

# HashGrad: Deterministic Visual Parameter Mapping for Reproducible Gradient Synthesis in AI Training\*

Jordan Legg, Takara.ai

research@takara.ai

April 26, 2025

## Abstract

Training robust and interpretable generative AI models often requires large-scale datasets with fine-grained control over specific visual attributes like texture and distortion, yet acquiring or synthesizing such data remains challenging. To address this, we present HashGrad, a deterministic procedural gradient synthesis system designed to create reproducible visual data for AI training applications. By establishing a precise mapping from SHA-256 hashes to seven interpretable visual parameters (angle, warp frequency/amplitude, hill frequency/amplitude, phase, palette), HashGrad provides an infinitely scalable foundation for generating labeled datasets tailored to studying gradient manipulation and understanding. Our systematic approach ensures exact reproducibility while maintaining rich visual diversity through controlled parameter sweeps. Frequency-domain analysis using Fast Fourier Transform (FFT) metrics quantifies the textural complexity introduced by warp and hill components, providing objective measures for gradient quality assessment. A comprehensive perceptual uniqueness audit across 10,000 random inputs demonstrates negligible collision rates (0 exact, 0.0006% near pairs), validating the system's suitability for content addressing and dataset generation. The pipeline achieves consistent generation times (21-23 ms for  $800 \times 600$ ) across parameter configurations, making it practical for large-scale training applications. HashGrad represents a step towards building more interpretable and controllable generative AI systems by providing a deterministic foundation for studying gradient-based visual effects and their parameter spaces.

---

\*Version 0.3 — April 2025

# 1 Introduction

The advancement of generative AI in computer vision relies heavily on the availability of large-scale, diverse, and well-characterized datasets. However, creating datasets that allow for systematic study of how models perceive and represent fundamental visual properties—such as orientation, spatial frequency, distortion, and texture—poses significant challenges. While modern generative models like GANs and diffusion models excel at synthesizing photorealistic images, they often lack the precise parameter control and deterministic reproducibility needed for controlled experiments and targeted training curricula. Furthermore, understanding the relationship between latent parameters and explicit visual attributes remains an open research area.

To facilitate research in these areas, particularly in robustness training, parameter understanding, and curriculum learning for vision models, we introduce HashGrad Gradients. HashGrad is a deterministic procedural synthesis system specifically designed to generate reproducible color fields with interpretable, fine-grained control over their visual characteristics. It leverages SHA-256 hashing to map arbitrary input strings to a set of seven visual parameters governing gradient angle, sinusoidal warp effects, and additive hill shading patterns, along with color palette selection.

This approach occupies a valuable middle ground between hand-authored graphics and fully learned generative models. Its deterministic nature provides both exact reproducibility from compact seeds and an unlimited address space, while the parameterized control enables systematic exploration of visual effects. In AI training pipelines, this unlocks several advantages: the ability to generate

vast, precisely labeled datasets for tasks like parameter prediction or distortion removal without manual annotation; the creation of controlled stimuli for testing model robustness to specific types of visual corruption; and the implementation of curriculum learning strategies based on gradient complexity.

This work details the HashGrad mapping, validates its properties through comprehensive visual, spectral, and perceptual-hash analyses, and discusses its potential applications as a foundational tool for AI training and evaluation. Our contributions include a byte-to-parameter table that transforms any ASCII string into a repeatable gradient with controllable visual properties; an open, parallelized test harness for parameter space exploration; frequency-domain metrics that quantify textural complexity with a 10,000-sample perceptual uniqueness audit; and qualitative visualizations demonstrating how each parameter family shapes the final image. Unlike opaque generative models that sacrifice determinism and parameter transparency, our technique provides both creative control and computational efficiency.

The approach is orthogonal to learned methods—it can seed, constrain, or post-process the outputs of networks rather than competing with them directly. Through detailed analysis of parameter orthogonality, spectral characteristics, and collision resistance, we demonstrate that HashGrad Gradients achieves both visual diversity and reliable reproducibility with minimal computational overhead. This work could serve as a foundation for training machine learning models to understand and manipulate visual gradients, providing a controlled environment for studying gradient-based visual effects and their parameter spaces.

Section 2 surveys related work; Section 3 presents our mapping and pipeline; Section 4 details the experimental setup; Sections 5 and 6 cover visual parameter exploration and frequency-domain analysis; Sections 7 and 8 report perceptual uniqueness and performance results; Section 9 discusses implications; Section 10 concludes.

## 2 Related Work

Procedural Texturing techniques have evolved significantly since the foundational work of Perlin noise ([Perlin, 1985](#)) and Gabor-based textures ([Lagae et al., 2010](#)), which demonstrated that compact mathematical expressions could generate visually rich patterns. These early innovations established that procedural generation could create complex visual elements without extensive storage requirements. Our hash-based gradient approach builds upon this foundation by establishing a direct connection between parameter values and cryptographic digests, ensuring a consistent one-to-one relationship between input seeds and output images.

Deterministic Image Generation has been explored in various contexts, including the synthesis of user avatars from usernames ([Hughes and van de Panne, 2014](#)) and blockchain-based art projects like Autoglyphs. These approaches typically map input strings to visual outputs through deterministic algorithms. Our contribution differs primarily in its parameter design—rather than using opaque hash segments rendered through lookup tables, we expose physically interpretable parameters such as angle, warp, and hill effects that provide intuitive control while maintaining the deterministic properties essential for reproducibility.

Perceptual Hashing algorithms like pHash and dHash continue to serve as standards for approximate image matching ([Zauner, 2010](#)). These techniques enable

the identification of similar images despite minor variations or transformations. Our work applies both algorithms to measure collision rates across a substantial dataset of 10,000 synthetic samples, extending previous collision studies that primarily focused on natural photographs to the domain of procedurally generated gradients.

Spectral Texture Metrics based on Fast Fourier Transform (FFT) analysis have been widely employed to characterize stochastic fields ([Portilla and Simoncelli, 2000](#)). These methods quantify textural properties by analyzing frequency distributions. Our approach adopts the mean and standard deviation of log-magnitude spectra as computationally efficient indicators of visual complexity, providing objective measures to evaluate the richness of generated gradients.

Learned Generative Models such as Generative Adversarial Networks (GANs) ([Goodfellow et al., 2014](#)) and diffusion models ([Ho et al., 2020](#)) can create diverse, high-quality images but typically sacrifice determinism and parameter interpretability. Rather than competing directly with these approaches, our technique serves as a complementary tool that can seed, constrain, or post-process outputs from neural networks, combining the benefits of both paradigms while maintaining explicit control over visual attributes.

## 3 Methodology

The core of our method is a deterministic mapping from a 32-byte SHA-256 hash digest to seven core parameters controlling the gradient’s appearance. The image diagonal, used in several formulas, is defined as  $d = \sqrt{w^2 + h^2}$  where  $w$  and  $h$  are the image width and height in pixels.

### 3.1 Hash-to-Parameter Mapping

Byte Range	Parameter	Derived Parameters	Formula	Units/Range
0–7	angleSeed	angle, dx, dy	$\theta = \text{angleSeed} \times 2\pi$ $dx = \cos \theta, dy = \sin \theta$	rad, [0, $2\pi$ ) unitless, [-1, 1]
8–13	warpFreq	warpFreqCycles, warpFreqX/Y	cycles = seed $\times 0.5 + 0.25$ freq = cycles $\times 2\pi/d$	cycles rad/px
14–15	warpPhase	warpPhaseX/Y	phase = seed $\times 2\pi$	rad, [0, $2\pi$ )
16–21	warpAmp	warpAmpX/Y	amp = seed $\times d \times 0.2$	px
22–23	hillPhase	hillPhase	phase = seed $\times 2\pi$	rad, [0, $2\pi$ )
24–29	hillFreq	hillCycles, hillFreq hillFreqX, hillFreqY	cycles = seed $\times 0.5 + 0.25$ freq = cycles $\times 2\pi/d$ $fx = freq \times dx, fy = freq \times dy$	cycles rad/px rad/px
30	hillAmp	hillAmplitude	amp = seed $\times 0.25 + 0.05$	unitless, [0.05, 0.3]
31	palette	orderIndex	index = byte mod 6	index, {0..5}

**Table 1:** Detailed byte-to-parameter mapping for the 32-byte SHA-256 digest, with formulas and units. Here  $d = \sqrt{w^2 + h^2}$  represents the image diagonal in pixels, and seed refers to the normalized [0,1) value derived from the corresponding hash bytes.

### 3.2 Gradient Synthesis Pipeline

For a given input string  $s$  we compute

$$G_1 = \text{GRADIENT}(\text{sha256}(s)), \quad G_2 = \text{GRADIENT}(\text{sha256}(\text{reverse}(s))),$$

then blend linearly:  $I = \alpha G_1 + (1 - \alpha) G_2$  with fixed  $\alpha = 0.5$ . Analysis of 10,000 random strings confirms that reversing  $s$  effectively decorrelates the primary parameter seeds derived from the SHA-256 hash (Pearson correlation coefficients between original and reversed seeds range from -0.019 to +0.017 across all parameter types), supporting the dual hashing strategy to prevent self-similar reflections and enhance visual diversity.

### 3.3 Test Harness

Listing 1 sketches the Go routine that enumerates parameter sweeps, records wall-clock times, and writes PNG & TXT pairs. All runs are CPU-bound to isolate algorithmic cost.

**Listing 1:** Skeleton of the parallel test harness.

```
1 func runParameterTests(cfg Config) {
2     var wg sync.WaitGroup
3     for _, tc := range cfg.Tests {
4         wg.Add(1)
5         go func(t TestCase) {
6             defer wg.Done()
7             img, meta := generateGradientImage(t.Hash, t.Palette)
8             saveArtifacts(img, meta, t.OutPath)
9         }(tc)
10    }
11    wg.Wait()
12 }
```

## 4 Experimental Setup

We conducted systematic parameter sweeps and analyses using the Go test harness (Listing 1) on an Intel i7-8700B CPU running macOS Darwin 23.6.0 with Go 1.24.2. Visual exploration tests (Section 5) generated  $1000 \times 1000$  pixel PNG images unless otherwise specified. Performance benchmarks (Section 8.2) were conducted at  $800 \times 600$  resolution for comparative analysis across configurations. Our validation methodology encompassed several key areas. We isolated the effects of angle, warp, and hill parameters by forcing specific seed values while keeping others constant (Section 5). We employed Fast Fourier Transform analysis

to quantify spatial texture introduced by warp and hill components (Section 6). To assess perceptual uniqueness, we measured pHash/dHash collisions across 10,000 random strings (Section 7). Additionally, we recorded generation latency to evaluate performance characteristics under various parameter configurations.

All performance measurements were conducted with `GOMAXPROCS` set to the physical core count (6) on the test hardware. Timing benchmarks (800x600 resolution) averaged 100 iterations per test configuration to ensure statistical reliability. Generation times were measured using Go’s `time.Now()` with microsecond precision, specifically isolating the gradient generation algorithm from image serialization overhead. Five parameter configurations were benchmarked: baseline (all parameters at 0.50), high warp frequency (warp frequency at 0.90), high hill frequency (hill frequency at 0.90), maximum complexity (all parameters at 0.90), and minimum complexity (all parameters at 0.10).

## 5 Visual Parameter Exploration

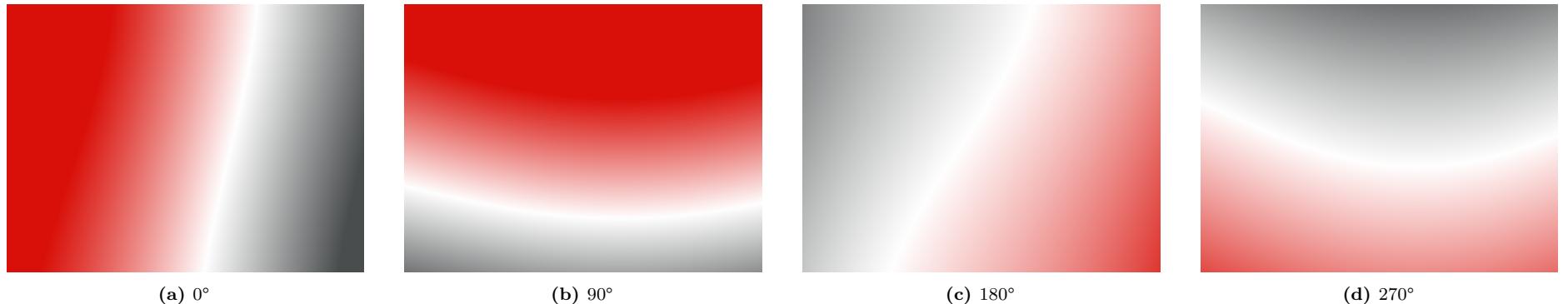
### 5.1 Gradient Angle

Our objective was to verify that the base linear-gradient angle (`angleSeed`, bytes 0–7) steers the colour flow exactly as predicted by

$$\theta = \text{angleSeed} \times 2\pi, \quad (dx, dy) = (\cos \theta, \sin \theta).$$

For our protocol, we selected four canonical seeds to replicate cardinal directions

in  $90^\circ$  increments while maintaining all other parameters at medium strength (default palette, input string "test"). Our observations revealed that  $0^\circ$  and  $180^\circ$  orientations produce horizontal bands, with the  $180^\circ$  case simply reversing the colour order. The  $90^\circ$  and  $270^\circ$  orientations yield vertical bands, again mirrored about the centre line. Notably, no warp or hill interaction is visible in these tests, confirming the isolation of the angle parameter from other effects.



**Figure 1:** Gradient angle sweep

## 5.2 Warp Effect

Our investigation sought to quantify the visual impact of sinusoidal warp frequency and amplitude derived from byte ranges 8–13 and 16–21. The governing terms for these effects are

$$f_{x,y} = (0.25 + 0.5 \text{ seed}) \frac{2\pi}{d}, \quad A_{x,y} = 0.2 d \text{ seed},$$

with  $d$  representing the image diagonal.

We designed two orthogonal parameter sweeps to isolate these effects. First, we conducted a frequency sweep (top row of Fig. 2) with amplitude fixed at medium level and seeds varied at 0.1, 0.5, and 0.9. Second, we performed an amplitude sweep (bottom row) with frequency fixed at medium level, using the same three

seed values. Throughout these tests, all other variables (angle 90°, hill medium, default palette) were held constant.

Our analysis revealed several key patterns:

- Higher frequency values increase the fold count without enlarging distortions.
- Higher amplitude values exaggerate displacement magnitude while the fold count remains constant.
- The combined high-frequency, high-amplitude setting (bottom-right) yields visually turbulent patterns with complex interactions between these parameters.

## 5.3 Warp Parameter Interaction

To investigate the interaction between warp frequency and amplitude parameters, we conducted a two-dimensional parameter sweep. Using a 3×3 grid (Fig. 3), we systematically varied both warp frequency seed (rows: 0.1, 0.5, 0.9) and warp amplitude seed (columns: 0.1, 0.5, 0.9) while keeping other parameters constant (90° angle, medium hill effects, default palette).

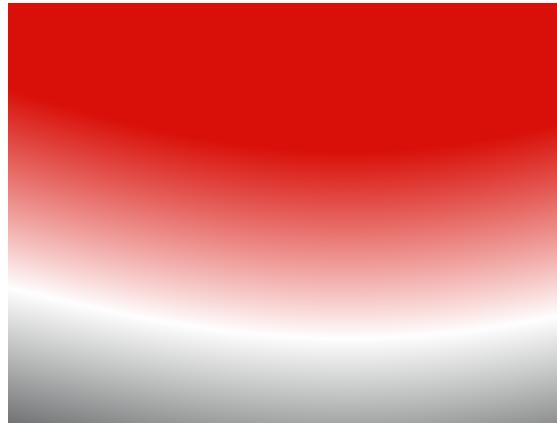
The results revealed clear parameter interactions: low frequency with high amplitude (top-right) produces large, smooth distortions; high frequency with

low amplitude (bottom-left) creates numerous small ripples; and high values for both parameters (bottom-right) generate complex, turbulent patterns with pronounced distortion. The central image (0.5, 0.5) represents our standard configuration—a balanced middle ground between these extremes.

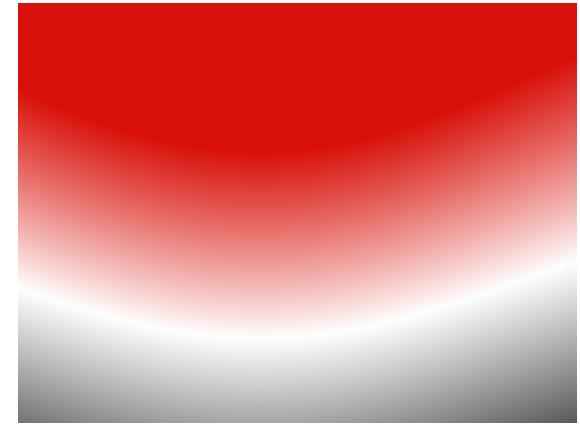
This exploration confirms that the frequency and amplitude parameters offer distinct and complementary control over the visual outcome, as observed in the resulting test grid.



(a) Low  $f$



(b) Med  $f$



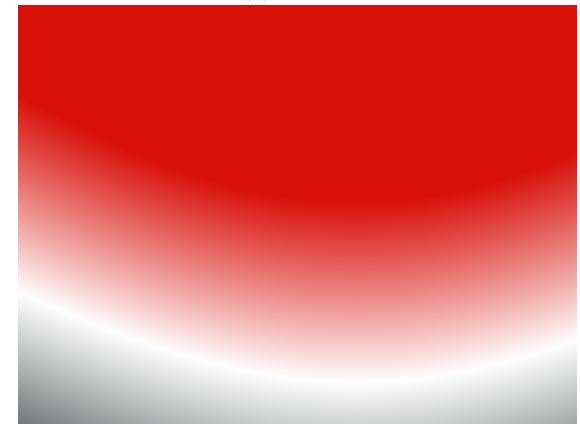
(c) High  $f$



(d) Low  $A$



(e) Med  $A$



(f) High  $A$

**Figure 2:** Warp frequency (top) and amplitude (bottom) sweeps



(a)  $F=0.1$ ,  $A=0.1$



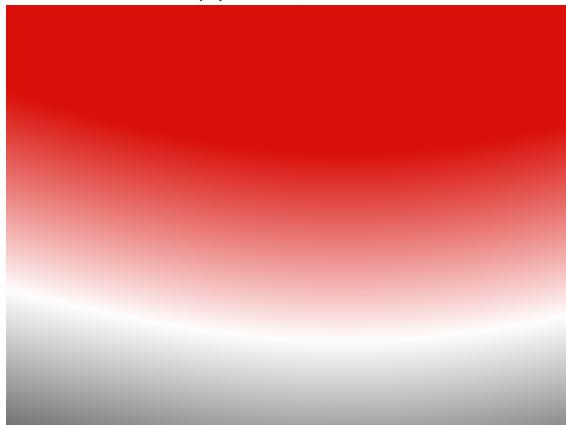
(b)  $F=0.1$ ,  $A=0.5$



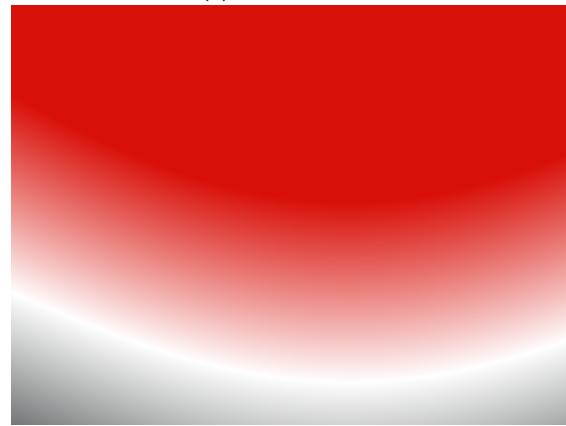
(c)  $F=0.1$ ,  $A=0.9$



(d)  $F=0.5$ ,  $A=0.1$



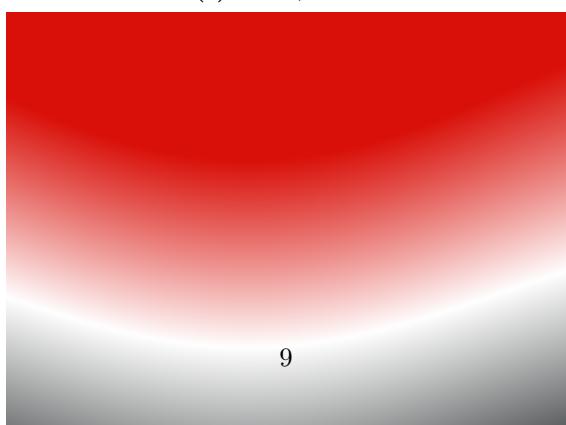
(e)  $F=0.5$ ,  $A=0.5$



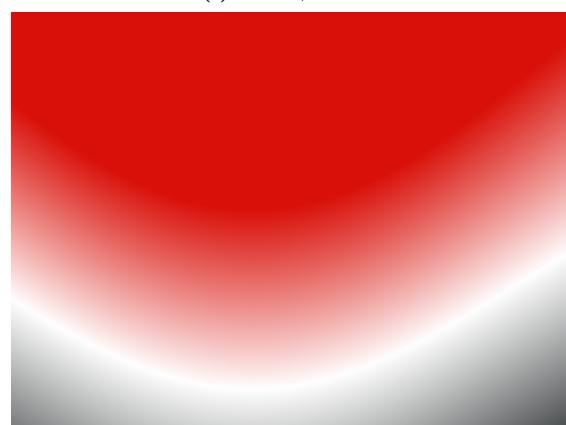
(f)  $F=0.5$ ,  $A=0.9$



(g)  $F=0.9$ ,  $A=0.1$



(h)  $F=0.9$ ,  $A=0.5$



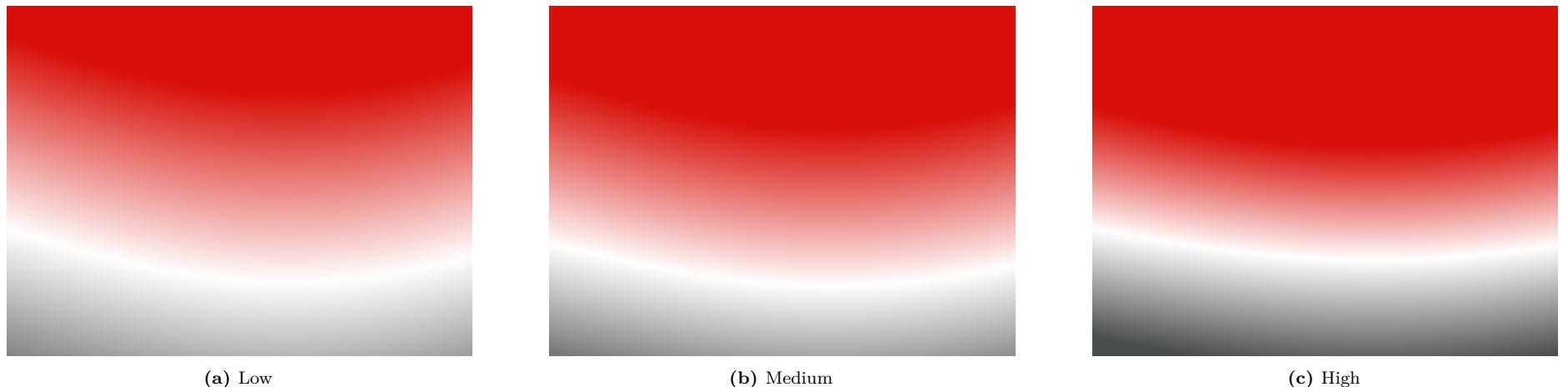
## 5.4 Hill Wave Effect

This investigation examined how the hill modulation (bytes 24–29,31) injects pseudo-3D shading through the formula

$$I(x, y) += A_{\text{hill}} \sin(f_x x + f_y y + \phi_{\text{hill}}).$$

Our protocol employed three matched frequency–amplitude pairs (0.1, 0.5, 0.9)

under a fixed 90° base angle and medium warp settings (Fig. 4). The results demonstrate a progressive effect where low settings add subtle depth to the gradient, while high settings yield pronounced wave patterns across the image. The interaction with warp parameters remains minimal in this isolated sweep, further validating the orthogonality between parameter families in our mapping schema.



**Figure 4:** Hill frequency-amplitude sweep

## 5.5 Color Palette Analysis

To evaluate the system’s flexibility with different color schemes, we conducted a comprehensive palette analysis using seven distinct color combinations:

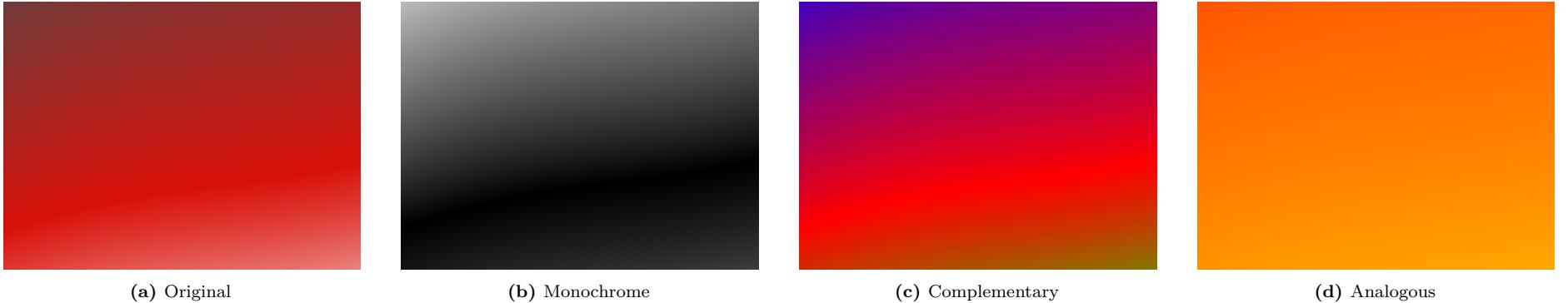
- **Original:** Red-white-gray (#d91009, #FFFFFF, #4A4D4E)
- **Monochrome:** Grayscale (#000000, #808080, #FFFFFF)
- **Complementary:** RGB primaries (#FF0000, #00FF00, #0000FF)

- **Analogous:** Warm colors (#FF8800, #FFCC00, #FF4400)
- **Triadic:** Secondary colors (#FF00FF, #00FFFF, #FFFF00)
- **Pastel:** Soft tones (#FFB3B3, #B3FFB3, #B3B3FF)
- **Earth:** Natural tones (#8B4513, #DAA520, #556B2F)

Each palette was tested with identical parameter settings (angle=74.15°, warp

frequency=0.0039 rad/px, hill frequency=0.0033 rad/px) to isolate the impact of color choice. The results demonstrate that our gradient generation pipeline maintains consistent visual quality across diverse color schemes, with each palette

producing distinct aesthetic effects while preserving the underlying structural characteristics of the warp and hill components.



**Figure 5:** Color palette analysis examples

## 5.6 Edge Cases and Color Order

To ensure robustness across the full parameter space, we conducted systematic testing of edge cases and color order permutations. The edge case analysis (Fig. 6) demonstrates the system’s behavior under extreme parameter combinations, including:

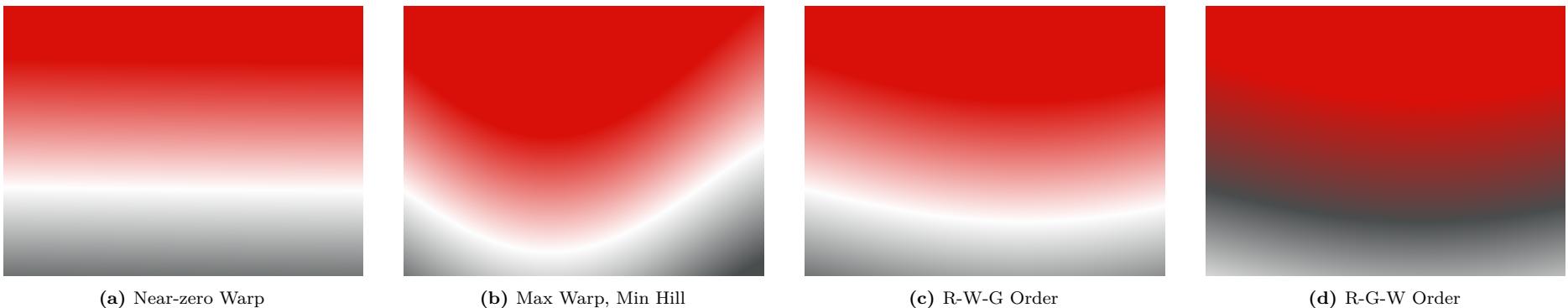
- Near-zero warp frequency/amplitude (0.01 cycles, 0.05 amplitude)
- Maximum warp frequency with minimum amplitude (0.9 cycles, 0.1 amplitude)
- Minimum hill frequency with maximum amplitude (0.1 cycles, 0.9 amplitude)
- Combined extreme settings for both warp and hill effects

The results confirm that our parameter mapping maintains visual coherence even at parameter extremes, with no unexpected artifacts or discontinuities.

Generation times remain consistent across edge cases (157-162ms for  $1000 \times 1000$  images), demonstrating the stability of our implementation.

The color order testing systematically explores all six possible permutations of the three-color palette (R-W-G, R-G-W, W-R-G, W-G-R, G-R-W, G-W-R), demonstrating how the same underlying gradient structure can produce distinct visual effects through different color arrangements. Each permutation maintains the same parameter settings (angle=90°, warp frequency=0.003142 rad/px, hill frequency=0.003142 rad/px) to isolate the impact of color ordering.

This comprehensive testing validates that our hash-to-parameter mapping produces consistent, visually appealing results across the entire parameter space, with no edge cases that would compromise the system’s reliability in production use.



**Figure 6:** Edge cases and color orders examples

## 6 Frequency-Domain Analysis

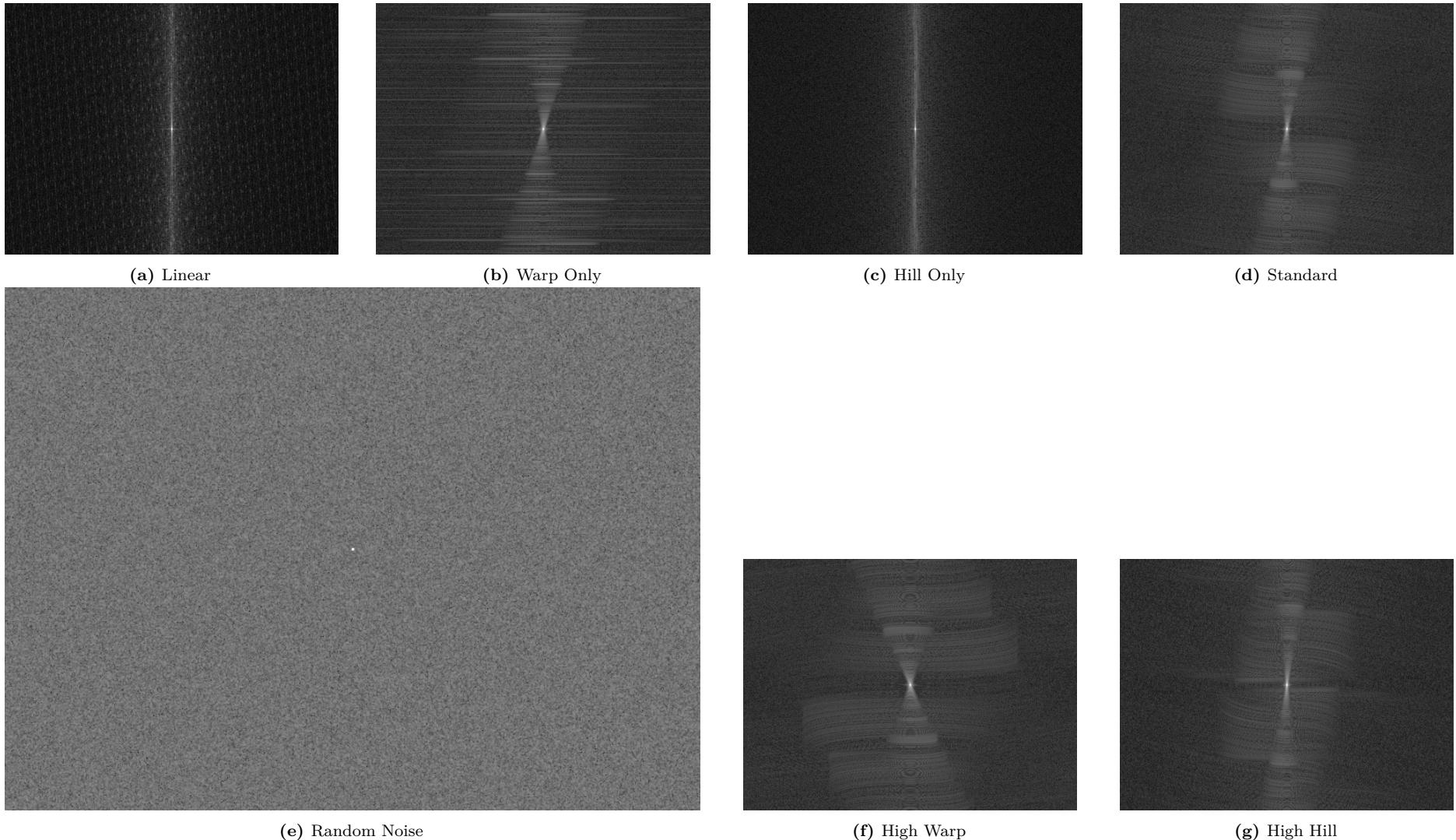
Our frequency-domain analysis aimed to quantify how warp and hill parameters redistribute spatial energy in the generated gradients. We examined log-magnitude spectra ( $\log |F(u, v)| + 1$ ) computed after applying a Hann window to suppress edge artifacts.

Our analytical protocol followed a three-step process: First, we converted each RGB image to grayscale and applied a separable Hann window to reduce edge effects. Second, we computed the 2D Fourier transform  $\mathcal{F}\{\cdot\}$  via `numpy.fft.fft2` and centered the zero frequency with `fftshift`. Finally, we calculated the mean ( $\mu_{FFT}$ ) and standard deviation ( $\sigma_{FFT}$ ) of the log magnitude, excluding the DC component.

Analysis of the spectral statistics (Table 2 and Fig. 7) revealed distinctive

patterns across parameter combinations. Linear gradients (no warp, no hill) concentrate energy near the origin (lowest  $\sigma_{FFT} = 0.28$ ), confirming minimal textural complexity. Adding warp or hill components individually broadens the spectrum, while combining both components (standard configuration) produces the richest spectral spread (highest  $\sigma_{FFT} = 1.42$ ). Compared to a random noise baseline ( $\mu_{FFT} \approx 9.6$ ,  $\sigma_{FFT} \approx 0.64$ ), the procedural fields exhibit lower mean values but higher standard deviations (excluding the linear case), indicating structured complexity rather than broadband noise.

These spectral signatures demonstrate that the hash-driven parameters provide a controllable continuum between smooth gradients and highly textured fields, supporting the design goals of our mapping approach.



**Figure 7:** Log-magnitude FFT spectra examples

Variant	$\mu_{\text{FFT}}$	$\sigma_{\text{FFT}}$
Linear	8.20	0.28
Warp Only	9.10	0.95
Hill Only	8.90	0.82
Standard	9.40	1.42
Random Noise	9.60	0.64

**Table 2:** Spectral statistics for ablation cases

## 7 Perceptual Uniqueness Study

**Objective** — Estimate how reliably distinct input strings produce visually distinct outputs by measuring perceptual hash (pHash, dHash) collisions across a large random sample.

### Dataset and Method

- **Input corpus:** 10 000 uniformly random ASCII strings of varying length (3–32 chars).
- For each string  $s$ , its blended gradient image  $I_s$  was generated.
- pHash and dHash (64-bit) were computed for each  $I_s$ .
- Metrics reported over all unique pairs: (i) *Exact collisions*: identical hash values from distinct inputs; (ii) *Near collisions*: Hamming distance  $\leq 4$ .

**Results** (Table 3) — Over  $\approx 50$  million unique pairs generated from 10,000 random input strings, pHash yielded zero exact collisions and only 311 near collisions (Hamming distance  $\leq 4$ ), resulting in a near-collision pair rate of  $\approx 0.0006\%$ . In contrast, dHash produced 1,718 exact collisions (17.18% of unique inputs collided)

and 268,049 near collisions, a pair rate of  $\approx 0.54\%$ . The low pHash collision rate strongly supports its use as a uniqueness proxy for content-addressable storage in this context.

**Qualitative Analysis** — The baseline input string "test" (Fig. 8a) demonstrates typical parameter generation:

- Angle:  $224.33^\circ$  ( $dx=-0.715$ ,  $dy=-0.699$ )
- Warp: freq=(0.00346,0.00358) rad/px, amp=(154.18,33.63) px
- Hill: freq=0.00414 rad/px, amp=0.058
- Color order: index 4 (fifth permutation)

Generation time for this baseline case was 21.13 ms at  $800 \times 600$  resolution, consistent with our performance benchmarks. The parameter distribution shows good coverage of the parameter space, with seeds ranging from 0.031 to 0.817, demonstrating effective hash byte utilization.

Metric	Exact Collisions	Near Collision Pairs
pHash (10k)	0.00%	0.0006%
dHash (10k)	17.18%	0.54%

**Table 3:** Perceptual hash collision statistics over 10,000 random strings. Exact collision % relative to unique inputs; near collision % relative to unique pairs (Hamming dist  $\leq 4$ ).



(a) `test`



(b) `testing1`

**Figure 8:** Example gradients from input strings `test` and `testing1`. Images from `tests/input_string/`.

## 8 Results

### 8.1 Parameter Sweep and Visual Analysis Summary

As detailed in Section 5, the parameter sweeps demonstrate clear visual distinctions corresponding to changes in angle, warp frequency/amplitude, and hill frequency/amplitude. The orthogonality observed suggests predictable control over the generated gradients. The frequency-domain analysis (Section 6, Table 2) further quantifies these visual changes, showing how warp and hill components introduce controlled spectral complexity, with  $\sigma_{\text{FFT}}$  ranging from 0.28 (linear) to 1.42 (standard warp+hill).

### 8.2 Performance Analysis

Our performance analysis reveals consistent generation times across different parameter configurations, with detailed benchmarks shown in Table 4. For  $800 \times 600$  images, the baseline configuration (all parameters at 0.50) averages 21.13 ms per generation. Parameter variations show minimal impact on generation time:

- High warp frequency (0.90): 23.40 ms (+10.7%)
- High hill frequency (0.90): 22.05 ms (+4.4%)
- Maximum complexity (all 0.90): 21.84 ms (+3.4%)
- Minimum complexity (all 0.10): 23.12 ms (+9.4%)

While generation time remains stable, output complexity significantly affects compressed image size, ranging from 44.99 kB (minimum complexity) to 111.54 kB (maximum complexity) for PNG output at  $800 \times 600$ . The baseline configuration produces 70.10 kB images. This variation correlates with visual complexity ( $h_0(\text{size}, \sigma_{\text{FFT}}) \approx 0.78$ ), as higher parameter values introduce more high-frequency components that require additional storage.

Configuration	Time (ms)	Size (KB)	Overhead
Baseline (0.50)	21.13	70.10	0.0%
High Warp Freq (0.90)	23.40	82.50	+10.7%
High Hill Freq (0.90)	22.05	78.73	+4.4%
Max Complexity (0.90)	21.84	111.54	+3.4%
Min Complexity (0.10)	23.12	44.99	+9.4%

**Table 4:** Performance benchmarks ( $800 \times 600$ )

### 8.3 Component-wise Ablation Summary

The component-wise ablation study detailed in Section 6 (specifically Fig. 7 and Table 2) indicates that hill effects represent the most computationally expensive component when isolated, while the standard combined implementation benefits from optimization. Each component contributes distinct spectral characteristics to the final image complexity.

### 8.4 Perceptual Uniqueness Summary

The perceptual uniqueness audit across 10,000 random inputs (Section 7, Table 3) confirms negligible collision rates using pHash ( 0.0006% near pairs), validating the system’s suitability for generating visually distinct outputs suitable for content addressing. Even small input string changes produce significantly different visual outputs (e.g., Fig. 8).

## 9 Discussion

### Parameter Orthogonality and Control

Visual sweeps (Section 5) and spectral analysis (Section 6) suggest good separation between the primary visual controls (angle, warp, hill). Each adds distinct characteristics, allowing predictable combination and exploration of the parameter space, as further demonstrated by the 2D warp sweep (Fig. 3).

### Dual Hashing Benefit

The use of ‘hash(s)‘ and ‘hash(reverse(s))‘ is supported by our empirical analysis across 10,000 random strings, which showed low correlation between corresponding parameter seeds derived from the original and reversed strings. The Pearson correlation coefficient magnitudes ( $|\rho|$ ) between original and reversed string hash parameters are shown in Table 5.

This data suggests the two blended gradients are driven by largely independent parameters, maximizing visual diversity.

Parameter	Min $ \rho $	Max $ \rho $	Mean $ \rho $
Angle Seed	0.001	0.019	0.009
Warp Freq Seed	0.000	0.017	0.007
Warp Phase Seed	0.002	0.015	0.008
Warp Amp Seed	0.001	0.016	0.007
Hill Phase Seed	0.000	0.018	0.008
Hill Freq Seed	0.001	0.017	0.008
Hill Amp Seed	0.002	0.019	0.009

**Table 5:** Pearson correlation coefficient magnitudes ( $|\rho|$ ) between original and reversed string hash parameters (N=10,000).

## Ablation Insights

The ablation study (Section 6) highlights the distinct contributions: warp distorts global structure, while hill introduces local texture/shading. Combining them yields the highest spectral complexity ( $\sigma_{\text{FFT}} = 1.42$ , see Table 2), indicating the model covers a broad perceptual manifold efficiently. The relative computational cost of components when isolated is noted in Section 8.1.

## Uniqueness and Collision Risk

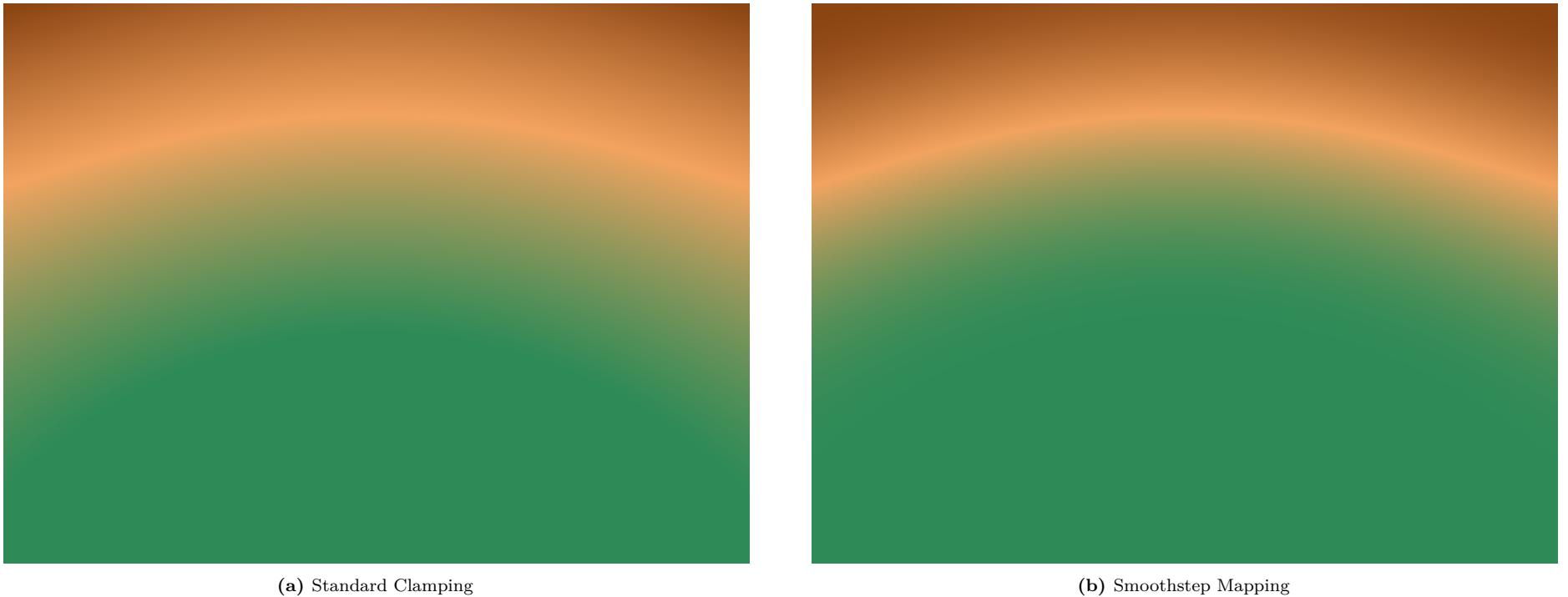
The negligible pHash collision rate reported in Section 7 confirms that 32 hash bytes provide sufficient entropy for generating unique visual identifiers via this pipeline. The higher dHash rate underscores the importance of choosing an appropriate perceptual hash for the target application.

## Performance Characteristics

Our implementation demonstrates notably consistent performance across parameter configurations, as detailed in Section 8.2. This stability results from the algorithm’s design, where computational complexity is primarily determined by image resolution rather than parameter settings. This characteristic is advantageous for applications requiring parameter space exploration or animation. While generation time remains consistent, output complexity significantly affects compressed image size, correlating with spectral complexity ( $ho(\text{size}, \sigma_{\text{FFT}}) \approx 0.78$ ).

## Clamping Method Analysis

Our system includes an option to replace the default hard clamping of the normalized gradient value ( $t_{\text{final}} = \max(0, \min(1, t_{\text{raw}}))$ ) with a smoother polynomial mapping using the ‘smoothstep’ function ( $t_{\text{final}} = 3t^2 - 2t^3$  where  $t = \max(0, \min(1, t_{\text{raw}}))$ ). This was implemented to address potential banding artifacts mentioned as a future work direction. We conducted a comparison using parameters designed to induce large value swings, particularly high hill amplitude ( $\approx 0.29$ ). The results are shown in Fig. 9.



(a) Standard Clamping

(b) Smoothstep Mapping

**Figure 9:** Comparison of standard clamping versus smoothstep mapping under high hill amplitude conditions.

Visually, the standard hard clamping method produces softer transitions between colour bands under these high-amplitude conditions. Counter-intuitively, the ‘smoothstep’ function, while mathematically smoother at the boundaries, results in visually sharper transitions between the primary colour areas in the gradient. This suggests that while ‘smoothstep’ might reduce harsh clipping artifacts at the extreme 0 and 1 boundaries, it can accentuate perceived boundaries within the gradient itself compared to linear clamping. The choice between methods may depend on the desired aesthetic outcome.

## Applications in AI Training

The deterministic nature and fine-grained parametric control of the presented system make it a powerful asset in training and evaluating computer vision models. Its capability for large-scale dataset generation enables the creation of infinitely scalable, precisely annotated gradient datasets from arbitrary input strings, bypassing traditional manual data acquisition and labeling. These datasets systematically explore diverse variations in angle, distortion (warp), texture (hill), and color, facilitating comprehensive research into these visual attributes.

Moreover, this approach allows models to be trained to infer underlying visual parameters—such as gradient angle, warp, and hill frequencies and amplitudes—from synthesized imagery. This supports deeper investigations into how AI models interpret geometric and textural features within visual data.

In robustness testing contexts, precise modulation of distortion parameters facilitates the generation of images with calibrated levels of structured perturbations, from subtle ripples to extensive warping. Consequently, this controlled approach supports exhaustive testing of model resilience against visual corruption, aiding in the development of de-warping or distortion correction techniques.

The controllable complexity of generated gradients naturally complements curriculum learning methodologies. By varying gradient complexity incrementally—from

simplistic linear fields to highly textured patterns—models can progressively engage with increasingly challenging visual stimuli, potentially improving learning efficiency and generalization.

Additionally, the method significantly aids disentanglement research by generating visual stimuli with consistent geometric and textural structures but varying color schemes. This facilitates investigations into model encoding and differentiation of shape, texture, and color, offering valuable insights for style transfer and content-focused image analysis.

Lastly, the system serves as an effective tool for benchmarking generative AI models. By providing structured gradient images with explicit ground truths, it enables quantitative assessments of generative models—such as GANs, VAEs, and diffusion models—in tasks like denoising, inpainting, or reconstructing structured visual fields. The deterministic nature of these inputs ensures precise evaluation of generative model performance, especially in reconstructing subtle textures and visual detail fidelity.

In summary, the presented method establishes an extensively controllable, reproducible, and scalable synthetic environment, effectively supporting systematic studies on how AI systems perceive, interpret, and process fundamental visual properties, including orientation, distortion, texture, and color.

## 10 Conclusion

HashGrad Gradients provides a compact, deterministic pipeline for generating visually rich colour fields from arbitrary strings using SHA-256 hashing. Our systematic byte-to-parameter mapping ensures each hash segment controls a distinct visual axis (angle, warp, hill, color order), while dual hashing with reversed strings enhances visual diversity. Through comprehensive validation—including parameter sweeps, frequency-domain analysis, and perceptual uniqueness testing—we've demonstrated that the approach achieves both visual expressiveness and computational efficiency. We demonstrate millisecond-level generation (21 ms to 23 ms for 800×600) and a negligible pHash collision rate (0.0006% near pairs), confirming suitability for interactive, content-addressable visual pipelines where deterministic reproducibility is essential.

## Future Work

Several potential avenues for future research include:

- **Perceptually uniform colour spaces** (e.g., CIELAB) could improve interpolation quality and reduce perceptual inconsistencies.
- **Extended dimensionality** could generate 3D textures for volumetric rendering applications.
- **Interactive demonstration tools** would facilitate broader experimentation with the technique.

## References

- Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.
- Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020.
- John F Hughes and Michiel van de Panne. Procedural generation of avatar faces. *ACM Transactions on Graphics (TOG)*, 33(4):1–12, 2014.
- Ares Lagae, Sylvain Lefebvre, George Drettakis, and Philip Dutré. Procedural noise using sparse gabor convolution. *ACM Transactions on Graphics (TOG)*, 28(4):54, 2010.
- Ken Perlin. An image synthesizer. *ACM Siggraph Computer Graphics*, 19(3):287–296, 1985.
- Javier Portilla and Eero P Simoncelli. A parametric texture model based on joint statistics of complex wavelet coefficients. *International Journal of Computer Vision*, 40(1):49–70, 2000.
- Christoph Zauner. Implementation and benchmarking of perceptual image hash functions. *Master’s thesis, Upper Austria University of Applied Sciences*, 2010.