. Values are normalized and unitless. Roughness is shown in blue, Bumpiness in orange, and Slipperiness in green.

drawn from a standard normal distribution, consistent with the rating normalization applied during model training. The resulting FAD score, calculated against the full set of real textures in our database, was 21.62. For comparison, state-of-the-art music generation models such as [26] typically achieve FAD scores below 5, highlighting the greater distributional distance in our generated textures.

V. Discussion

By introducing DreamTexture, the first generative diffusion model that maps psychophysical descriptors to texture signals, we aim to initiate a broader conversation about how generative AI can contribute to advancements in haptics. While the strengths of diffusion-based generation are evident in our results, it is equally important to acknowledge and address the model's current limitations.

1) Diffusion Model Performance: Our approach is the first to use this particular input space, and therefore, no directly comparable baselines exist. However, we can draw several conclusions based on our results.

Texture Generation Quality Our model successfully learns to generate valid texture signals without collapsing to a limited variety, as evidenced by the diversity of outputs in Figure 9. A clear mapping between the input space and generated outputs is established, as shown in Figure 7 and Figure 10.

The full DALE-ACTG pipeline achieves a signal RMSE of **0.0056**, indicating strong reconstruction fidelity in the time domain. However, performance in the frequency domain is less impressive, with a Fourier RMSE of **0.58**. These results

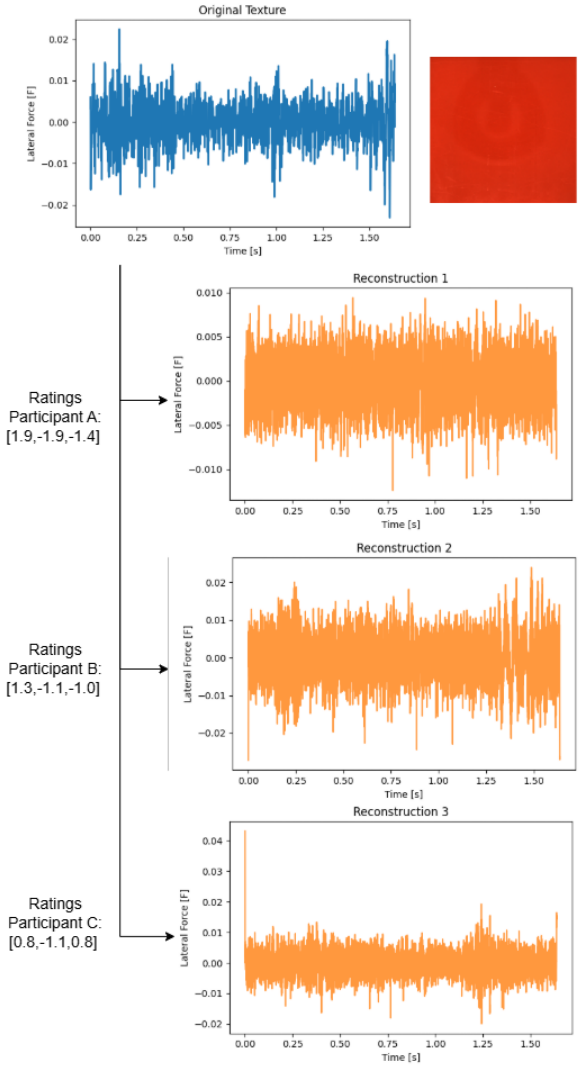


Fig. 9: Example using plastic as the reference texture. A single randomly selected segment from the database is shown in blue (top), with three reconstructions in orange below—each generated from a distinct set of ratings, shown on the left. The rating vectors vary significantly, leading to corresponding variations in the generated textures, including differences in signal magnitude.

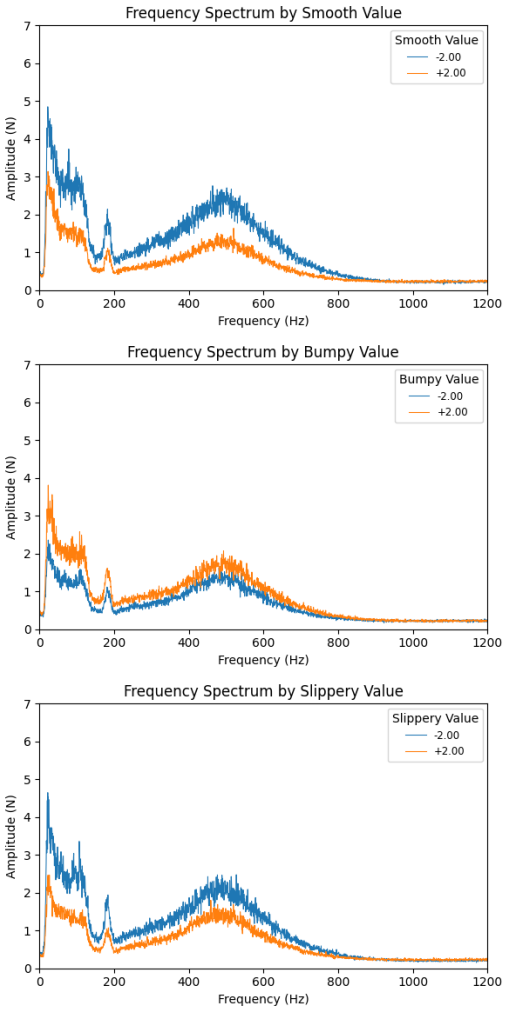


Fig. 10: Comparison of the average frequency spectra for textures generated at the extreme values (+2.00 and –2.00) of each perceptual rating dimension. For clarity, "Smooth" is shown in place of "Rough" to reflect the positive direction of the axis. The spectra corresponding to –2.00 values are shown in blue, while those for +2.00 are shown in orange.

suggest that while the waveform is closely matched, some spectral characteristics are not accurately captured.

LCTG Reconstruction Bottleneck The performance of the encoder of the LCTG (Latent-Conditioned Texture Generator) acts as an upper bound on reconstruction quality, since the DALE can at best approximate it, and therefore, we consider ways to improve its accuracy. We observe that high-magnitude frequency peaks are not faithfully reconstructed. Interestingly, some reconstructions include frequencies above 1000 Hz, despite these being filtered out of the training dataset. Since such components are imperceptible and potentially harmful to perceptual fidelity, this mismatch may reduce the texture sensations' realism and accuracy.

To improve the performance, several strategies may be helpful:

- **Expand latent space:** Increasing the size of the latent representation may allow the model to store and reconstruct richer frequency information.
- **Data balancing:** Augment the dataset to include more textures with high amplitude frequency peaks, improving the model's ability to learn such cases.
- **Filtering before loss:** Applying a frequency-domain filter before loss computation may prevent high-frequency artifacts from influencing the learning process.

Loss Function Insights During testing, we found that the optimal weight for the frequency-aware loss component of the LCTG was zero, meaning its inclusion did not improve RMSE scores. This suggests that the current model or training setup struggles to optimize for frequency reconstruction alongside

time-domain accuracy. Several factors could contribute to this:

- RMSE in the time domain is more sensitive to low-frequency errors, leading the model to prioritize these components.
- The model architecture or training procedure may not be well-suited to capture complex spectral patterns.
- The presence of high-frequency noise might be a compensatory effect for underrepresented low-frequency energy.

We suggest experimenting with alternative frequency-domain loss components, such as one that is filtered or weighted to prioritize perceptually relevant components (e.g., $<$1000 Hz). Alternately, experiment with scheduled loss weighting to guide training phases.

Diversity of Generations Although the model demonstrates strong performance in terms of RMSE, the diversity of the generated signals could be improved. As shown in Figure 10, the frequency spectra of generated signals exhibit limited sensitivity to variations in the input psychophysical ratings. In contrast, Figure 7 reveals that the generated signals vary in magnitude and display frictional peaks at different times. Additional examples in Appendix D further illustrate that the model is capable of producing a broad range of texture signals, indicating some degree of generative diversity.

The Fréchet Audio Distance (FAD) score remains high, suggesting a significant difference between the distributions of the generated and real texture signals. One plausible explanation is that the generative distribution is narrower than the true data distribution. Another contributing factor may be the use of normally distributed psychophysical ratings during FAD computation, which may not accurately reflect the actual distribution of textures in the psychophysical space (as illustrated in Appendix B). This mismatch could be a major contributor to the elevated FAD score, despite not necessarily indicating poor generation quality. To more accurately assess distributional alignment, alternative sampling strategies that better capture the empirical distribution of the psychophysical space should be explored.

2) Effect of Psychophysical Ratings: The influence of psychophysical ratings on the generated textures is largely determined by the DALE model's capacity to learn a meaningful mapping between the psychophysical space and the latent representation.

The question may arise whether the model genuinely conditions on the input ratings or merely reproduces the target distribution without incorporating the conditioning information. Supporting this concern, Figure 8 shows that the latent representation changes by only approximately 1% in response to a full standard-deviation increase in a single rating dimension. However, we do observe that the magnitude of latent shift increases with the difference in roughness and bumpiness ratings, suggesting that some (albeit weak) relationship is being captured. Complementing this, Figure 10 indicates that the average frequency magnitude increases by up to 60% across certain frequency bands when comparing signals generated from extreme values in the psychophysical dimensions, implying that the model does not entirely ignore the conditioning inputs.

Nevertheless, qualitative inspection reveals that the influence of each psychophysical dimension appears highly similar, typically resulting in a gain across the signal. This raises concerns about the model's ability to meaningfully differentiate between the psychophysical axes. Despite correct implementation, we suspect that the limited expressivity of the ratings is primarily attributable to issues in the training data. The mapping relies on the existence of a strong correlation between the original texture signals in the dataset and their associated psychophysical ratings, since DALE is trained to emulate an encoder that maps texture signals to latent representations.

While the RMSE scores from the full pipeline suggest that the model is capable of generating textures that resemble the originals, the low correlations observed between the psychophysical ratings and the texture signals cast doubt on the validity of the conditioning framework. As such, we cannot conclude that a strong or meaningful relationship between the employed psychophysical feature space and the generated texture signals has been successfully established.

3) Dataset Limitations: Several limitations in the SENS3 dataset became apparent during development. First, the selected dimensions are not completely independent. Roughness/Smoothness and Bumpiness are correlated (see Figure 11), reducing the model's ability to isolate these features in conditioning. Attempting to change one while keeping the other fixed becomes perceptually ambiguous, since textures like that do not exist and perhaps can not exist. Second, Slipperiness ratings are unevenly distributed, showing a strong bias toward slippery textures (see Appendix C). Most values cluster in the upper portion of the scale, resulting in an underrepresentation of rough or sticky textures. This likely reduces the model's sensitivity to that axis.

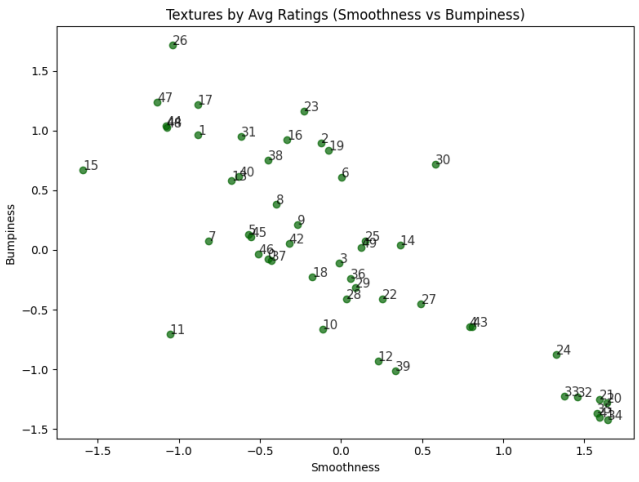


Fig. 11: Each of the fifty textures' locations on the smoothness and bumpiness axes. Each dot represents one texture and is labeled with the number it has in the SENS3 catalog.

Third, participant disagreement is notable: the same texture

often receives ratings that vary by more than one standard deviation across individuals. This subjectivity challenges the model's ability to learn a clean mapping from psychophysical space to texture signal and forces the model to rely less on the conditioning and more on the overall characteristics of the complete texture distribution. We observe the spread of ratings within each particular texture and plot the ones with the largest spread in figure 12. For the textures not included, the standard deviations are shown in Appendix C.

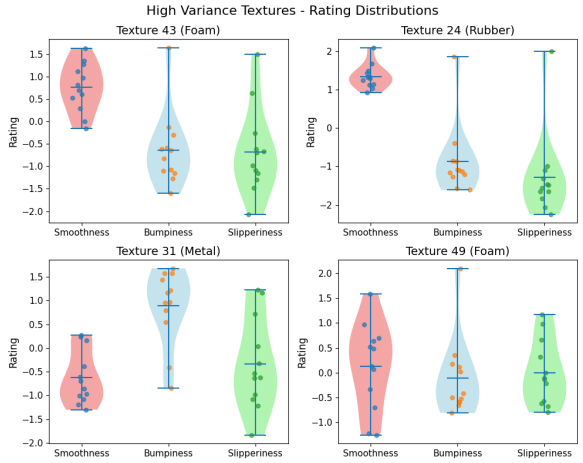


Fig. 12: The four textures that had the most spread across participants in the ratings are depicted here. Ratings have been normalised. Under each violin, the corresponding rating is listed. Each datapoint is visualised as a dot.

These limitations highlight why we can not draw any conclusions about the strength of the model's conditioning on subjective inputs. In future work, a larger, more balanced dataset with controlled psychophysical sampling would help address these issues. This could perhaps be achieved by giving example textures for the extremes of each axis or giving clearer definitions of what each adjective means.

Possibly, to increase the size of the database, the model could learn with shorter samples. Our choice of 1.6384 seconds was made early in the design process, selected to be a power of two, which facilitates reliable downsampling and upsampling within the model architecture, while also being as short as possible to maximize the number of available training samples. This length was long enough to preserve the presence of lower-frequency components in the signal. However, since frequencies below 20 Hz are removed through bandpass filtering, shorter durations could have been used without significant information loss.

4) Generated Texture Database: Texture signal recordings are inherently noisy due to the high sensitivity of the sensors and the variability introduced by individual finger dynamics. As a result, the same texture produces different signals across participants. In our current dataset, texture signals are associated with perceptual ratings collected from a different group of participants than those who originally recorded the signals. This mismatch introduces a potential source of noise and uncertainty in the learning process. Furthermore, since the ratings were collected from interactions with real textures, our training pipeline relies on the friction modulation device's ability to accurately reproduce those recorded sensations.

As a future direction, we propose constructing a fully synthetic texture database composed of digitally generated signals with controlled spectral content. By collecting perceptual ratings directly from participants interacting with these digitally rendered textures via a friction modulation device, we can establish a clean one-to-one correspondence between signal and rating. This approach is expected to benefit model training by reducing noise and ensuring consistent signal-label pairs. However, it may come at the cost of realism, as synthetic textures tend to be more periodic and less complex than real-world textures. Moreover, models trained exclusively on digitally generated signals may overfit to the specific characteristics of the rendering of the specific friction modulation device that is used and generalize poorly to other haptic display technologies.

5) Model Efficiency: Our results indicate that the model is well-suited for real-time inference. As shown in Figure 6, inference times remain consistently under one second when using fewer than 50 diffusion steps. Notably, with the use of a velocity-based diffusion objective, reconstruction performance does not increase beyond approximately 20 inference steps, suggesting that low step counts are sufficient for real-time deployment while maintaining good performance.

This efficiency opens the possibility for real-time texture generation conditioned on interaction parameters, such as scanning speed and applied force. Such an application would require the model to adapt texture outputs dynamically via injection mechanisms. However, realizing this functionality would necessitate a substantially larger dataset, comprising a wide range of texture signals recorded under diverse force–velocity conditions to ensure sufficient coverage of the interaction parameter space.

In terms of training efficiency, the model demonstrates fast convergence, with 40 training epochs on the full dataset completing in under 12 hours. Nonetheless, due to time constraints, we were unable to perform an exhaustive hyperparameter search across all components of the architecture. A comprehensive exploration of the model's hyperparameter landscape remains an important direction for future work.

6) Human Validation: An essential direction for future work is the empirical validation of the model using human participants. Since the model operates within a subjective perceptual space, its effectiveness cannot be fully assessed without relying on human perception. Objective metrics alone are insufficient to determine whether the generated textures correspond to the intended psychophysical ratings, since we are using a machine learning model to find correlations that we don't know of.

In the present work, we chose not to include a user study. This decision was based on the fact that the current model architecture and training data require further refinement to ensure that perceptual ratings have a clear and consistent

influence on the generated textures. Currently, the dataset lacks the consistency and balance necessary to reliably support perceptual validation.

Informal experimentation using an electrovibration display revealed that differences in roughness were sometimes perceptible across generated textures. However, variations along the other perceptual dimensions were often difficult to distinguish, and small to medium changes in input ratings did not consistently translate into noticeable perceptual differences. These findings highlight the need for an improved training dataset and more precise control over the texture generation before formal perceptual studies can be conducted with confidence.

VI. Conclusions

We presented *DreamTexture*, a two-stage latent diffusion model designed to generate tactile texture signals conditioned on perceptual descriptors in psychophysical space. By leveraging a modular architecture, DreamTexture effectively bridges subjective human language and haptic signal generation, enabling real-time synthesis of textures that reflect perceptual qualities such as roughness, bumpiness, and slipperiness.

Our evaluation demonstrates that diffusion models are well-suited for modeling tactile signals, achieving low reconstruction errors and good qualitative results. These results indicate that diffusion-based methods can effectively capture the complex structure of real-world texture data.

Our analysis further reveals that the model exhibits some responsiveness to variations in psychophysical input, with changes in the perceptual descriptors resulting in measurable differences in the generated textures. However, this effect remains limited, likely due to constraints in the dataset, including correlated ratings, conflicting ratings, and the lack of direct correspondence between signal-recording and rating participants.

In terms of practical deployment, the model performs real-time inference on consumer-grade hardware, with fast sampling enabled by DDIM and an efficient decoding pipeline. This positions DreamTexture as a viable solution for interactive applications such as haptic prototyping and texture authoring.

While as discussed in previous sections, some challenges remain, the proposed architecture lays a foundation for future advances, such as more expressive conditioning schemes, adaptive user-in-the-loop feedback, broader compatibility with tactile rendering platforms, and user studies.

Overall, DreamTexture advances the field of generative haptics by integrating psychophysical understanding with state-of-the-art generative modeling, paving the way toward more intuitive, personalized, and scalable tactile content creation.

References

[1] "New Neuroscience Study: Haptics Intensifies Emotions, Increases Engagement, Memorability." BioSpace, 23 Oct. 2018, www.biospace.com/new-neuroscience-study-haptics-intensifies-emotions-increases-engagement-memorability.

[2] R. Noe, "Researchers Develop Textile-Based Haptic Feedback System - Core77," Core77, Aug. 31, 2023. https://www.core77.com/posts/125612/Researchers-Develop-Textile-Based-Haptic-Feedback-System (accessed Jun. 02, 2025).

[3] Q. Wu, J. Li, H. Seifi, and K. Hornbæk, "Vipins: Combining a pin array and vibrotactile actuators to render complex shapes and textures," International Journal of Human-Computer Studies, vol. 197, p. 103464, Feb. 2025, doi: https://doi.org/10.1016/j.ijhcs.2025.103464.

[4] "Gloves G1 — HaptX," HaptX. https://haptx.com/gloves-g1/

[5] Perlin, Ken., ACM Digital Library., ACM Special Interest Group on Computer-Human Interaction., & ACM Special Interest Group on Computer Graphics and Interactive Techniques. (2010). TeslaTouch: Electrovibration for Touch Surfaces. ACM.

[6] Bansal, R. (n.d.). Fundamentals of engineering electromagnetics.

[7] J. K. Balasubramanian, B. L. Kodak, and Yasemin Vardar, "SENS3: Multisensory Database of Finger-Surface Interactions and Corresponding Sensations," Lecture notes in computer science, pp. 262–277, Nov. 2024, doi: https://doi.org/10.1007/978-3-031-70058-3_21.

[8] D. P. Kingma and M. Welling, "Auto-Encoding Variational Bayes," arXiv.org, Dec. 20, 2013. https://arxiv.org/abs/1312.6114

[9] I. J. Goodfellow et al., "Generative Adversarial Networks," arXiv.org, Jun. 10, 2014. https://arxiv.org/abs/1406.2661

[10] Betker, J., Goh, G., Jing, L., Brooks, T., Wang, J., Li, L., Ouyang, L., Zhuang, J., Lee, J., Guo, Y., Manassra, W., Dhariwal, P., Chu, C., Jiao, Y., & Ramesh, A. (n.d.). Improving Image Generation with Better Captions.

[11] Huang, Q., Park, D. S., Wang, T., Denk, T. I., Ly, A., Chen, N., Zhang, Z., Zhang, Z., Yu, J., Frank, C., Engel, J., Le, Q. v., Chan, W., Chen, Z., & Han, W. (2023). Noise2Music: Text-conditioned Music Generation with Diffusion Models. http://arxiv.org/abs/2302.03917

[12] Xi, Q., Wang, F., Tao, L., Zhang, H., Jiang, X., & Wu, J. (2024). CM-AVAE: Cross-Modal Adversarial Variational Autoencoder for Visual-to-Tactile Data Generation. IEEE Robotics and Automation Letters, 9(6), 5214–5221. https://doi.org/10.1109/LRA.2024.3387146

[13] Y. Chen, L. Huang, and T. Gou, "Applications and Advances of Artificial Intelligence in Music Generation:A Review," arXiv.org, 2024. https://arxiv.org/abs/2409.03715

[14] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-Resolution Image Synthesis with Latent Diffusion Models," Apr. 2022. Available: https://arxiv.org/pdf/2112.10752

[15] K. Preechakul, N. Chatthee, S. Wizadwongsa, and S. Suwajanakorn, "Diffusion Autoencoders: Toward a Meaningful and Decodable Representation," arXiv:2111.15640 [cs], Mar. 2022, Accessed: Sep. 29, 2022. [Online]. Available: https://arxiv.org/abs/2111.15640

[16] J. Ho, A. Jain, and P. Abbeel, "Denoising Diffusion Probabilistic Models," arxiv.org, Jun. 2020, doi: https://doi.org/10.48550/arXiv.2006.11239.

[17] Sohl-Dickstein, J., Weiss, E. A., Maheswaranathan, N., & Ganguli, S. (2015). Deep Unsupervised Learning using Nonequilibrium Thermodynamics. http://arxiv.org/abs/1503.03585

[18] T. Dujardin, J. Gaubil, and T. Richard, "Denoising Diffusion Implicit Models Generative Modelling: Project report." Accessed: Jun. 03, 2025. [Online]. Available: https://www.jgaubil.com/docs/gamma_ddim.pdf

[19] S. Butterworth, "On the theory of filter amplifiers," Experimental Wireless & the Wireless Engineer, vol.7, pp.536–541, Oct.1930.

[20] Shultz, C., Peshkin, M., & Colgate, J. E. (2018). The application of tactile, audible, and ultrasonic forces to human fingertips using broadband electroadhesion. IEEE Transactions on Haptics, 11(2), 279–290. https://doi.org/10.1109/TOH.2018.2793867

[21] S. Takaki, H. Kameoka, and J. Yamagishi, "Training a Neural Speech Waveform Model using Spectral Losses of Short-Time Fourier Transform and Continuous Wavelet Transform," arXiv.org, 2019. https://arxiv.org/abs/1903.12392 (accessed Jun. 03, 2025).

[22] Okamoto, S., Nagano, H. and Yamada, Y. (2013) 'Psychophysical dimensions of tactile perception of textures', IEEE Transactions on Haptics, 6(1), pp. 81–93. doi:10.1109/toh.2012.32.

[23] Hassan, W., Abdulali, A., & Jeon, S. (2020). Authoring new haptic textures based on interpolation of real textures in affective space. IEEE Transactions on Industrial Electronics, 67(1), 667–676. https://doi.org/10.1109/TIE.2019.2914572

[24] C. Bernard, J. Monnoyer, M. Wiertlewski, and S. Ystad, "Rhythm perception is shared between audio and haptics," Scientific Reports, vol. 12, no. 1, Mar. 2022, doi: https://doi.org/10.1038/s41598-022-08152-w.

[25] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation," arXiv.org, May 18, 2015. https://arxiv.org/abs/1505.04597

[26] Schneider, F., Kamal, O., Jin, Z. and Schölkopf, B. (2024) 'Moûsai: Efficient Text-to-Music Diffusion Models', Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 8050–8068. doi:10.18653/v1/2024.acl-long.437.

[27] Chen, M., Radford, A., Child, R., Wu, J., Jun, H., Dhariwal, P., Luan, D., & Sutskever, I. (n.d.). Generative Pretraining from Pixels.

[28] K. Kilgour, M. Zuluaga, Dominik Roblek, and M. Sharifi, "Fréchet Audio Distance: A Reference-Free Metric for Evaluating Music Enhancement Algorithms," Sep. 2019, doi: https://doi.org/10.21437/interspeech.2019-2219.

[29] J. K. Balasubramanian, D. M. Pool, and Y. Vardar, "Sliding Speed Influences Electrovibration-Induced Finger Friction Dynamics on Touchscreens," arXiv.org, 2025. https://arxiv.org/abs/2505.11162 (accessed Jun. 03, 2025).

[30] Yusuke Ujitoko and Y. Ban, "Vibrotactile Signal Generation from Texture Images or Attributes Using Generative Adversarial Network," Lecture notes in computer science, pp. 25–36, Jan. 2018, doi: https://doi.org/10.1007/978-3-319-93399-3_3.

[31] Richardson, B. A., Vardar, Y., Wallraven, C., & Kuchenbecker, K. J. (2022). Learning to Feel Textures: Predicting Perceptual Similarities From Unconstrained Finger-Surface Interactions. IEEE Transactions on Haptics, 15(4), 705–717. https://doi.org/10.1109/TOH.2022.3212701

[32] Cai, S., Zhao, L., Ban, Y., Narumi, T., Liu, Y. and Zhu, K. (2022) 'GAN-based image-to-friction generation for tactile simulation of fabric material', Computers & Graphics, 102, pp. 460–473. doi:10.1016/j.cag.2021.09.007.

A

Detailed Model Architecture

This appendix contains all of the parameters and the exact layers of the models we trained and used to generate the results found in this paper.

LCTG Architecture

Mel-Spectrogram Conversion

- **Mel spectrogram settings:**
  - Number of mel channels: 64
  - Mel sample rate: 10,000 Hz
  - Log normalization: enabled
- **FFT settings:**
  - FFT size: 1024
  - Hop length: 256
  - Window length: 1024
- **Other options:**
  - Center waveform before FFT: disabled
  - Normalize waveform: disabled
  - Normalize log-mel: disabled

Encoder

- **Encoder settings:**
  - Input channels: 512
  - Output channels: 64 (Reduced to this number of channels in the final layer)
  - Channel multipliers: 1,1 (This means there are two sets of convolutions, each multiplying the number of channels by 1 in total)
  - Downscaling factor: 2
  - Number of convolution blocks: 12
  - Bottleneck activation: Tanh

UNet

- **UNet settings:**
  - Latent injection depth: 6 (This is the layer(s) at which conditioning happens)

  - Channels per level: 8, 32, 64, 128, 256, 512, 1024 (total layers = 7)
  - Downscaling factors per level: 1, 4, 4, 4, 2, 2, 2 (Downscaling of the data length)
  - Number of blocks per level: 1, 2, 2, 2, 2, 4, 4 (Amount of convolutional blocks at each level, stride depends on downscaling factor)

DALE Architecture

U-Net settings:

- **U-Net settings:**
  - Input channels: 64 (must match the latent representation)
  - Channels per layer: 64, 128, 256, 512, 1024 (total layers = 5)
  - Downscaling factors per layer: 1, 1, 2, 2, 2
  - Number of blocks per layer: 1, 2, 4, 4, 4
  - Attention enabled on layers: 0, 0, 1, 1, 1 (0 is disabled, 1 is enabled)
  - Cross-attention enabled on layers: 1, 0, 1, 1, 1
  - Attention heads: 8
  - Features per attention layer: 64

B

(Hyper)parameter selection

LCTG parameters

The encoder parameters were estimated based on [2]. The UNet parameters were set based on manual trial-and-error, considering the computational limitations while maximizing the performance. Training hyperparameters were determined through hyperparameter optimization, which involved fully training the model multiple times (40 epochs) and comparing validation losses for each run, as shown in Table B.1. The final weights were based on the run with the best final validation loss. The following parameters were chosen:

- `learning rate = 1e-4`; Chosen from the range [1e-5,1e-2]
- `weight_decay = 0`; Chosen from range [0,1e-3]
- `batch_size = 48`; Chosen from options [16,32,48,64,128]
- `optimizer = Adam`; Chosen from options [Adam, AdamW]

Table B.1: Hyperparameters tried in each of the ten runs and the validation loss achieved. This loss is the RMSE for a singular diffusion step. Run 9 had the best performance.

Run	Learning Rate	Weight Decay	Batch Size	Optimizer	Validation Loss
1	$1e{-5}$	0	128	Adam	0.143
2	$1e{-5}$	0	64	Adam	0.089
3	$1e{-4}$	$1e{-3}$	48	Adam	0.098
4	$1e{-2}$	$1e{-2}$	48	Adam	0.105
5	$1e{-3}$	$1e{-2}$	48	AdamW	0.114
6	$1e{-3}$	0	48	AdamW	0.099
7	$1e{-4}$	$1e{-3}$	16	Adam	0.089
8	$1e{-4}$	0	32	Adam	0.094
9	$1e{-4}$	0	48	Adam	0.086
10	$1e{-4}$	0	48	AdamW	0.092

DALE parameters:

The UNet parameters were set based on manual trial-and-error, considering the computational limitations while maximizing the performance. The hyperparameters we chose were equal to those for the LCTG.

C

Distribution of SENS3 Database

This appendix shows a more detailed analysis of the SENS3 database [1], importantly the average texture rating for each texture in the input space and the distribution of individual ratings per texture.

In the SENS3 database, the textures are distributed as follows:

- Textures 1-12: Fabric
- Textures 13-16: Sandpaper
- Textures 17-20: Vinyl
- Textures 21-22: Plastic
- Textures 23-24: Leather
- Texture 25: Rubber
- Textures 26-31: Paper
- Textures 32-36: Metal
- Textures 37-42: Wood
- Textures 43-50: Foam

Figures C.1, and C.2 illustrate the distribution of all 50 textures from the SENS3 database across pairs of psychophysical axes (bumpiness vs roughness is present in the paper's text). Each plot corresponds to a unique combination of two axes, showing how the textures are positioned within the selected dimensions of the psychophysical space.

For each texture, we quantified the variability in participant ratings across the three perceptual dimensions. These inter-participant standard deviations, visualized in Figure C.3, reflect the degree of consensus or disagreement among raters. Notably, most textures exhibit considerable variability in at least one dimension, with standard deviations frequently exceeding half a normalized rating unit (which is expressed in standard deviations of inter-participant ratings for that particular dimension).

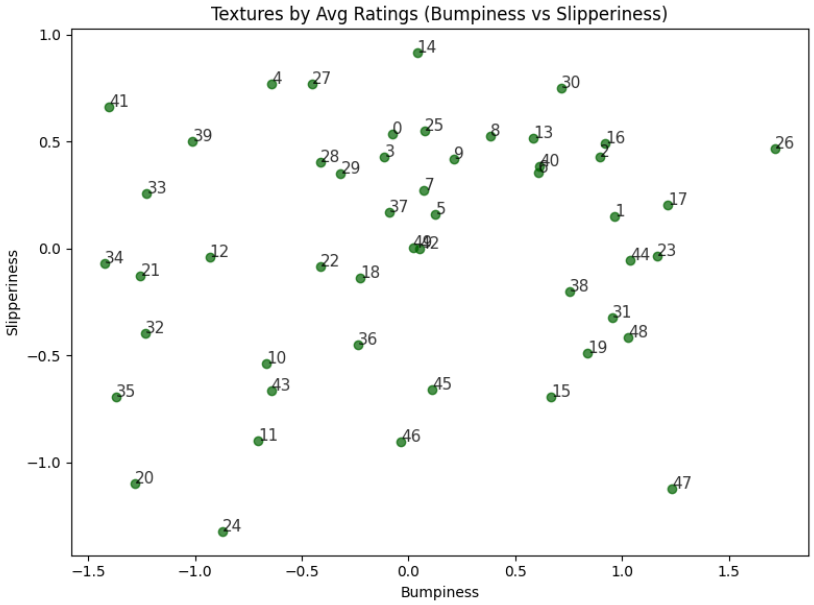


Figure C.1: Each of the fifty textures' locations on the bumpiness and slipperiness axes. Each dot represents one texture and is labeled with the number it has in the SENS3 catalog.

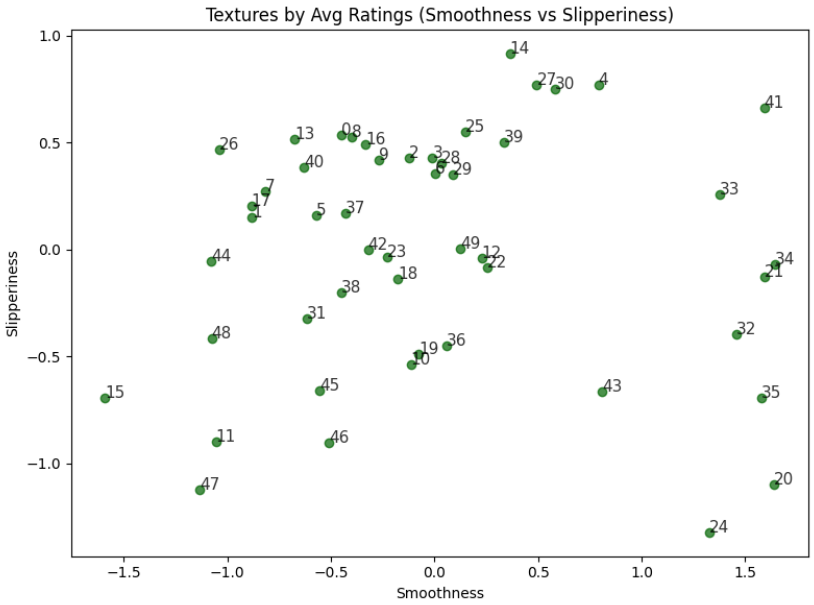


Figure C.2: Each of the fifty textures' locations on the roughness and slipperiness axes. Each dot represents one texture and is labeled with the number it has in the SENS3 catalog.

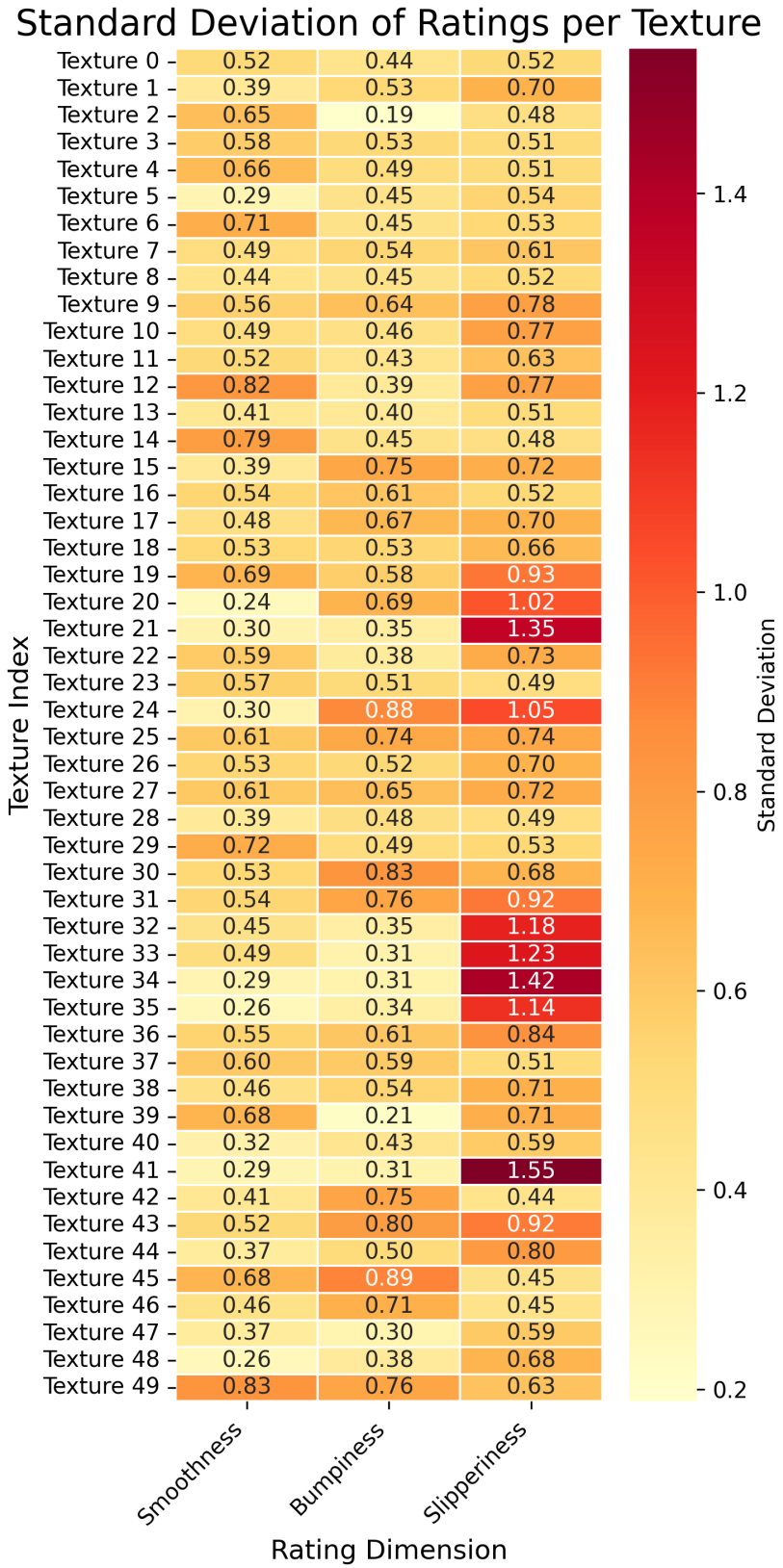


Figure C.3: Standard deviation of participant ratings for each texture across the three perceptual dimensions. Red indicates greater variability, suggesting higher disagreement among participants, while yellow indicates greater consensus. All values are based on normalized ratings.

D

More Inference Results

This appendix presents a selection of textures generated by the model for various input rating vectors. For each rating configuration, we display three generated textures to illustrate the model's variability in output given identical inputs, apart from the noise seed. The selected rating vectors include all cases where one adjective is set to either -1.0 or +1.0, while the other two are held at 0.0. Additionally, we include one case where all three adjectives are set to 0.0. This selection covers a representative range of the input space while keeping the number of signals manageable for visual inspection. Signals are shown in Figures D.1 and D.2.

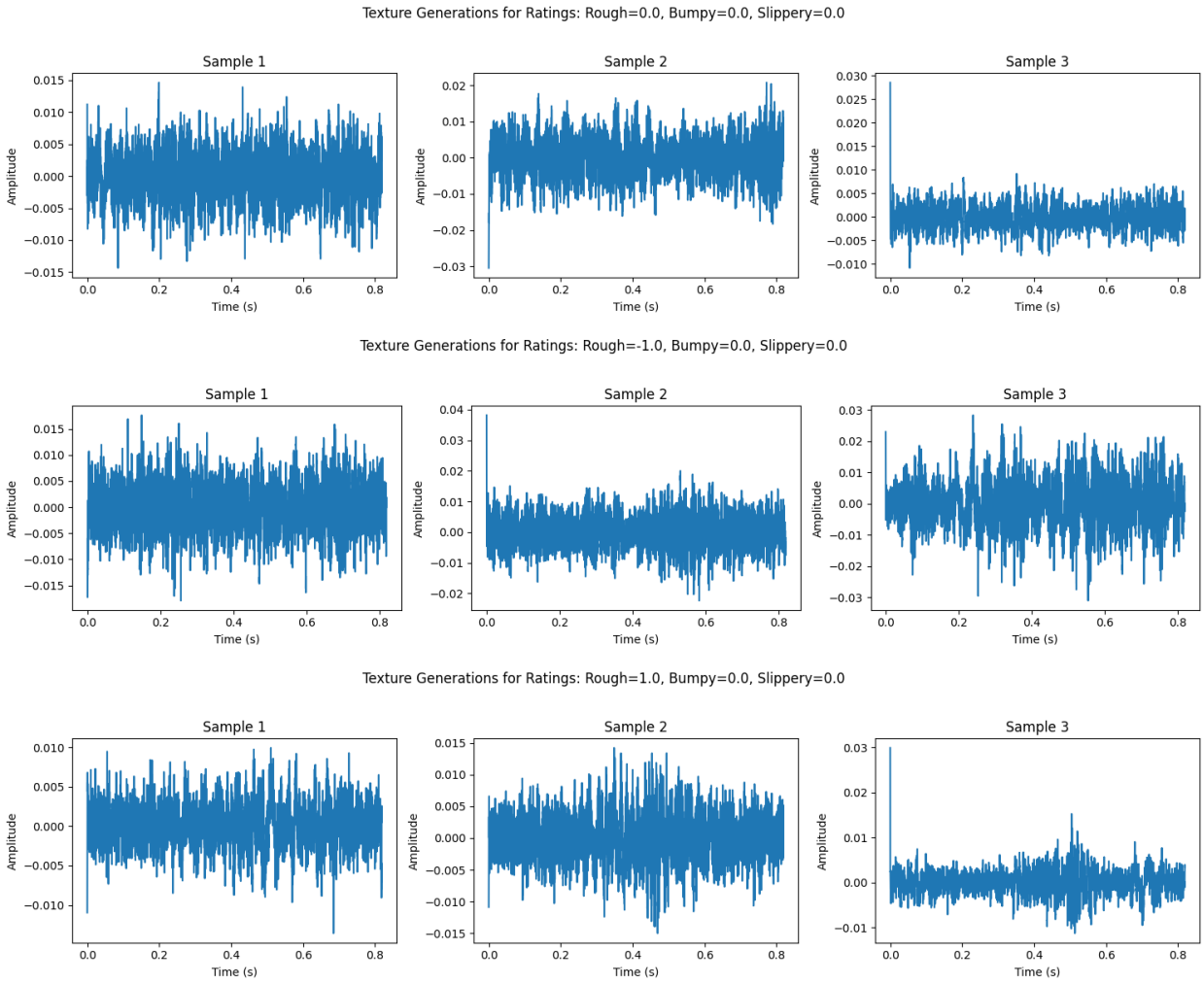


Figure D.1: Texture signals generated by our full inference pipeline. The input rating values are shown above the generated textures.

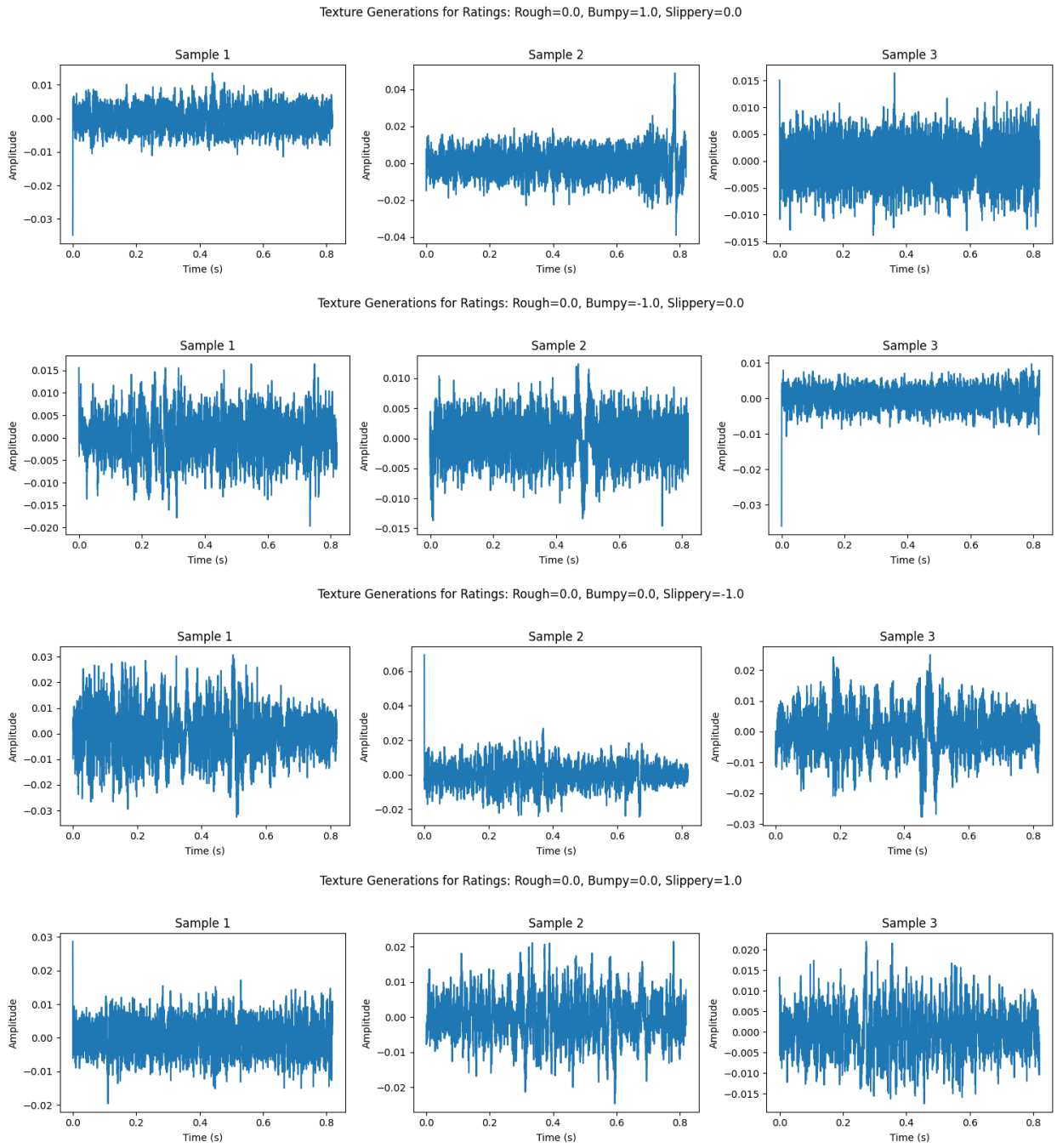


Figure D.2: Texture signals generated by our full inference pipeline. The input rating values are shown above the generated textures.

References

[1] J. K. Balasubramanian, B. L. Kodak, and Yasemin Vardar. “SENS3: Multisensory Database of Finger-Surface Interactions and Corresponding Sensations”. In: (2024), pp. 262–277. DOI: 10.1007/978-3-031-70058-3_21.

[2] F. Schneider et al. “Moûsai: Efficient Text-to-Music Diffusion Models”. In: (2024), pp. 8050–8068. DOI: 10.18653/v1/2024.acl-long.437.