TOKENIZE THE PHYSICS, NOT THE PIXELS: TESSTOK AND FOLD-FLOW QUANTIZATION FOR MATERIAL MICROSTRUCTURES

Anonymous authors
Paper under double-blind review

ABSTRACT

Visual tokenizers compress images into the tokens that latent generative models learn from. Existing tokenizers are built for natural images: a fixed latent grid, a codebook or straight-through estimator, and a perceptual objective. A physical field is instead a set of objects that a simulation reads directly. *Can a discrete tokenizer compress a material microstructure field by two orders of magnitude and return a field a crystal-plasticity solver can run?* We present **TessTok**, which encodes a polycrystal as sparse quantized Laguerre grain seeds and regrows it deterministically. Its **Fold-Flow Quantizer (FFQ)** needs no codebook or straight-through estimator and learns where to place its levels. At 10.4 kbit per 256×256 field reconstruction, TessTok reaches 24.4 dB and 8.9 rFID and matches 93% of the grains, against 22.1 dB, 45.1 and 69% for ViT-VQGAN at 13.3 kbit. At matched bits, FFQ beats every codebook and binary quantizer on all pixel and orientation metrics with no codebook to collapse. TessTok also keeps 90% of grain area at 4.4 kbit, below the rate of MAGVIT-v2. On ImageNet, with its grain-specific parts switched off, the same code gives the highest PSNR among the latest tokenizers, though not the best perceptual scores and compression rate. Tokenizing physical-object fields instead of pixels, lets our model work at many bit budgets, and lets a reconstruction be governed by the physical simulation needs.

1 INTRODUCTION

The tokenizer fixes the latent that every downstream generative model inherits (van den Oord et al., 2017; Esser et al., 2021; Rombach et al., 2022), and improving it, rather than the generator, has repeatedly moved the field: a larger lookup-free vocabulary let a masked language model overtake diffusion on ImageNet (Yu et al., 2024a), and a better bottleneck let 32 tokens suffice for an image (Yu et al., 2024b). A tokenizer is also a codec, so its quality is a rate-distortion statement, not only a perceptual one. Tokenizers are continuous or discrete, with a latent that is tied in patches to a 2D grid or a 1D sequence decoupled from position (Yu et al., 2022; Chen et al., 2025a; Bachmann et al., 2025; Chen et al., 2025b). Within discrete tokenizers the quantizer is either a learned codebook, which collapses and needs commitment and codebook losses, or a lookup-free rule, which needs a straight-through estimator and usually an entropy penalty to keep its alphabet alive (Mentzer et al., 2024; Zhao et al., 2025; Zhuang et al., 2026). Across these choices, discrete tokenizers buy compression at the price of fidelity. All of these choices were made for natural images/photographs. *For a field whose pixel values are physical quantities, what should a token be?*

Why the natural-image tokenizer fails here. A polycrystalline microstructure is an orientation map: a piecewise-constant field of eg. 440 grains in which each RGB value encodes a crystallographic orientation (Krause et al., 2026). Five mismatches follow. (i) Content is objects, not area, so a fixed latent grid spends the same budget on a coarse region and a dense one. (ii) Colour error is orientation error: a few RGB levels are several degrees of misorientation under hexagonal symmetry (Nolze & Hielscher, 2016; Patala et al., 2012). (iii) Boundaries are crisp, so blur or invented texture is not a plausible reconstruction, it is a wrong grain and therefore, a different microstructure. (iv) The consumer is a solver, not a human: the decoded map feeds a crystal-plasticity simulation inside an inverse-design or ICME loop (Roters et al., 2019; Diehl et al., 2020), which reads a label field and one orientation per grain and never reads texture, a task-grounded fidelity criterion

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

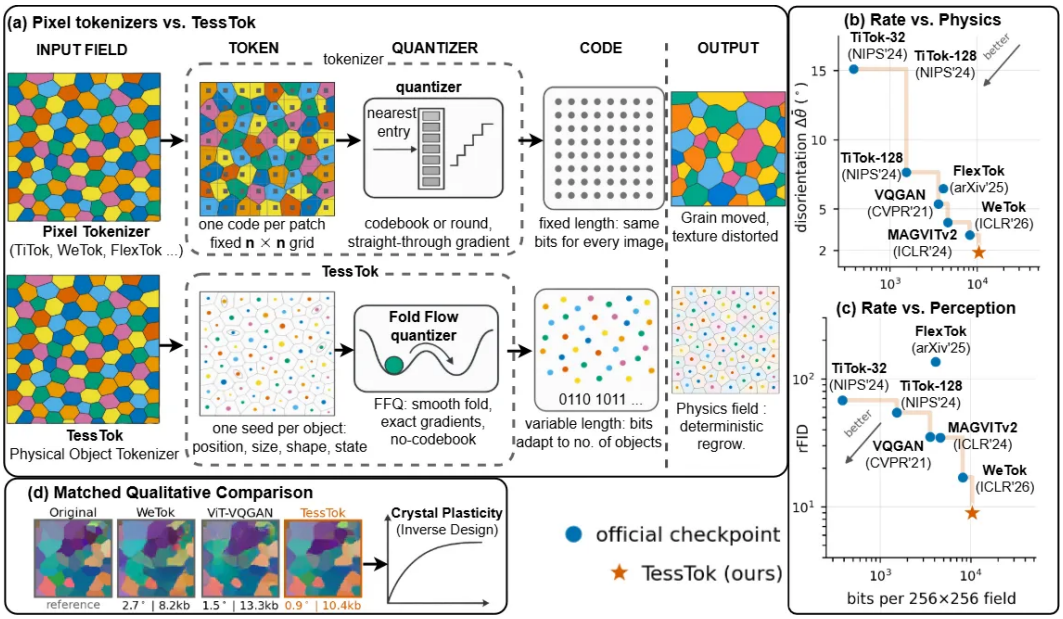

Figure 1: **What a token should be for a physical field.** (a) A pixel tokenizer sends one code per patch at a fixed rate, TessTok one seed per object at a variable rate. (b, c) Native bits against $\Delta\bar{\theta}$, the disorientation of volume-mean orientations, and against rFID of microstructure reconstructions; circles are released checkpoints, the star is TessTok and the step line is the rate-fidelity frontier. (d) One 68×68 window with per-image $\Delta\bar{\theta}$ and rate or original microstructure and it's reconstructios.

natural images lack. (v) The dataset is simulation-generated and and the code should serve as the search space of inverse design, so that 10^2 to $10^3\times$ is the operating range of blackbox/bayesian optimizers (Kusampudi & Diehl, 2023; Park et al., 2026; Zheng et al., 2026; Tripp et al., 2020).

What we do better to counter it. Each mechanism follows from a mismatch. Because of (i) and (iii) a token is a *Laguerre grain seed*, from which the decoder regrows the tessellation deterministically (Fig. 1); because a seed is a handful of clustered scalars, we quantize it with **FFQ**, an iterated smooth fold composed with a learned monotone warp; because of (i) and (v) an existence gate transmits a chosen fraction of the cells and an entropy model prices them, so one trained model can run at a range of budgets. Because of (ii) and (iv), the evaluation is done in orientation space. The seed token and FFQ are the two new parts, and the rest is deliberately ordinary so that they are what the experiments measure. On 10,100 held-out microstructures test set, TessTok gains 2.26 dB and 36.2 rFID over the strongest released baseline at 22% fewer bits. It raises matched grain area of reconstrcted microstructure from 0.69 to 0.93. With the encoder and decoder held fixed, replacing the dense VQ grid by grain seeds and their power-diagram geometry nearly halves rFID, from 13.56 to 6.99, and raises matched area from 0.912 to 0.951 (Tab. 4): the gain comes from the token definition, not from the backbone.

1. **TessTok**, a tokenizer whose tokens are the generators of a physical field rather than patches of the field: a variable-length codec whose reconstruction is a direct function of the transmitted code, and whose rate is set at inference by how many seeds it keeps, so one model traces a whole rate-fidelity curve (§3.2, Tab. 1, Fig. 3).
2. **FFQ**, a scalar quantizer with a exact gradients and no codebook, entropy penalty or straight-through estimator, of which FSQ is a limiting case (§3.3, Prop. 1, Tab. 2).
3. **A three-axis benchmark** of 29 trained or released tokenizers, quantizers and decoders on 15 magnesium-alloy families, scored by how well grain shapes and crystal orientations survive. It shows that rFID, the perceptual score this field ranks on, is a poor guide to grain fidelity, and that plain PSNR tracks orientation error more closely (App. E).
4. **Generalizability of the tokenizer contribution to natural images**: with its three microstructure-specific parts switched off, the same seed code gives the highest PSNR of

the released tokenizers on ImageNet, though not the best perceptual or rate compression scores (§3.4, Tab. 3). We do not claim state of the art there.

2 Related work

Learned vector quantization. VQ-VAE and VQGAN assign each latent to its nearest codebook entry and pay with a codebook loss, a commitment loss, a straight-through estimator (Bengio et al., 2013) and codebook collapse. A long line of work repairs the codebook with transformer ends, residual stages, a linear reparameterisation or a gradient-preserving rotation (Lee et al., 2022; Zhu et al., 2025; Fifty et al., 2025). *In contrast*, TessTok has no codebook to repair.

Lookup-free and scalar quantization. LFQ binarises each latent dimension and adds an entropy penalty to keep the alphabet used (Yu et al., 2024a); BSQ projects to a hypersphere and factorises that penalty; FSQ rounds each channel onto a small uniform grid and relies entirely on a straight-through estimator (Mentzer et al., 2024); group-wise lookup-free quantization trades group size against approximation error; lattice quantizers replace the uniform grid by a denser packing. Every one of them discretizes with a non-differentiable operation and places its levels by a fixed rule, and each needs at least one of a learned codebook, an entropy penalty or a straight-through estimator. In contrast, FFQ needs none of the three: it is smooth end to end, and its levels are learned from the data at zero side information. Tab. 5 in App. F compares them axis by axis.

Adaptive-length tokenizers and scientific codecs. TiTok, SoftVQ-VAE, FlexTok and DC-AE decouple token count from resolution, but the tokens stay abstract and the length is a design choice rather than a measured cost (Yu et al., 2024b; Chen et al., 2025a; Bachmann et al., 2025; Chen et al., 2025b). *In contrast*, TessTok's token count is set by a gate at a keep-fraction chosen at inference and every kept token is priced by an entropy model, so the length *is* the rate. Domain-grounded tokenization is established for proteins, where structures are compressed into discrete alphabets that downstream models consume directly (van Kempen et al., 2024; Wang et al., 2024), and neural image compression, where a learned entropy model turns a latent into an achievable bitstream (Ballé et al., 2016; Ballé et al., 2018). In materials science, Laguerre tessellations are the standard parametric description of a polycrystal (Aurenhammer, 1987; Lyckegaard et al., 2011; Quey et al., 2011) and are usually *fitted* to a measured microstructure. *In contrast*, we make the tessellation generator the token itself and learn to predict, quantize and entropy-code it inside a tokenizer.

3 Method

A tokenizer is a triple $T = (\mathrm{Enc}, Q, \mathrm{Dec})$: an encoder maps an image to a latent, a quantizer maps the latent to a finite code c, and a decoder reconstructs from c. We write $R(c)$ for the length of c in bits and reserve the word *rate* for R, never for a token count, because a token count hides the alphabet size. Fig. 2 shows the pipeline, App. A collects the symbols and App. B maps the properties we require of a physics-field tokenizer to the evidence for each.

3.1 Honest bit accounting

We report compression as bits per image, computed from the symbols a decoder would actually read,

$$R \;=\; \underbrace{\log_2 \binom{N}{K}}_{\text{occupancy}} \;+\; \underbrace{\textstyle\sum_{k=1}^{K} \ell(s_k)}_{\text{payload}}, \tag{1}$$

where $N{=}1024$ is the number of candidate cells, K the number of transmitted seeds, and $\ell(s_k)$ the code length our entropy model assigns to seed s_k. With $[L_{\text{off}}, L_w, L_{\text{col}}, L_{\text{shape}}] = [24, 24, 8, 12]$, a seed costs at most $2\log_2 24 + \log_2 24 + 3\log_2 8 = 22.75$ bits, plus $2\log_2 12 = 7.17$ bits for the seeds that transmit anisotropy. Our main model sends 444.1 seeds on average, so occupancy costs $1{,}005$ bits and the payload costs $11{,}823$ bits uniformly coded and $9{,}421$ bits under the entropy model, giving $\mathbf{10{,}426}$ bits against $256{\times}256{\times}3{\times}8$ bits of input, a ratio of $151\times$. Every rate here is measured this way, with hard-rounded codes and the decoder driven only by the code; we never pair a soft-quantized reconstruction with a hard-quantized rate.

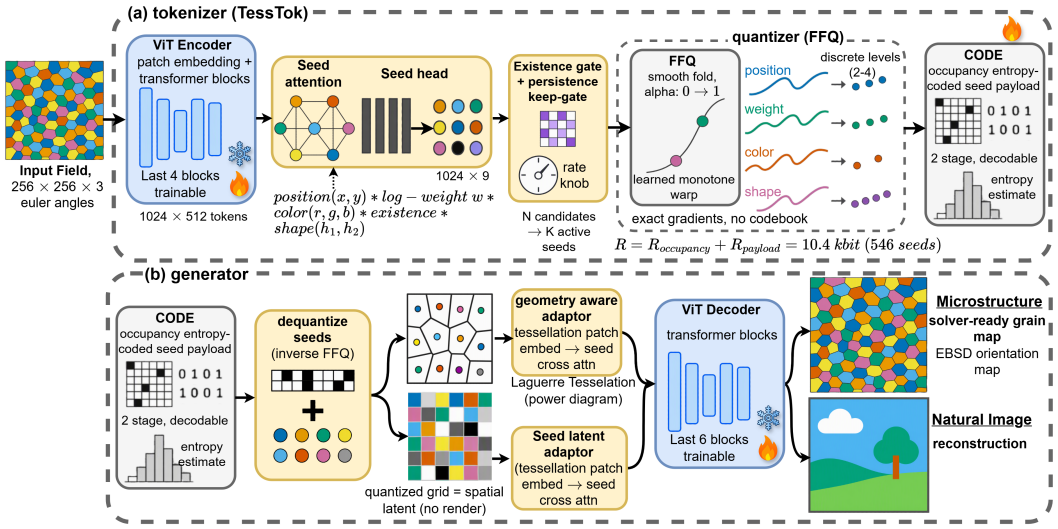

Figure 2: **TessTok, from field to seeds to bits and back.** (a) A seed generator proposes one candidate per cell of a 32×32 grid, an existence gate keeps K of $N{=}1024$, FFQ discretizes each transmitted scalar and a two-stage entropy model prices it. (b) Right of the codec boundary the decoder sees only the code: the microstructure branch regrows a power diagram and the natural-image branch reads the seed grid to decode the microstructure field and natural image, respectively.

3.2 Grain seeds as tokens

A polycrystal is, to first order, a Laguerre tessellation. Given seeds with positions $x_k \in [0,1]^2$ and weights $w_k > 0$, the cell of seed k is

$$C_k = \{p : \|p-x_k\|^2 - w_k^{\text{eff}}(p) \le \|p-x_j\|^2 - w_j^{\text{eff}}(p)\ \forall j\}, \quad w_k^{\text{eff}}(p) = w_k + a_k\cos 2\theta + b_k \sin 2\theta, \tag{2}$$

with θ the angle of $p - x_k$. The pair (a_k, b_k) turns an isotropic cell into an elongated one, which extruded alloys require. A seed is therefore a complete description of one grain: where it is, how much territory it claims, how elongated that territory is, and which orientation it carries.

A ViT encoder produces 1024 tokens; a seed-attention block lets the proposals coordinate, because neighbouring grains compete for the same territory and independent predictions duplicate or drop grains; a convolutional head emits a 32×32 field of nine channels, $(\Delta x, \Delta y, \log w, r, g, b, e, a, b)$. Offsets are bounded inside their own cell, $\log w$ to a fixed range, and the anisotropy pair is scaled by the weight so that (2) stays well posed. During training the tessellation is rendered softly, $\mathrm{softmax}(-d/\tau)$ with τ annealed to 10^{-3}; at inference it is the hard $\arg\min$ of (2), so the decoder never invents a boundary. Grain targets from connected components supervise the *raw* seed head only and never enter the reconstruction path.

3.3 FFQ: the Fold-Flow Quantizer

Rounding has zero gradient almost everywhere, so every lookup-free quantizer trains through a straight-through estimator whose backward pass is not the derivative of its forward pass. Learned codebooks avoid that but introduce collapse and two auxiliary losses. Seed marginals are also far from uniform, so a uniform level grid spends resolution where there is no data. Let u be a value in level units. The *fold*

$$\Phi_\alpha(u) = u - \frac{\alpha}{2\pi}\sin(2\pi u), \qquad \Phi'_\alpha(k) = 1-\alpha, \quad \Phi'_\alpha(k+\tfrac{1}{2}) = 1+\alpha, \quad k \in \mathbb{Z}, \tag{3}$$

is smooth, fixes every integer, contracts towards integers and expands away from half-integers. One application does not quantize; we iterate it five times, which drives a worst-case residual of 0.25 level units below 0.01. A per-channel monotone warp g, piecewise linear on 16 bins with knots from a cumulative softmax and initialised to the identity, moves the levels to where the data lives.

FFQ is their composition,

$$q(x) = g^{-1}\big(\Phi_\alpha^{(5)}\big(g(x)(L-1)\big)/(L-1)\big), \qquad \mathrm{Index}(x) = \mathrm{round}\big((L-1)\,g(x)\big), \tag{4}$$

with dequantization $g^{-1}(i/(L-1))$. A commitment potential $\sin^2(\pi(L-1)g(x))$, exactly zero on the levels, pulls the head onto the grid during training and is dropped at inference.

Proposition 1 (Homotopy and exactness). *For $\alpha \in [0,1)$ the map (4) is C^∞ in x and its gradient is the true derivative of its forward pass. At $\alpha{=}0$ it is the identity, so annealing $\alpha : 0 \to 1$ is a homotopy from no quantization to hard quantization. At $\alpha{=}1$ with $g = \mathrm{id}$ the iterated fold converges pointwise to* round *off the half-integers, so FSQ is the limiting case of FFQ with an identity warp. The warp is a global parameter held by the decoder, so learning it costs no side information.*

Proof sketch: Φ_α is a smooth perturbation of the identity by a 1-periodic sine, so integers are fixed points with multiplier $1-\alpha$ and half-integers repel with multiplier $1+\alpha$; iterating contracts every point to its nearest integer as $\alpha \to 1$. The derivation and the residual after k folds are in App. D, with the mechanism plotted in Fig. 5. FFQ costs 16 parameters per channel and five sine evaluations per scalar, with no lookup and no codebook memory.

3.4 Existence gate, rate model and objective

Each cell carries an existence logit, a binary concrete variable with annealed temperature during training (Maddison et al., 2016; Jang et al., 2017) and a hard threshold at inference. A persistence-ordered keep-gate then retains a fraction ρ of the active cells; ρ is sampled in $[0.4, 1]$ during training, so one model learns the whole rate-distortion curve and ρ becomes an inference-time knob. The payload in (1) comes from a two-stage entropy model, $p(w \mid \text{cell})\, p(\text{offset}, \text{colour}, \text{shape} \mid \text{cell}, w)$, whose context is available to the decoder in the order it decodes; anisotropy is sent only by seeds above the median weight, a rule the decoder reproduces, so it costs no signalling. The objective is one sum,

$$\mathcal{L} = \|\hat{I} - I\|_1 + \|\hat{I} - I\|_2^2 + \lambda_p \mathrm{LPIPS}(\hat{I}, I) + \lambda_r \|\tilde{I} - I\|_1 + \mathcal{L}_{\text{seed}} + \mathcal{L}_{\text{commit}} + R/\lambda_R, \tag{5}$$

with $\tilde{I}$ the rendered tessellation and LPIPS the distance of Zhang et al. (2018). What is *absent* matters as much: no codebook loss, no entropy penalty, no adversarial loss, and no orientation loss. The physical metrics of §4.1 are never optimised, which is what makes them an evaluation.

By construction the decoded field is piecewise constant with sharp interfaces, carries one orientation per grain, and is a valid label map for any code. For decoding, the hard power diagram is patch-embedded and fused with the transmitted seeds by two geometry-aware cross-attention blocks, in which each query attends to its 16 nearest seeds with a learned relative-geometry bias and a bias proportional to the Laguerre power margin of (2).

Natural-image branch. Three parts of the decoder attached to TessTok are microstructure-specific and are switched off for natural images: the power-diagram render, the power-margin bias and the connected-component grain oracle in $\mathcal{L}_{\text{seed}}$, which a self-supervised seed teacher replaces. The encoder, seed generator, FFQ, gate and entropy model keep their architecture and a render-free decoder reads the quantized seed grid, so the ImageNet experiment (Tab. 3) tests the same tokenizer rather than a second system.

4 Experiments

4.1 Setup

Data. 101,000 microstructure orientation maps from 15 extruded magnesium-alloy families, synthesised as representative volume elements with DREAM.3D (Groeber & Jackson, 2014) and stored as 300×300 RGB maps in which each grain colour is the stereographic projection of its orientation quaternion about the class-mean orientation; 90,900 train and 10,100 test, stratified by family, 440 grains per image on average. **Baselines.** On the *tokenizer axis* we run the released checkpoints of VQGAN, ViT-VQGAN, RQ-VAE, TiTok, MAGVIT-v2, WeTok, SoftVQ-VAE, FlexTok, a lattice-quantized ViT tokenizer, DC-AE and SD-VAE on our test set without adaptation. App. H repeats the whole axis with every baseline fine-tuned on our training images, with the last half of its blocks

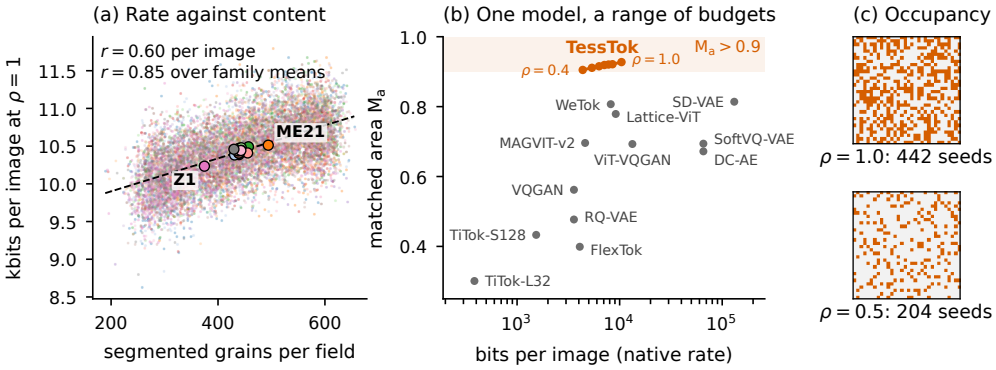

Figure 3: **One model, a range of budgets.** (a) Bits per field at $\rho = 1$ against grain count, one colour per alloy family, with family means and a least-squares fit. (b) Matched area against rate for one TessTok checkpoint swept over $\rho \in [0.4, 1]$, released tokenizers in grey. (c) Transmitted seeds of one ZX10 field at $\rho = 1.0$ and 0.5.

trainable under the same L_1 and LPIPS objective plus its own quantizer loss. On the *quantizer axis* we swap VQ, SimVQ, FSQ, LFQ, BSQ and a lattice quantizer into TessTok on the same normalised seed scalars at matched bits, leaving tokenizer and decoder untouched. On the *decoder axis* we co-train a fresh TessTok with each of 16 decoder families, so no row benefits from a frozen tokenizer; we report the most important ones here. **Rate.** We report each method's native rate, the code length its own decoder reads: tokens times $\log_2$ codebook size for the vector-quantized tokenizers, tokens times bits per token for the lookup-free ones, latent dimensions at 32-bit precision for the continuous ones, and the measured entropy-coded occupancy plus payload of Eq. 1 for TessTok. App. I gives the arithmetic. **Metrics.** Pixel: PSNR, MS-SSIM, LPIPS (Zhang et al., 2018), and rFID (Heusel et al., 2017). Grain geometry, from a Segment Anything (Kirillov et al., 2023) segmentation applied identically to originals and reconstructions: grain count and mean-area relative error ϵ_{size}. Orientation: the exact 1-Wasserstein distance W_1^{ODF} between orientation distribution functions on a Rodrigues grid with crystallographic disorientation as ground cost, the disorientation $\Delta\bar{\theta}$ between the volume-mean orientations of a field and its reconstruction, and the matched area fraction $\mathrm{M_a}$ over grains paired by intersection over union. The suite evaluates what the solver needs: the geometry terms are what a crystal-plasticity pre-processor extracts from a decoded label map, $\Delta\bar{\theta}$ is the orientation error the solver integrates, and W_1^{ODF} is the texture that sets yield anisotropy (Bunge, 1982).

The evaluation and comparison metrics. The metric this field ranks tokenizers by, rFID, predicts orientation fidelity less well than plain PSNR: over the 29 evaluated models it orders $\Delta\bar{\theta}$ at Spearman $\rho_s = 0.78$ against -0.89 for PSNR, and mean grain-size error is weaker still at 0.67, which is why our tables report matched area ($\rho_s = -0.79$) instead (Fig. 8a, b in App. E). Ranking the thirteen tokenizers of Tab. 1 by rFID instead of by disorientation moves seven of them by three places or more. With FSQ in place of FFQ, TessTok is behind on PSNR for 28% of the test fields yet ahead on disorientation for 78%, and the two differences are almost uncorrelated, $\rho_s = -0.10$ (Fig. 8c). A decibel is not a grain, so every comparison below reports the grain and orientation metrics beside the pixel ones. Every method is evaluated on the same test set, resolution, segmenter and metric code, with every rate in bits, so the comparison evaluates the representation rather than the recipe. Training uses four H200 GPUs with Adam (App. C), and every number is a single training run, since the three axes already take more than thirty.

4.2 Tokenizers and quantizers at $10^2\times$ compression

Tokenizer axis. Among the discrete tokenizers of Tab. 1 TessTok gives the best value of every column, and only ViT-VQGAN spends more bits. Against ViT-VQGAN, the strongest released baseline on PSNR and the one that shares our encoder and decoder ends, it gains 2.26 dB and 36.2 rFID at 22% fewer bits, cuts $\Delta\bar{\theta}$ from $3.16°$ to $1.85°$ and raises matched grain area from 0.693 to 0.927. Only SD-VAE, a continuous tokenizer at $12.6\times$ our rate, is ahead on any column, by 0.21 dB of PSNR.

Table 1: Tokenizer axis on the 10,100 test fields, released checkpoints without adaptation (fine-tuned in App. H). Angles in degrees; continuous tokenizers below the rule.

Method	Bits	PSNR↑	MS-SSIM↑	LPIPS↓	rFID↓	W_1^{ODF}↓	$\Delta\bar{\theta}$↓	$\mathrm{M_a}$↑
TiTok-L32 *(NeurIPS'24)*	384	15.15	0.238	0.538	67.7	15.80	15.11	0.301
TiTok-S128 *(NeurIPS'24)*	1,536	15.83	0.387	0.458	54.4	10.73	7.65	0.433
RQ-VAE *(CVPR'22)*	3,584	17.44	0.506	0.379	89.7	8.76	4.90	0.477
VQGAN *(CVPR'21)*	3,584	17.71	0.511	0.375	35.1	9.44	5.34	0.562
FlexTok *(ICML'25)*	4,087	15.45	0.368	0.446	135.6	9.17	6.45	0.399
MAGVIT-v2 *(ICLR'24)*	4,608	18.80	0.652	0.290	34.8	8.43	4.01	0.696
WeTok *(ICLR'26)*	8,192	21.05	0.787	0.192	16.9	6.97	3.11	0.807
Lattice-ViT	9,162	20.67	0.760	0.210	39.4	6.78	2.85	0.779
ViT-VQGAN *(ICLR'22)*	13,312	22.10	0.824	0.169	45.1	6.54	3.16	0.693
TessTok (ours)	**10,426**	**24.36**	**0.908**	**0.105**	**8.93**	**4.89**	**1.85**	**0.927**
SoftVQ-VAE *(CVPR'25)*	65,536	20.08	0.694	0.246	27.1	6.82	3.08	0.694
DC-AE *(ICLR'25)*	65,536	20.24	0.698	0.242	54.6	6.29	3.00	0.672
SD-VAE *(CVPR'22)*	131,072	24.57	0.880	0.109	14.2	4.91	1.85	0.814

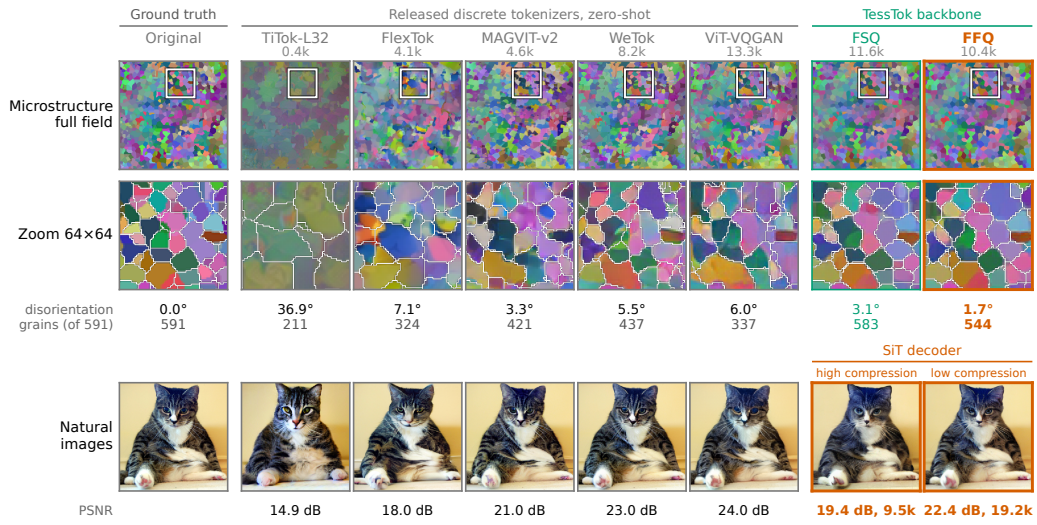

Figure 4: **What survives the bottleneck.** Top: a ZX10 Mg-alloy field at its family median of TessTok disorientation, with a zoom window chosen from the ground truth. Middle: the zoom with segmented grain boundaries, mean disorientation and grain count. Bottom: an ImageNet test image with PSNR; the last two tiles are TessTok with the SiT decoder at 9.5 and 19.2 kbit.

No released tokenizer returns a field a solver could use, since the best of them mislays 19% of the grain area. Read against the fine-tuned protocol of App. H, the margin over ViT-VQGAN narrows to 0.75 dB and 4.7 rFID and fine-tuned SD-VAE overtakes TessTok at 12.6× the rate, yet the ordering among discrete tokenizers survives (Tab. 7). Fig. 1(b, c) plots this frontier and Fig. 4 serves as the qualitative example on one field.

Quantizer axis at matched bits. Tab. 2 swaps the quantizer inside TessTok and leaves everything else fixed. The family matters more than the tuning: on a handful of scalars per seed, every codebook, binary and lattice quantizer trails FSQ by 1.5 to 4.4 dB at the same budget, because a binary code needs many dimensions before its approximation error becomes small, and SoftVQ trails by 7.3 dB, since its soft posterior cannot be read back from a hard code.

Among scalar quantizers what is left to win is level placement. Under the full schedule FFQ leads FSQ on every pixel column at 10% fewer bits and cuts $\Delta\bar{\theta}$ from 2.90° to 1.85°, while FSQ keeps a small lead on rFID and matched area; the FSQ row was trained for 60 epochs, so this comparison is not budget matched. The case for FFQ is a better operating point reached without the straight-through estimator, at 16 parameters per channel, not a large single-number gain.

Table 2: Quantizer axis: only the quantizer inside TessTok is swapped, at a matched 14-bit alphabet per seed. Rows above the rule share a 60-epoch schedule; FFQ, below it, is the full schedule. STE: straight-through estimator.

Quantizer	Codebook	STE	PSNR↑	MS-SSIM↑	LPIPS↓	rFID↓	W_1^{ODF}↓	$\Delta\bar{\theta}$↓	$\mathrm{M_a}$↑
SoftVQ *(CVPR'25)*	learned	soft	17.01	0.633	0.304	101.1	7.88	4.07	0.849
LFQ *(ICLR'24)*	none	yes	19.92	0.699	0.239	58.3	7.36	3.83	0.925
RVQ *(CVPR'22)*	learned	yes	20.23	0.714	0.240	55.4	7.22	3.35	0.913
BSQ *(ICLR'25)*	none	yes	22.03	0.820	0.171	31.7	6.46	2.74	0.928
Lattice	none	yes	22.06	0.825	0.169	27.2	6.21	2.62	0.928
SimVQ *(ICCV'25)*	learned	yes	22.54	0.839	0.158	14.4	6.30	2.86	0.938
VQ *(NeurIPS'17)*	learned	yes	22.77	0.847	0.151	12.3	6.19	2.81	0.941
FSQ *(ICLR'24)*	none	yes	24.29	0.892	0.111	**8.44**	6.04	2.90	**0.945**
FFQ (ours), full schedule	**none**	**no**	**24.36**	**0.908**	**0.105**	8.93	**4.89**	**1.85**	0.927

4.3 Rate control, decoder choice and generality

One model, a range of budgets. Sweeping the keep-fraction ρ of §3.4 from 1.0 to 0.4 at inference traces a rate-fidelity curve from a *single* trained model, where each fixed-rate tokenizer gives one point per training run (Fig. 3b). At $\rho = 0.4$ TessTok spends $4{,}365$ bits, fewer than MAGVIT-v2, and keeps $\mathrm{M_a} = 0.905$, above every baseline at any rate. Texture gives way before the grain map: rFID rises from 8.9 to 20.5 while $\mathrm{M_a}$ falls by 2.4%, because the seeds the gate drops carry little area and their cells pass to their neighbours (Fig. 3c). The budget is set by ρ rather than by the field, so at fixed ρ bits track grain count only through the entropy model (r=0.60, Fig. 3a; App. I).

Deterministic decoding wins. The upper block of Tab. 4 adds mechanisms to the dense baseline that shares our ViT ends: grain seeds, the geometry-aware adaptor and the Laguerre power-margin bias together nearly halve rFID, from 13.56 to 6.99, and add 0.039 of matched area, and the rate budget then buys 22% fewer bits than the dense code for 1.9 rFID while improving both orientation metrics. Every generative decoder in the lower block of Tab. 4 is worse than the deterministic one on every physical metric: the best, SDXL-VAE, loses 3.6 dB and 6 points of matched area, and diffusion and autoregressive heads lose more. A generative decoder samples a plausible field, and at this compression a hallucinated boundary is a new integration point for the solver. The same held on ImageNet, where a SiT rectified-flow decoder of 890M parameters trails the deterministic one on every column at the high rate despite 40% more bits (Tab. 3).

Natural images. With its three microstructure-specific parts switched off, the seed code is a competitive pixel codec on ImageNet (Tab. 3): at 13.8 kbit the ViT-decoder model gives the highest PSNR of any row, 24.41 dB, and beats ViT-VQGAN, which shares its encoder and decoder ends, on rFID, while WeTok at 8.2 kbit stays ahead on LPIPS and rFID. The test images come from the ImageNet training split that the released checkpoints were trained on, which favours them (App. J).

5 Conclusion

A tokenizer inherits the assumptions of the data it was designed for. For photographs those are a fixed grid, an abstract code and a perceptual criterion; for a physical field they should be the field's own objects, a rate set by how many of them are sent, and a fidelity criterion taken from whatever consumes the output. The three axes exemplifies it: every fixed-grid tokenizer below 10 kbit loses grains outright, once the token is a physical scalar the quantizer's only job is level placement, which FFQ does without a codebook, entropy penalty or straight-through estimator, and a field bound for a solver wants a deterministic decoder, the opposite of where natural-image tokenizers are heading (cost in App. L). One checkpoint also serves a range of budgets and loses texture before grains: kept to 40% of its seeds it spends 4.4 kbit, fewer than MAGVIT-v2, and still matches 90% of grain area, more than any released tokenizer at any rate. That makes a TessTok code a design variable for an inverse-design loop rather than an archive format. On ImageNet the seed code is a competitive pixel codec but not a perceptual one: the mechanisms travel, while the large margins belong to object-structured fields.

Table 3: ImageNet reconstruction on 10,090 held-out images, in three blocks: released discrete tokenizers, TessTok at the lowest and highest rate of each decoder (each row a separate training, K its mean seed count) and a continuous baseline. Bold marks the best value within a block and * the best discrete value overall.

Method	Bits	K	PSNR↑	MS-SSIM↑	LPIPS↓	rFID↓
Released discrete tokenizers, zero-shot						
TiTok-L32 *(NeurIPS'24)*	384	–	16.35	0.511	0.318	5.46
TiTok-S128 *(NeurIPS'24)*	1,536	–	17.87	0.650	0.214	4.02
RQ-VAE *(CVPR'22)*	3,584	–	20.27	0.770	0.169	8.66
VQGAN *(CVPR'21)*	3,584	–	20.58	0.775	0.164	8.93
FlexTok *(ICML'25)*	4,087	–	19.06	0.706	0.212	3.76
MAGVIT-v2 *(ICLR'24)*	4,608	–	21.99	0.846	0.118	3.13
WeTok *(ICLR'26)*	8,192	–	23.72	0.898	**0.085***	**1.97***
Lattice-ViT	9,162	–	22.99	0.877	0.095	2.40
ViT-VQGAN *(ICLR'22)*	13,312	–	**24.27**	**0.916***	0.098	4.93
TessTok (ours), decoder named						
SiT-RF decoder	9,499	870	20.39	0.804	0.197	10.15
SiT-RF decoder	19,248	1,024	23.16	0.883	0.128	4.72
ViT decoder	8,509	1,024	22.76	0.864	0.190	12.80
ViT decoder	13,762	1,024	**24.41***	**0.910**	**0.120**	**4.01**
Continuous						
SoftVQ-VAE *(CVPR'25)*	65,536	–	23.16	0.870	0.093	2.00

Table 4: Mechanisms added cumulatively to the dense baseline (upper) and decoders co-trained with a fresh tokenizer per row (lower), all trained on our data. The deterministic decoder row is the 14.3 kbit variant of Tab. 11.

Configuration	Bits	PSNR↑	MS-SSIM↑	rFID↓	W_1^{ODF}↓	$\Delta\bar{\theta}$↓	$\mathrm{M_a}$↑
Mechanisms, cumulative							
Dense grid + VQ (ViT-VQGAN ends)	13,312	23.61	0.881	13.56	5.45	2.33	0.912
+ Laguerre power-margin bias	–	23.78	0.883	6.99	6.38	2.66	**0.951**
+ extreme rate budget (reported)	**10,426**	**24.36**	**0.908**	8.93	**4.89**	**1.85**	0.927
Decoder swapped, tokenizer co-trained per row							
ViT, deterministic (ours)	–	**24.75**	**0.918**	**5.66**	**4.87**	**1.80**	**0.942**
SDXL-VAE *(ICLR'24)*, VAE	–	21.12	0.803	34.8	7.16	3.05	0.884
BigGAN *(ICLR'19)*, GAN	–	18.99	0.743	19.7	6.50	3.71	0.812
SD3.5 MMDiT *(ICML'24)*, diffusion	–	18.82	0.656	15.3	6.47	2.85	0.737
MAR-L *(NeurIPS. 2024)*, autoregressive	–	17.55	0.553	105.2	6.75	3.26	0.625
DiT-XL *(ICCV'23)*, diffusion	–	17.06	0.444	55.1	9.10	6.50	0.576
MaskGIT *(CVPR'22)*, masked LM	–	15.46	0.350	158.3	9.65	7.58	0.286

Limitations. The grid is set by the smallest grain, grains below two cells share a seed with a neighbour, so the finest family, ME21 at 494 grains per field has the lowest matched area, 0.912 against 0.941 for the coarsest (App. H). The gate does not yet allocate seeds by content: at $\rho = 1$ it keeps 444 ± 12 seeds whatever the field holds, so bits follow grain count only through the entropy model, and below $\rho = 0.9$ each further step of ρ costs more matched area than the last (App. I). Convex cells return a concave grain, such as a recrystallised necklace, with a straightened boundary (App. K). Colour is constant within a cell, which suits the undeformed grain maps a solver takes as input but cannot represent intragranular gradients; crystal symmetry enters the metrics but not the colour space, and the 8-bit orientation encoding puts a floor under every metric. We test one hexagonal material system in two dimensions; three-dimensional power diagrams are a direct extension of Eq. (2) but untested, and other crystal systems will need retraining. On ImageNet the test images come from the training split of the released checkpoints, which favours them, and no TessTok point matches WeTok's LPIPS or rFID. TessTok in future will be generalized to fields without object structure, such as turbulence or diffuse-interface phase fields. App. M collects each limitation with its evidence and every negative result.

Reproducibility statement

The tokenizer, quantizer, rate model, evaluation suite and run drivers for all three axes are released in the anonymous repository linked in the supplementary material. App. C gives the hyperparameter table, the data split script, the segmentation checkpoint, seeds and hardware; App. E gives the formulas and reference implementation of every metric; App. D gives the derivation behind Prop. 1. Every number is produced by the released pipeline from the released checkpoints.

AI use statement

We used generative AI tools for code editing, for drafting and copy-editing text, and for figure layout. We did not use them to generate data, to develop the theory, to formulate or prove mathematical claims, or to design the experiments. All code was verified and tested by the authors, all references were checked manually against primary sources, and all derivations were checked. We take responsibility for the final content.

Ethics statement

The microstructures are synthetic and carry no personal or proprietary information. The intended use, faster inverse design of structural alloys, carries the general dual-use risk of any materials-design tool and no specific risk we are aware of.

References

F. Aurenhammer. Power diagrams: Properties, algorithms and applications. *SIAM Journal on Computing*, 16(1):78–96, 1987. doi: 10.1137/0216006. URL https://doi.org/10.1137/0216006.

Roman Bachmann, Jesse Allardice, David Mizrahi, Enrico Fini, Oğuzhan Fatih Kar, Elmira Amirloo, Alaaeldin El-Nouby, Amir Zamir, and Afshin Dehghan. Flextok: Resampling images into 1d token sequences of flexible length. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=DgdOkUUBzf.

Johannes Ballé, Valero Laparra, and Eero P. Simoncelli. End-to-end optimized image compression. *ArXiv*, abs/1611.01704, 2016. URL https://api.semanticscholar.org/CorpusID:6775391.

Johannes Ballé, David Minnen, Saurabh Singh, Sung Jin Hwang, and Nick Johnston. Variational image compression with a scale hyperprior. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=rkcQFMZRb.

Yoshua Bengio, Nicholas Léonard, and Aaron C. Courville. Estimating or propagating gradients through stochastic neurons for conditional computation. *ArXiv*, abs/1308.3432, 2013. URL https://api.semanticscholar.org/CorpusID:18406556.

Hans-Joachim Bunge. *Texture Analysis in Materials Science: Mathematical Methods*. Butterworths, London, 1982.

Hao Chen, Ze Wang, Xiang Li, Ximeng Sun, Fangyi Chen, Jiang Liu, Jindong Wang, Bhiksha Raj, Zicheng Liu, and Emad Barsoum. Softvq-vae: Efficient 1-dimensional continuous tokenizer. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 28358–28370, 2025a. doi: 10.1109/CVPR52734.2025.02641.

Junyu Chen, Han Cai, Junsong Chen, Enze Xie, Shang Yang, Haotian Tang, Muyang Li, and Song Han. Deep compression autoencoder for efficient high-resolution diffusion models. In *The Thirteenth International Conference on Learning Representations*, 2025b. URL https://openreview.net/forum?id=wH8XXUOUZU.

Martin Diehl, Ding Wang, Chuanlai Liu, Jaber Rezaei Mianroodi, Fengbo Han, Duancheng Ma, Piet J. J. Kok, Franz Roters, and Pratheek Shanthraj. Solving material mechanics and multiphysics problems of metals with complex microstructures using DAMASK—the Düsseldorf advanced

material simulation kit. *Advanced Engineering Materials*, 22(5):190044, 2020. doi: 10.1002/adem.201901044.

Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming transformers for high-resolution image synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12873–12883, June 2021.

Christopher Fifty, Ronald Guenther Junkins, Dennis Duan, Aniketh Iyengar, Jerry Weihong Liu, Ehsan Amid, Sebastian Thrun, and Christopher Re. Restructuring vector quantization with the rotation trick. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=GMwRl2e9Y1.

Michael A. Groeber and Michael A. Jackson. DREAM.3D: A digital representation environment for the analysis of microstructure in 3D. *Integrating Materials and Manufacturing Innovation*, 3 (1):56–72, 2014. doi: 10.1186/2193-9772-3-5.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Neural Information Processing Systems*, 2017. URL https://api.semanticscholar.org/CorpusID:326772.

Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=rkE3y85ee.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014. URL https://api.semanticscholar.org/CorpusID:6628106.

Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollar, and Ross Girshick. Segment anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4015–4026, October 2023.

Maximilian Krause, Thomas Böhlke, and Matti Schneider. Generating high-fidelity microstructures of polycrystalline materials with prescribed higher-order texture tensors. *Computer Methods in Applied Mechanics and Engineering*, 452:118690, 2026. ISSN 0045-7825. doi: https://doi.org/10.1016/j.cma.2025.118690. URL https://www.sciencedirect.com/science/article/pii/S0045782525009624.

Navyanth Kusampudi and Martin Diehl. Inverse design of dual-phase steel microstructures using generative machine learning model and bayesian optimization. *International Journal of Plasticity*, 171:103776, 2023. ISSN 0749-6419. doi: https://doi.org/10.1016/j.ijplas.2023.103776. URL https://www.sciencedirect.com/science/article/pii/S0749641923002607.

Doyup Lee, Chiheon Kim, Saehoon Kim, Minsu Cho, and Wook-Shin Han. Autoregressive image generation using residual quantization. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11513–11522, 2022. doi: 10.1109/CVPR52688.2022.01123.

Allan Lyckegaard, Erik Lauridsen, Wolfgang Ludwig, Richard Fonda, and H.F. Poulsen. On the use of laguerre tessellations for representations of 3d grain structures. *Advanced Engineering Materials*, 13:165 – 170, 03 2011. doi: 10.1002/adem.201000258.

Chris J. Maddison, Andriy Mnih, and Yee Whye Teh. The concrete distribution: A continuous relaxation of discrete random variables. *ArXiv*, abs/1611.00712, 2016. URL https://api.semanticscholar.org/CorpusID:14307651.

Fabian Mentzer, David Minnen, Eirikur Agustsson, and Michael Tschannen. Finite scalar quantization: VQ-VAE made simple. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=8ishA3LxN8.

G. Nolze and R. Hielscher. Orientations – perfectly colored. *Journal of Applied Crystallography*, 49:1786–1802, 2016. doi: 10.1107/S1600576716012942.

Jaewan Park, Shashank Kushwaha, Junyan He, Seid Koric, Qibang Liu, Iwona Jasiuk, and Diab Abueidda. Nonlinear inverse design of mechanical multi-material metamaterials enabled by video denoising diffusion and structure identifier. *Engineering Applications of Artificial Intelligence*, 172:114368, 2026. ISSN 0952-1976. doi: https://doi.org/10.1016/j.engappai.2026.114368. URL https://www.sciencedirect.com/science/article/pii/S0952197626006494.

S. Patala, J. K. Mason, and C. A. Schuh. Improved representations of misorientation information for grain boundary science and engineering. *Progress in Materials Science*, 57:1383–1425, 2012. doi: 10.1016/j.pmatsci.2012.04.002.

R. Quey, P.R. Dawson, and F. Barbe. Large-scale 3d random polycrystals for the finite element method: Generation, meshing and remeshing. *Computer Methods in Applied Mechanics and Engineering*, 200(17):1729–1745, 2011. ISSN 0045-7825. doi: https://doi.org/10.1016/j.cma.2011.01.002. URL https://www.sciencedirect.com/science/article/pii/S004578251100003X.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10684–10695, June 2022.

F. Roters, M. Diehl, P. Shanthraj, P. Eisenlohr, C. Reuber, S.L. Wong, T. Maiti, A. Ebrahimi, T. Hochrainer, H.-O. Fabritius, S. Nikolov, M. Friák, N. Fujita, N. Grilli, K.G.F. Janssens, N. Jia, P.J.J. Kok, D. Ma, F. Meier, E. Werner, M. Stricker, D. Weygand, and D. Raabe. Damask – the düsseldorf advanced material simulation kit for modeling multi-physics crystal plasticity, thermal, and damage phenomena from the single crystal up to the component scale. *Computational Materials Science*, 158:420–478, 2019. ISSN 0927-0256. doi: https://doi.org/10.1016/j.commatsci.2018.04.030. URL https://www.sciencedirect.com/science/article/pii/S0927025618302714.

Austin Tripp, Erik Daxberger, and José Miguel Hernández-Lobato. Sample-efficient optimization in the latent space of deep generative models via weighted retraining. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA, 2020. Curran Associates Inc. ISBN 9781713829546.

Aaron van den Oord, Oriol Vinyals, and koray kavukcuoglu. Neural discrete representation learning. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/7a98af17e63a0ac09ce2e96d03992fbc-Paper.pdf.

Michel van Kempen, Stephanie S. Kim, Charlotte Tumescheit, Milot Mirdita, Jeongjae Lee, Cameron L. M. Gilchrist, Johannes Söding, and Martin Steinegger. Fast and accurate protein structure search with Foldseek. *Nature Biotechnology*, 42(2):243–246, 2024. doi: 10.1038/s41587-023-01773-0.

Xinyou Wang, Zaixiang Zheng, Fei Ye, Dongyu Xue, Shujian Huang, and Quanquan Gu. Dplm-2: A multimodal diffusion protein language model. *ArXiv*, abs/2410.13782, 2024. URL https://api.semanticscholar.org/CorpusID:273403705.

Jiahui Yu, Xin Li, Jing Yu Koh, Han Zhang, Ruoming Pang, James Qin, Alexander Ku, Yuanzhong Xu, Jason Baldridge, and Yonghui Wu. Vector-quantized image modeling with improved VQGAN. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=pfNyExj7z2.

Lijun Yu, Jose Lezama, Nitesh Bharadwaj Gundavarapu, Luca Versari, Kihyuk Sohn, David Minnen, Yong Cheng, Agrim Gupta, Xiuye Gu, Alexander G Hauptmann, Boqing Gong, Ming-Hsuan Yang, Irfan Essa, David A Ross, and Lu Jiang. Language model beats diffusion - tokenizer is key to visual generation. In *The Twelfth International Conference on Learning Representations*, 2024a. URL https://openreview.net/forum?id=gzqrANCF4g.

Qihang Yu, Mark Weber, Xueqing Deng, Xiaohui Shen, Daniel Cremers, and Liang-Chieh Chen. An image is worth 32 tokens for reconstruction and generation. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024b. URL https://openreview.net/forum?id=tOXoQPRzPL.

Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The Unreasonable Effectiveness of Deep Features as a Perceptual Metric . In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 586–595, Los Alamitos, CA, USA, June 2018. IEEE Computer Society. doi: 10.1109/CVPR.2018.00068. URL https://doi.ieeecomputersociety.org/10.1109/CVPR.2018.00068.

Yue Zhao, Yuanjun Xiong, and Philipp Kraehenbuehl. Image and video tokenization with binary spherical quantization. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=yGnsH3gQ6U.

Li Zheng, Siddhant Kumar, and Dennis M. Kochmann. Algebraic language models for inverse design of metamaterials via diffusion transformers. *Nature Machine Intelligence*, 8(4):628–640, 2026. ISSN 2522-5839. doi: 10.1038/s42256-026-01218-8. URL https://doi.org/10.1038/s42256-026-01218-8.

Yongxin Zhu, Bocheng Li, Yifei Xin, Zhihua Xia, and Linli Xu. Addressing representation collapse in vector quantized models with one linear layer. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 22968–22977, 2025. doi: 10.1109/ICCV51701.2025.02132.

Shaobin Zhuang, Yiwei Guo, Fangyikang Wang, Canmiao Fu, Zhipeng Huang, Zeyue Tian, Xiaohui Li, Ying Zhang, Chen Li, and Yali Wang. Wetok: Powerful discrete tokenization for high-fidelity visual reconstruction. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=QteJJF57yG.

A NOTATION

Symbol	Meaning
$I, \hat{I}, \tilde{I}$	input field, decoded field, rendered tessellation
$N = 1024$	candidate cells, one per position of the 32×32 grid
K	transmitted seeds in an image, decided by the existence gate
$s_k = (x_k, w_k, a_k, b_k, c_k)$	seed k: position, weight, anisotropy pair, colour
C_k	Laguerre cell owned by seed k, Eq. (2)
$w_k^{\text{eff}}(p)$	direction-dependent weight of seed k at pixel p
τ	temperature of the soft power diagram, annealed to 10^{-3}
ρ	keep-fraction of the persistence gate, the inference rate knob
L	number of levels of one FFQ channel, $[24, 24, 8, 12]$
Φ_α	fold of Eq. (3), α annealed $0 \to 1$
g	learned monotone warp, piecewise linear on 16 bins
R	rate in bits per image, Eq. (1)
ϵ_{size}	relative error of mean grain area
W_1^{ODF}	1-Wasserstein distance between orientation distribution functions, degrees
$\mathrm{M_a}$	fraction of original grain area matched to a reconstructed grain

B PRINCIPLES AND WHERE EACH IS VERIFIED

We state the properties a tokenizer for a physical field should have, and map each to the evidence in this paper. A principle with no evidence column is a slogan.

	Principle	Verified in
P1	**Solver-ready fidelity.** Grain geometry and orientation, the inputs a crystal-plasticity solver reads, survive the bottleneck; pixel error is a proxy only.	Tab. 1, Fig. 8, Fig. 4
P2	**Graceful rate control.** One model spans a range of budgets, and as the rate falls fidelity degrades in a stated order, texture before the grain map.	Fig. 3, Tab. 12
P3	**Honest rate.** The reconstruction is a pure function of the transmitted code and every reported bit is decodable with zero side information.	§3.1, App. C
P4	**Exact-gradient discretization.** No codebook, no entropy penalty, no straight-through estimator.	Tab. 2, App. D
P5	**Uniformity across families.** Fidelity on the hardest family stays close to the mean rather than collapsing.	App. H
P6	**Modality generality.** The same encoder, seed generator, quantizer and rate model remain competitive on pixel fidelity when only the decode adaptor changes.	App. J

C Reproducibility details

Data. The corpus is built from 15 alloy directories of synthetic representative volume elements and split once, stratified by family, into 90,900 training and 10,100 test images by the split script released with the code. Images are stored as 300×300 RGB PNGs and resized to 256×256 with nearest-neighbour interpolation, which preserves the piecewise-constant structure; every method receives exactly this input. Grain targets used to supervise the raw seed head are computed on the fly by connected components on quantized colour, never from the test split.

Configuration.

Group	Setting	Value	Note
Ends	backbone	ViT-VQGAN, 512 dim, depth 8, patch 8	pretrained
	trainable blocks	last 4 encoder, last 6 decoder	rest frozen
Seeds	grid	32×32, one candidate per cell	$N = 1024$
	channels	9: $\Delta x, \Delta y, \log w, rgb, e, a, b$	
	offset bound	0.556 cell widths	keeps seeds in their cell
	$\log w$ range	$[-10.82, -3.91]$	fixed, decodable
FFQ	levels	$[24, 24, 8, 12]$	offset, weight, colour, shape
	fold iterations	5	residual < 0.01 level
	warp bins	16 per channel	identity at init
	α schedule	$0 \to 1$	annealed over training
Render	τ	10^{-3} train, 3×10^{-5} eval	hard at inference
Gate	training keep-fraction	$\rho \sim \mathcal{U}[0.4, 1]$	one model, whole curve
Rate	entropy model	two stage, 64 hidden	Eq. (1)
Optim.	optimizer	Adam (Kingma & Ba, 2014)	
	learning rates	9×10^{-5} bottleneck, 1.5×10^{-5} decoder, 3×10^{-7} encoder	
	EMA decay	0.999	
	hardware	4× H200, DDP	

Training chain of the reported checkpoint. The headline model is not a single run: a from-scratch stage establishes the seed representation, a refinement stage tightens reconstruction at a fixed alphabet, and a rate-budget stage anneals the active-seed target towards the operating point. We release every intermediate checkpoint and the driver scripts, and we report the ablation ladder of Tab. 4 along that same chain so that each row is an existing run rather than a reconstruction of one.

Evaluation. Reconstruction, segmentation and metric computation are three separate stages with cached intermediates, so that a metric change never re-runs a model and a model change never re-

runs the segmenter on the originals. Segmentation of the originals is computed once and shared by every method in every table.

D THEORY FOR FFQ

Fixed points and multipliers. Write $\Phi_\alpha(u) = u - \frac{\alpha}{2\pi}\sin(2\pi u)$. Then $\Phi_\alpha(k) = k$ for every $k \in \mathbb{Z}$, and $\Phi'_\alpha(u) = 1 - \alpha\cos(2\pi u)$, so $\Phi'_\alpha(k) = 1 - \alpha$ and $\Phi'_\alpha(k + \frac{1}{2}) = 1 + \alpha$. For $\alpha \in (0, 1)$ every integer is an attracting fixed point with multiplier $1 - \alpha$ and every half-integer is repelling with multiplier $1 + \alpha$. Because Φ_α is monotone for $\alpha < 1$, the interval $(k - \frac{1}{2}, k + \frac{1}{2})$ is invariant, so no value can cross into the basin of another level: the assignment of a scalar to its level is the nearest-level assignment, exactly as a hard rounding would give (Fig. 5).

Residual after k folds. Let $r_0 = u - \mathrm{round}(u)$ be the initial residual, $|r_0| \le \frac{1}{2}$. Near a fixed point the map contracts at rate $1 - \alpha$, and a second-order expansion gives $|r_k| \le (1 - \alpha)^k |r_0| + O(|r_0|^3)$ per step, so the worst case $|r_0| = \frac{1}{2}$ contracts by more than an order of magnitude in five steps at $\alpha \to 1$. This is why a single fold does not quantize and five do: one application moves a midpoint only slightly, while the composition drives it onto the level. The number of iterations is the one hyperparameter of the fold; Fig. 6 shows the potential the fold descends and the annealing schedule.

Limits. At $\alpha = 0$, $\Phi_0 = \mathrm{id}$ and FFQ is the identity, so training can begin with no quantization error and anneal into a hard quantizer without changing the objective. At $\alpha = 1$ with $g = \mathrm{id}$, the iterated fold converges pointwise to round away from the half-integers, so FSQ is exactly the limiting case of FFQ with a uniform level placement, and FFQ with a learned warp can only improve on it in the sense of placing levels closer to the data. Both statements together are Prop. 1.

Why the warp costs nothing. The warp g is a global parameter of the model, not a per-image quantity. The encoder transmits the integer index $\mathrm{Index}(x) = \mathrm{round}((L-1)g(x))$ and the decoder applies g^{-1}, which it already holds. Therefore learning the level placement changes the distortion at a given rate without changing the rate, which is the property a codebook does not have: a learned codebook must be stored, and a codebook that collapses wastes the alphabet it was given. Fig. 7 traces one scalar through the warp and shows how the inverse warp reallocates the reconstruction levels.

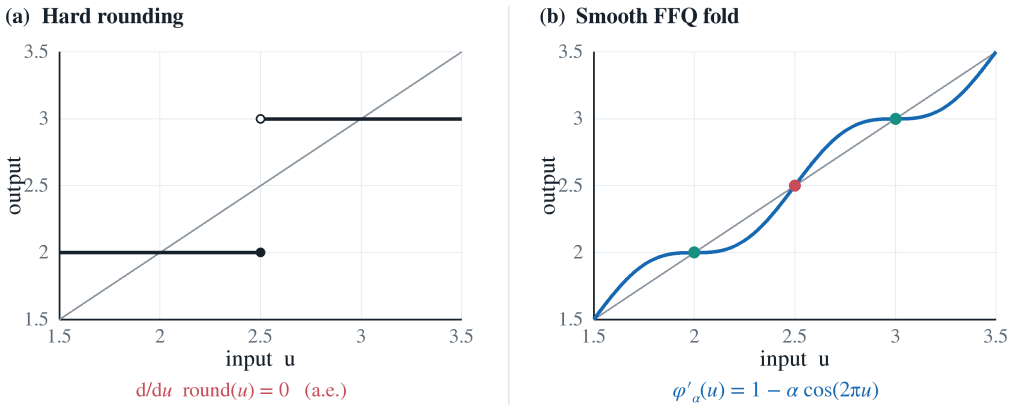

Figure 5: **Hard rounding blocks the gradient, the fold does not.** (a) Rounding is piecewise constant with zero derivative almost everywhere, so a straight-through estimator must supply a gradient that is not the derivative of the forward map; markers show one tie-breaking convention at the midpoint. (b) One fold at α=1 keeps the same fixed points while staying differentiable. Green marks attracting integers, red the repelling midpoint.

Algorithm.

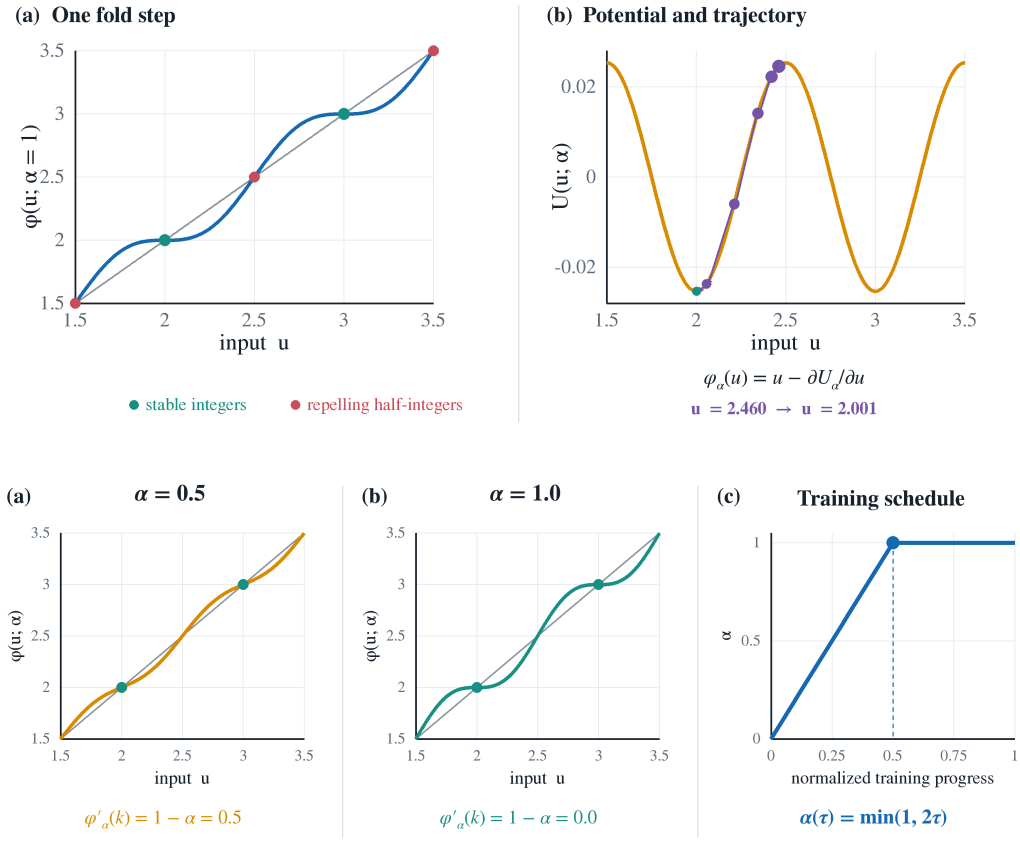

Figure 6: **The fold is an attraction map and α sets its strength.** Top: one fold step, and the periodic potential $U_\alpha(u) = -\alpha\cos(2\pi u)/4\pi^2$ whose negative gradient the fold follows; five iterations carry u=2.460 to 2.001. Bottom: the fold at α=0.5 and at α=1, where the derivative at an integer falls from 0.5 to 0, with the schedule that reaches α=1 halfway through training.

Algorithm 1 FFQ encode and decode for one channel with L levels

Encode

$u \leftarrow g(x)\,(L-1)$	*warp into level units*
if training: **for** $i = 1..5$: $u \leftarrow u - \frac{\alpha}{2\pi}\sin(2\pi u)$	*smooth, exact gradient*
$i \leftarrow \mathrm{round}(u)$	*transmitted symbol*
$\ell \leftarrow -\log_2 p(i \mid \text{context})$	*rate, Eq.* (1)

Decode

$x \leftarrow g^{-1}(i/(L-1))$	*g is held by the decoder*

E METRIC DEFINITIONS

From colour to orientation. Each pixel of the dataset stores the stereographic projection $S \in [-1,1]^3$ of a unit quaternion, shifted so that the class-mean orientation is the identity. Decoding inverts the projection, $q_w = (1-\|S\|^2)/(1+\|S\|^2)$ and $q_{xyz} = 2S/(1+\|S\|^2)$, undoes the mean shift, and folds the result into the fundamental zone of the hexagonal point group. Every orientation metric below is computed on the decoded quaternion field of the original and of the reconstruction, never on RGB values, which is what makes an error in degrees rather than in colour units.

Disorientation. For two orientations q_1, q_2 the disorientation is $\theta(q_1,q_2) = \min_{S\in\mathcal{S}} 2\arccos\left|\mathrm{Re}(S\,q_1^* q_2)\right|$, minimised over the 12 symmetry operations $\mathcal{S}$ of the hexagonal group, and lies in $[0°, 90°]$. It is the quantity a crystal-plasticity solver is sensitive to, because it determines the Schmid factors of the active slip and twin systems.

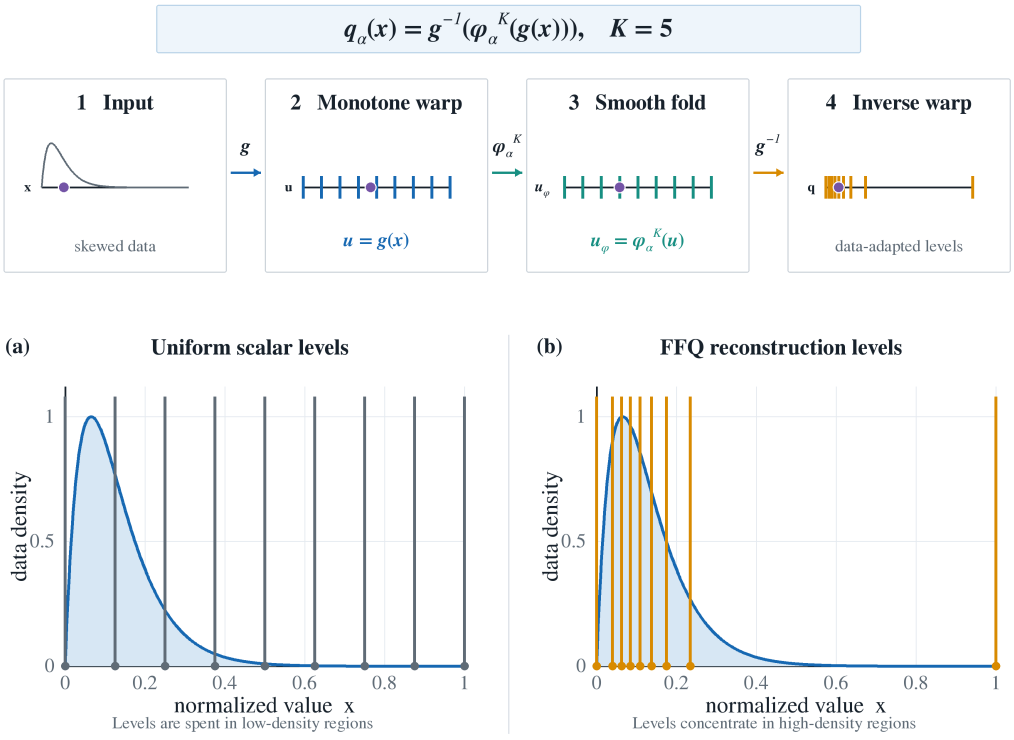

Figure 7: **FFQ quantizes in a learned coordinate system.** Top: one scalar from data space to symbol and back under $q_\alpha(x) = g^{-1}(\Phi_\alpha^{(K)}(g(x)))$ with $K{=}5$; the grid is uniform in warped space and non-uniform after g^{-1}. Bottom: with the same nine-symbol alphabet, uniform quantization spends levels in the tail while the inverse warp concentrates interior levels where the data is. The density is schematic.

Orientation distribution distance W_1^{ODF}. Both orientation sets are mapped to Rodrigues vectors after flipping to the positive hemisphere, binned on a shared grid whose extent is set by a percentile of the original, and compared by the exact 1-Wasserstein distance with disorientation between bin centres as the ground cost, solved as a balanced transportation linear program. The grid is shared, so identical distributions give exactly zero. The grain-level variant weights each grain by its area, so it measures the texture the solver integrates rather than the pixel histogram.

Grain metrics. Original and reconstruction are segmented with the same model and settings, and the segmentation of the originals is cached and reused for every method. From the label maps we compute grain count, the relative error of mean grain area ϵ_{size}, and the absolute error of mean aspect ratio. Grains are matched between original and reconstruction by pixel intersection over union above a threshold, and $\mathrm{M_a}$ is the fraction of original grain area that found a match. Together these are the contract a solver pre-processor would extract from a decoded map: which grains exist, how large they are, and which orientation each carries.

Why these and not a single number. A scalar built in the feature space of a pretrained network would inherit that network's training distribution, which is exactly the failure mode we are arguing against. The suite above is defined entirely in physical units, is never optimised during training, and is computed with the same implementation for every row of every table. Fig. 8 is the empirical side of this argument.

F Extended related work

Tab. 5 places FFQ against the quantizers of §2 on the axes that decide what each one needs to be trained.

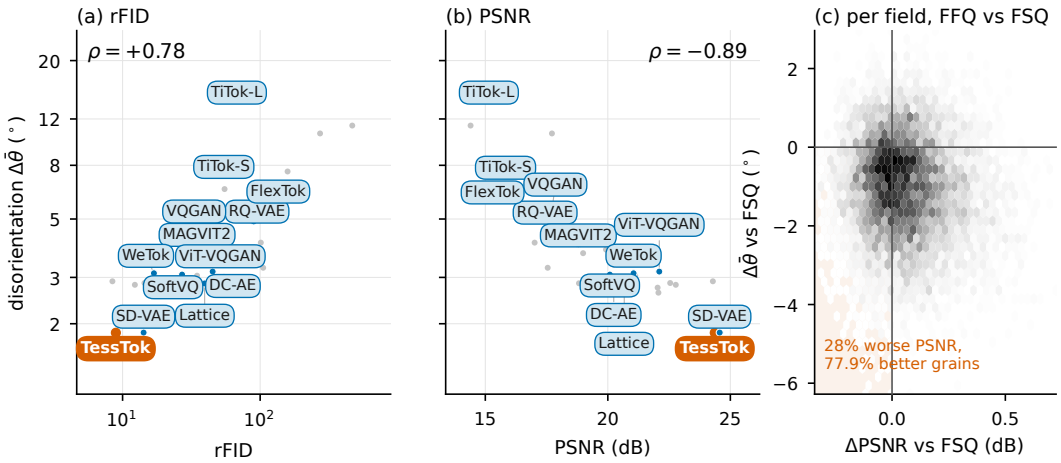

Figure 8: **Which metric predicts grain fidelity.** (a) rFID and (b) PSNR against $\Delta\bar{\theta}$ on a log disorientation axis over all 29 evaluated models, with Spearman ρ_s in the corner. Capsules name the released tokenizers and carry a leader to their true point where nudged apart, and grey dots are the quantizer and decoder variants that ρ_s also covers. (c) Per-field difference in PSNR and $\Delta\bar{\theta}$ between TessTok and FSQ-in-TessTok as a density over the 10,100 test fields, the shaded quadrant marking fields where the two criteria disagree.

Table 5: Quantizer positioning. Each prior method needs at least one of a learned codebook, an entropy penalty or a straight-through estimator; FFQ needs none and is the only one whose levels are learned from the data at zero side information. K is the codebook size, d the code dimension. FFQ's commitment term is a training-time potential, zero exactly on the levels, dropped at inference. STE: straight-through estimator.

Quantizer	Codebook	Entropy penalty	Codebook / commit loss	Gradient at bottleneck	Level placement	Lookup cost
VQ (NeurIPS'17)	learned	no	both	STE	learned	$O(Kd)$
RQ (CVPR'22)	learned	no	both	STE	learned	$O(DKd)$
SimVQ (ICCV'25)	learned	no	commit	STE	learned	$O(Kd)$
SoftVQ (CVPR'25)	learned	KL	none	soft	learned	$O(Kd)$
FSQ (ICLR'24)	none	no	none	STE	uniform	$O(1)$
LFQ (ICLR'24)	none	yes	commit	STE	binary	$O(1)$
BSQ (ICLR'25)	none	yes	commit	STE	binary, spherical	$O(1)$
GQ	none	yes	commit	STE	binary, grouped	$O(1)$
Lattice	none	no	none	STE	lattice	$O(\text{decode})$
FFQ (ours)	**none**	**no**	**commit only**	**exact**	**learned**	$O(1)$

Repairing the codebook. Beyond the main text, several lines of work target codebook collapse directly: reducing the code dimension while enlarging the vocabulary, re-initialising dead entries during training, and reparameterising the codebook so that gradient updates reach unused entries. These methods improve utilization but keep the lookup, the auxiliary losses and the estimator. TessTok removes the object they repair.

Variable-length and content-adaptive codes. Adaptive-length tokenizers choose how many tokens to spend per image, usually through a learned halting rule or a nested ordering of a fixed sequence. The length is then a model decision evaluated at a chosen operating point. Classical codecs instead let an entropy coder determine the length from the symbols, which is the convention we follow: the rate is measured, not selected, and the same checkpoint spans a range of budgets.

Microstructure representation in materials science. Parametric descriptions of polycrystals are standard: Voronoi and Laguerre tessellations, spherical-harmonic descriptors of texture, and n-point statistics for two-phase media. They are normally fitted by optimisation to a measured structure, one structure at a time, and are not learned, not amortised and not paired with a decoder. Generative work

on microstructures mostly targets plausible samples evaluated by image metrics. Our contribution is to put the classical parameterisation inside a learned, rate-priced tokenizer and to evaluate it by what a solver reads.

G Ablation notes and negative results

Removing the warp. With the learned warp removed and $\alpha \to 1$, FFQ becomes FSQ (Prop. 1), so the FSQ row of Tab. 2 is this ablation, trained for 60 epochs rather than the full schedule.

Negative results. Two mechanisms did not earn their keep and we report them rather than quietly remove them. An *area-adaptive colour alphabet*, giving large grains finer colour levels than small ones at no signalling cost, collapsed to a single tier in the final configuration: at this budget the extra levels did not pay for themselves. A *boundary loss* on the soft tessellation, intended to sharpen interfaces, was neutral once the temperature anneal was in place and was set to zero weight.

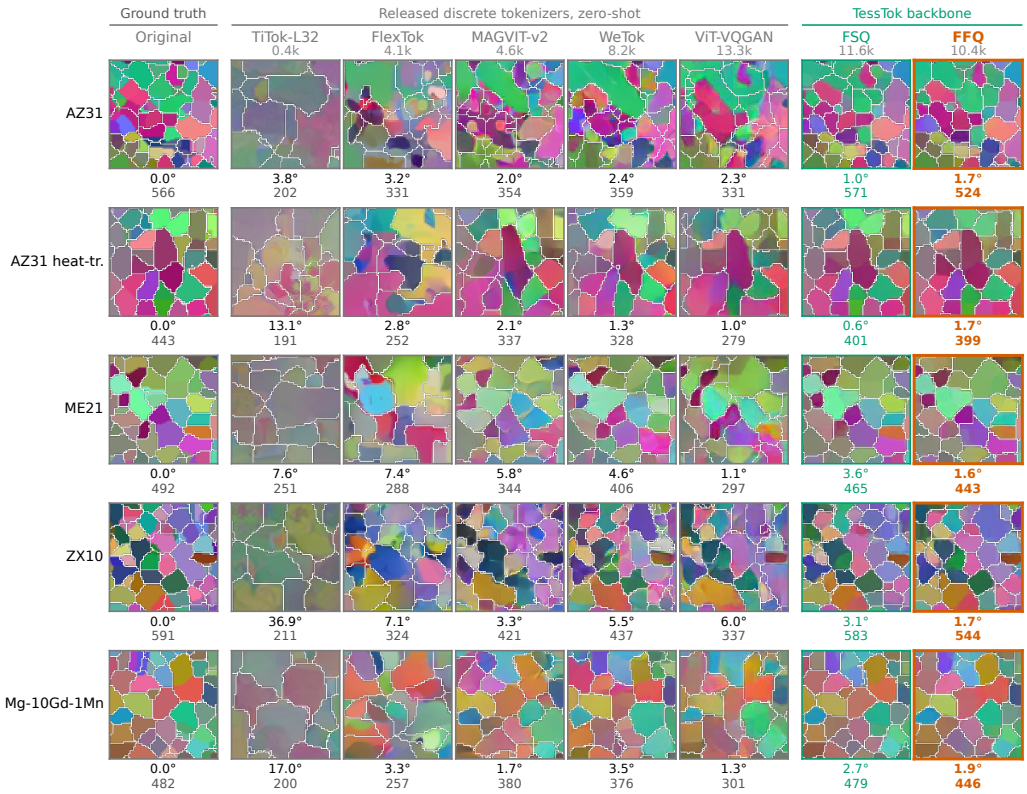

Figure 9: **Five alloy families, same columns as Fig. 4.** Rows span the grain-size and anisotropy range; each is its family's median field by TessTok disorientation, with its own ground-truth-chosen window. Disorientation on a single field is noisy, so FSQ leads two rows while FFQ leads the full test set (Tab. 2).

H Extended quantitative results

Tab. 8, Tab. 9 and Tab. 10 give every measured metric for the three axes; the main text reports a subset chosen for readability. Two columns deserve a comment. The disorientation of the volume-mean orientations, listed as $\Delta\bar{\theta}$, is a weak discriminator, because a method can preserve the mean texture while misplacing every grain, which is why we always read it together with matched area. The grain-level orientation EMD, $W_1^{\mathrm{ODF}}{}_g$, is the area-weighted counterpart of the pixel statistic and tracks it closely, which is a consistency check on the segmentation rather than an independent result.

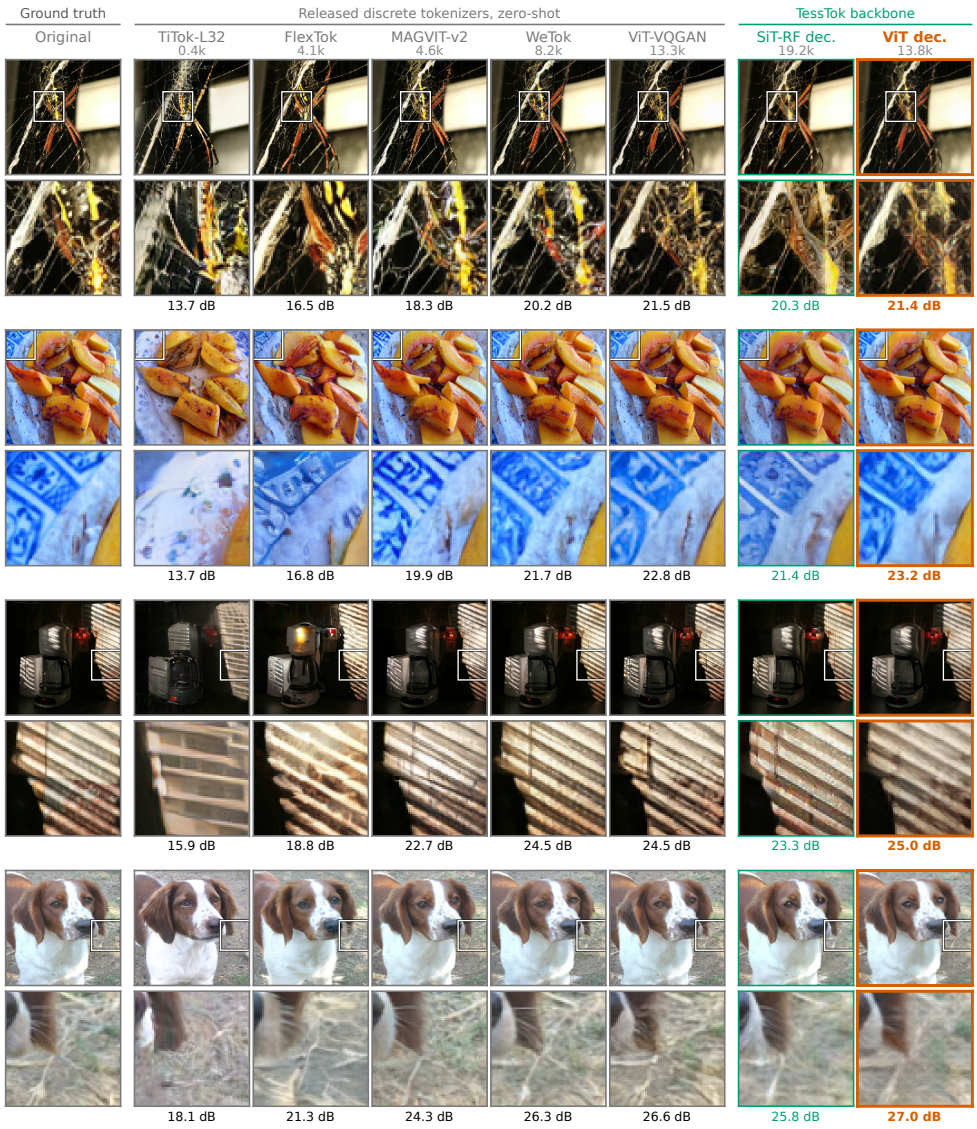

Figure 10: **ImageNet reconstructions.** Four held-out images at the 20th, 40th, 60th and 80th percentiles of TessTok PSNR, among those at least 256 px on each side, each shown whole (upper) and as a 64×64 zoom chosen from the original alone (lower), with per-image PSNR below. Columns are the checkpoints of Fig. 4, with native bits under each header. The last two columns are TessTok with the SiT rectified-flow decoder and with the deterministic ViT decoder, each at its own rate.

Fine-tuning the baselines. Fine-tuning raises matched area for every baseline and lowers $\Delta\bar{\theta}$ for all but the lattice tokenizer, but it costs WeTok and the lattice tokenizer PSNR and raises rFID for seven of the twelve, because our fine-tuning objective carries no adversarial term (Tab. 7).

Per-family behaviour. Fidelity is nearly uniform across the 15 families (Tab. 6): matched area spans 0.912 to 0.941 and $\Delta\bar{\theta}$ spans 1.69° to 2.01°. What varies is grain count. PSNR falls from 25.60 dB on Z1, the coarsest family at 375 grains per field, to 23.54 dB on ME21, the finest at 494 (Spearman $\rho_s = -0.86$ over families), and ME21 also has the lowest matched area, consistent with small grains sharing a seed. The hardest family stays within 0.015 of the mean matched area.

Table 6: Per-family results of the reported TessTok model on the 10,100 test fields, sorted by mean grain count. n: test fields per family; Grains: mean segmented grain count of the originals. Angles in degrees.

Family	n	Grains	PSNR↑	W_1^{ODF}↓	$\Delta\bar{\theta}$↓	$\mathrm{M_a}$↑
Z1	698	375	25.60	5.24	2.01	0.941
ZX10	750	430	24.48	4.82	1.86	0.932
AZ31, heat-treated	750	431	24.70	5.05	1.83	0.924
Mg-5Gd-0.5Mn	486	437	24.37	4.88	1.87	0.927
Mg-5Gd	668	438	24.37	4.89	1.82	0.931
Mg-5Gd-1Mn	520	439	24.42	4.84	1.85	0.928
AZ31	801	439	24.52	5.10	1.82	0.922
Mg-10Gd-0.5Mn	658	439	24.13	4.90	1.87	0.929
Mg-2Gd	699	440	24.37	4.87	1.84	0.929
ZNd10	745	443	24.35	4.80	1.79	0.928
Mg-2Gd-0.5Mn	601	444	24.17	4.84	1.78	0.927
Mg-10Gd	750	444	24.17	4.85	1.91	0.931
Mg-2Gd-1Mn	596	455	24.25	4.75	1.81	0.924
Mg-10Gd-1Mn	628	457	23.96	4.80	1.98	0.927
ME21	750	494	23.54	4.69	1.69	0.912

Table 7: Tokenizer axis under both protocols: the released checkpoint applied without adaptation (zs), and the same checkpoint fine-tuned on our training images with the last half of its blocks trainable (ft). Native bits. Tab. 1 reports the zs columns.

		PSNR↑		rFID↓		$\Delta\bar{\theta}$↓		$\mathrm{M_a}$↑	
Method	Bits	ft	zs	ft	zs	ft	zs	ft	zs
TiTok-L32	384	15.62	15.15	189.0	67.7	10.96	15.11	0.570	0.301
TiTok-S128	1,536	17.02	15.83	158.2	54.4	5.66	7.65	0.660	0.433
RQ-VAE	3,584	18.36	17.44	72.4	89.7	3.53	4.90	0.720	0.477
VQGAN	3,584	19.12	17.71	128.6	35.1	3.86	5.34	0.666	0.562
FlexTok	4,087	15.59	15.45	10.9	135.6	3.73	6.45	0.632	0.399
MAGVIT-v2	4,608	19.76	18.80	65.8	34.8	3.44	4.01	0.772	0.696
WeTok	8,192	20.66	21.05	33.2	16.9	2.91	3.11	0.844	0.807
Lattice-ViT	9,162	19.62	20.67	66.3	39.4	3.36	2.85	0.790	0.779
ViT-VQGAN	13,312	23.61	22.10	13.6	45.1	2.33	3.16	0.912	0.693
SoftVQ-VAE	65,536	22.24	20.08	33.5	27.1	2.14	3.08	0.805	0.694
DC-AE	65,536	23.26	20.24	19.5	54.6	2.10	3.00	0.842	0.672
SD-VAE	131,072	27.47	24.57	6.0	14.2	1.26	1.85	0.955	0.814

I RATE ACCOUNTING AND THE KEEP-FRACTION SWEEP

Where the bits go. Tab. 11 decomposes the rate of the two operating points reported in the main text. Three observations follow. The occupancy map is a small and almost constant part of the budget, near 1 kbit, because $\log_2 \binom{N}{K}$ is flat around $K \approx N/2$. The entropy model recovers between 9% and 20% of the payload relative to a uniform code, and it recovers more at the lower rate, where the surviving seeds are more predictable from their cell position. Anisotropy is transmitted by roughly half the seeds, following the median-weight rule of §3.4, so it costs about 2 kbit rather than the 4 kbit it would cost if every seed sent it.

One model, a range of budgets. Because the persistence gate is trained over a range of keep-fractions, the same checkpoint can be evaluated at any ρ without retraining; Tab. 12 reports the sweep plotted in Fig. 3 over $\rho \in \{1.0, 0.9, \ldots, 0.4\}$. ρ is the requested fraction of active seeds. The gate is a soft sigmoid around the persistence quantile, so the realised fraction sits below it, 0.78 at $\rho = 0.9$ and 0.38 at $\rho = 0.4$, which is why the first step saves the most bits. Across the sweep, bits follow the transmitted seed count almost exactly ($r = 0.998$ over the seven points). Within one ρ they do not: at $\rho = 1$ the existence gate keeps 444 seeds with a standard deviation of 12 whatever the field holds (seed count against grain count, $r = 0.03$), so the per-image variation in bits comes

Table 8: Tokenizer axis under the *fine-tuned* protocol, all measured metrics: every baseline is trained further on our training images with the last half of its blocks trainable under the same objective. Discrete block first, sorted by rate, then continuous baselines. TessTok rows give entropy-coded and uniform bits. Compare with Tab. 1, which uses the released checkpoints.

Method	Bits	PSNR↑	SSIM↑	MS-SSIM↑	LPIPS$_a$↓	LPIPS$_v$↓	rFID↓	ϵ_{size}↓	ϵ_{AR}↓	W_1^{ODF}↓	$\Delta\bar{\theta}$↓	$W_1^{\text{ODF}}{}_g$↓
TiTok-L32	384	15.62	0.391	0.249	0.658	0.604	189.0	0.418	0.033	16.99	10.96	21.12
TiTok-S128	1536	17.02	0.445	0.444	0.469	0.468	158.2	0.304	0.023	9.04	5.66	12.09
RQ-VAE	3584	18.36	0.503	0.577	0.341	0.392	72.4	0.213	0.019	7.44	3.53	11.21
VQGAN	3584	19.12	0.550	0.620	0.298	0.322	128.6	0.352	0.019	7.80	3.86	11.01
FlexTok	4087	15.59	0.399	0.416	0.403	0.484	10.9	0.180	0.019	7.15	3.73	10.28
MAGVIT-v2	4608	19.76	0.588	0.699	0.254	0.296	65.8	0.190	0.023	7.26	3.44	10.43
WeTok	8192	21.05	0.629	0.787	0.192	0.264	16.9	0.273	0.044	6.97	3.11	10.90
Lattice-ViT	9162	19.62	0.594	0.697	0.249	0.287	66.3	0.274	0.016	7.15	3.36	10.21
ViT-VQGAN	13312	23.61	0.749	0.881	0.126	0.157	13.6	0.103	0.023	5.45	2.33	9.78
TessTok	10426	24.36	0.766	0.908	0.105	0.150	8.9	0.086	0.014	4.89	1.85	9.19
TessTok	14286	24.75	0.786	0.918	0.096	0.135	5.7	0.064	0.016	4.87	1.80	8.82
SoftVQ-VAE	65536	22.24	0.652	0.809	0.168	0.233	33.5	0.257	0.029	5.34	2.14	10.31
DC-AE	65536	23.26	0.718	0.835	0.143	0.186	19.5	0.332	0.024	5.30	2.10	8.63
SD-VAE	131072	27.47	0.862	0.950	0.060	0.085	6.0	0.061	0.009	3.53	1.26	6.02

Table 9: Quantizer axis, all measured metrics, tokenizer and decoder fixed. Rows above the rule share the 60-epoch schedule; FFQ, below it, is the full schedule and is not budget matched.

Quantizer	PSNR↑	SSIM↑	MS-SSIM↑	LPIPS$_a$↓	LPIPS$_v$↓	rFID↓	ϵ_{size}↓	W_1^{ODF}↓	$W_1^{\text{ODF}}{}_g$↓	M_a↑
SoftVQ	17.01	0.439	0.633	0.304	0.411	101.1	0.091	7.88	17.89	0.849
LFQ	19.92	0.586	0.699	0.239	0.283	58.3	0.065	7.36	11.22	0.925
RVQ	20.23	0.600	0.714	0.240	0.271	55.4	0.027	7.22	11.72	0.913
BSQ	22.03	0.678	0.820	0.171	0.220	31.7	0.030	6.46	10.38	0.928
Lattice	22.06	0.684	0.825	0.169	0.211	27.2	0.031	6.21	10.01	0.928
SimVQ	22.54	0.713	0.839	0.158	0.193	14.4	0.043	6.30	9.58	0.938
VQ	22.77	0.719	0.847	0.151	0.187	12.3	0.037	6.19	9.55	0.941
FSQ	24.29	0.770	0.892	0.111	0.150	8.4	0.046	6.04	9.45	0.945
FFQ (ours), full schedule	24.36	0.766	0.908	0.105	0.150	8.9	0.086	4.89	9.19	0.927

from the entropy-coded payload and follows the grain count at $r = 0.60$ per image and 0.85 over family means (Fig. 3a). The current gate does not allocate seeds by content. As ρ falls, orientation and texture degrade before the grain map: over the full sweep W_1^{ODF} rises by 7%, $\Delta\bar{\theta}$ by 21% and rFID by a factor of 2.3, while M_a falls by 2.4% and stays above 0.9 throughout (Fig. 3b). M_a is area-weighted, and the seeds the gate drops first carry little area: at $\rho = 0.4$, 167 seeds for about 440 grains still match 90% of the area. Removing a seed hands its cell to its neighbours in the power diagram, so the render cannot add a grain.

J NATURAL IMAGES

Why this experiment exists. A reviewer is entitled to ask whether the gains come from the tokenizer or from the domain. The experiment answers it structurally rather than rhetorically: the encoder, the seed generator, FFQ, the existence gate and the entropy model are the same modules as in the microstructure model, retrained on photographs, and only the decoder changes, from the tessellation path to a render-free one (§3.4). There is no power diagram, no grain oracle and no segmentation anywhere in this setting.

Setup. Training uses $90{,}810$ ImageNet-1k images over all $1{,}000$ classes at 256×256 with the oracle seed loss disabled, since photographs have no grains, and a self-supervised seed teacher in its place: the raw seed field is optimised for a few steps against a low-resolution differentiable render of the target and the result is distilled back into the seed head. Evaluation uses the $10{,}090$ held-out images and reports PSNR, MS-SSIM, LPIPS and rFID over all pairs with the metric code of the main text, and no segmentation-based metric, because the grain metrics are undefined here. Both pools are drawn from the ImageNet-1k *training* split, on which every released checkpoint was trained, so the released rows may have seen the test images and the comparison favours them. We report

Table 10: Decoder axis, all measured metrics. A fresh tokenizer is co-trained with each decoder, so no row benefits from a frozen tokenizer; the most important of the 16 decoder families we trained are listed and the others are left out. The deterministic ViT decoder is the one used everywhere else in the paper.

Decoder	Family	PSNR↑	SSIM↑	MS-SSIM↑	LPIPS$_v$↓	rFID↓	ϵ_{size}↓	W_1^{ODF}↓	$W_1^{\text{ODF}}{}_g$↓	M_a↑
ViT (ours)	deterministic	24.75	0.786	0.918	0.135	5.7	0.064	4.87	8.82	0.942
SDXL-VAE	VAE	21.12	0.657	0.803	0.250	34.8	0.072	7.16	9.75	0.884
BigGAN	GAN	18.99	0.543	0.743	0.371	19.7	0.243	6.50	14.53	0.812
SD3.5 MMDiT	diffusion	18.82	0.503	0.656	0.388	15.3	0.181	6.47	11.63	0.737
SiT-XL	flow	17.72	0.535	0.498	0.588	272.7	11.52	15.73	22.98	0.279
MAR-L	autoregr.	17.55	0.447	0.553	0.480	105.2	0.392	6.75	14.90	0.625
DiT-XL	diffusion	17.06	0.454	0.444	0.537	55.1	0.307	9.10	13.97	0.576
MaskGIT	masked LM	15.46	0.350	0.350	0.656	158.3	1.660	9.65	21.54	0.286
StyleGAN disc.	GAN	14.40	0.247	0.099	0.808	467.4	217.2	17.62	45.40	0.059

Table 11: Rate decomposition, averaged over the 10,100 test images, with hard-rounded codes. Uniform prices every symbol at $\log_2 L$ bits; entropy prices it with the two-stage model that the decoder also holds.

	Reported operating point	Higher-rate variant
Active seeds K	444.1	545.7
Seeds transmitting anisotropy	239.4 (53.9%)	308.5 (56.5%)
Occupancy $\log_2 \binom{1024}{K}$	1,005	983
Payload, uniform	11,823	14,629
Payload, entropy model	9,421	13,303
Total, uniform	12,828	15,612
Total, entropy model	**10,426**	**14,286**
Compression ratio	151×	110×

two decoders on the same seed code: the deterministic ViT decoder of the microstructure model, initialised from the ViT-VQGAN ends, and a SiT rectified-flow decoder of 890M parameters. Unlike the keep-fraction sweep of App. I, each TessTok row of Tab. 3 is a separate training, and both ViT rows transmit all 1,024 cells. The target is parity at comparable rate, not state of the art.

Results. At the high rate the deterministic decoder is ahead on every column: at 13.8 kbit it beats the rectified-flow decoder at 19.2 kbit. At the low rate it leads on PSNR, 22.76 against 20.39 dB at 8.5 against 9.5 kbit, and on MS-SSIM and LPIPS, while the rectified-flow decoder is ahead on rFID, 10.15 against 12.80. Against ViT-VQGAN, which shares the encoder and decoder ends and spends 13.3 kbit, the 13.8 kbit point is ahead on PSNR, 24.41 against 24.27 dB, and on rFID, 4.01 against 4.93, and behind on MS-SSIM and LPIPS. It gives the highest PSNR of any row in Tab. 3, but WeTok at 8.2 kbit is better on LPIPS and rFID than every TessTok point, and MAGVIT-v2 at 4.6 kbit on rFID. The seed code is therefore a competitive pixel codec on photographs and not a perceptual one, which is what the objectives of §3.4 ask of it. Fig. 10 shows where the gap sits: both TessTok decoders keep edges and layout, and the ViT decoder softens fine texture such as grass and fur.

K Qualitative results and failure cases

Fig. 9 repeats Fig. 4 across five families. Three failure modes are visible in it and all three are geometric. *Concave grains.* A recrystallised necklace is one grain in the label map and at least two convex cells in a Laguerre tessellation, so its boundary is straightened. *Sub-cell grains.* Grains smaller than roughly two grid cells cannot receive their own seed and merge with a neighbour; the merged region takes the neighbour's orientation, which preserves the texture statistic while costing matched area. *Boundary triple junctions.* Where three cells meet, a small error in one weight moves the junction along the boundary, which is visible in the error map as a thin line rather than a region. None of these produces invented structure, which is what distinguishes a deterministic tessellation decoder from a generative one.

Table 12: Keep-fraction sweep of one TessTok checkpoint on the 10,100 test fields. ρ is the requested keep-fraction, K the mean number of transmitted seeds and Bits the mean entropy-coded occupancy plus payload. The $\rho = 1.0$ row is the TessTok row of Tab. 1.

ρ	K	Bits	PSNR↑	MS-SSIM↑	LPIPS↓	rFID↓	W_1^{ODF}↓	$\Delta\bar{\theta}$↓	$\mathrm{M_a}$↑
1.0	444.1	10,426	24.36	0.908	0.105	8.93	4.89	1.85	0.927
0.9	344.8	8,513	24.06	0.902	0.112	10.12	4.99	1.97	0.921
0.8	311.7	7,796	23.99	0.900	0.113	11.40	5.04	2.01	0.920
0.7	281.1	7,100	23.89	0.897	0.116	13.07	5.12	2.08	0.919
0.6	247.7	6,303	23.72	0.894	0.119	15.16	5.16	2.14	0.915
0.5	208.9	5,370	23.55	0.890	0.123	17.63	5.17	2.17	0.911
0.4	167.1	4,365	23.38	0.885	0.127	20.50	5.25	2.24	0.905

L Complexity and cost

Asymptotics. FFQ is $O(1)$ per scalar with no memory beyond 16 parameters per channel. The entropy model is $O(K)$ in the number of transmitted seeds. The power-diagram render is $O(P \cdot K)$ in pixels and active seeds and is evaluated in chunks so that the $P \times K$ matrix is never materialised; it is the only term specific to the microstructure branch and it disappears in the natural-image branch. The geometry-aware adaptor attends to the 16 nearest seeds per query, so it is $O(L \cdot 16)$ rather than $O(L \cdot K)$.

Measured cost. The reported model has 71.8M parameters, of which 51.9M are trained: 19.7M in the seed generator, adaptor and entropy model, and 12.6M and 19.6M in the last four encoder and last six decoder blocks of the pretrained ViT-VQGAN ends. One epoch over the training set takes about 19 minutes on four H200 GPUs, so the training chain of App. C costs roughly 90 GPU-hours, estimated from checkpoint times.

M Extended limitations and negative results

Limitations, restated with their evidence. The convex-cell limitation is shown in App. K. The sub-cell-grain limitation shows per family in App. H, where the finest family has the lowest matched area, and is bounded by the keep-fraction sweep of App. I: below $\rho = 0.9$ the matched area lost per step grows as ρ falls, yet at $\rho = 0.4$ it is still 0.905 (Tab. 12). The constant-orientation-per-cell limitation is structural: intragranular gradients are not representable at all, which is correct for the undeformed grain maps a solver takes as input and wrong for deformed-state measurements. The 8-bit encoding of the dataset sets a floor under every metric we report, so none of our numbers should be read as approaching zero error.

What we did not do. We do not run a crystal-plasticity simulation end to end on reconstructed maps. We show that the quantities such a solver reads are preserved, and we treat a property proxy computed from the decoded orientation field as an external check on the metric suite, but a full simulation-to-simulation comparison is future work and we do not claim it. We also evaluate one material system with hexagonal symmetry in two dimensions; three-dimensional power diagrams are a direct extension of Eq. (2) and are untested here.

Negative results, collected. The area-adaptive colour alphabet collapsed to a single tier and was removed. The boundary loss was neutral once the temperature anneal was in place. Generative decoders, which are the direction natural-image tokenizers are moving, were worse than a deterministic decoder on every physical metric at every budget we tried, and on ImageNet a SiT rectified-flow decoder was worse than the deterministic one on every metric except the rFID of the lowest-rate pair. No TessTok point on ImageNet matches WeTok's LPIPS or rFID (Tab. 3).