`essentialism`: An On-Chain Generative Work That Forgets Two Photographs

Benjamin Meadows
Independent Artist and Researcher
ben@benjaminmeadows.com

November 2026

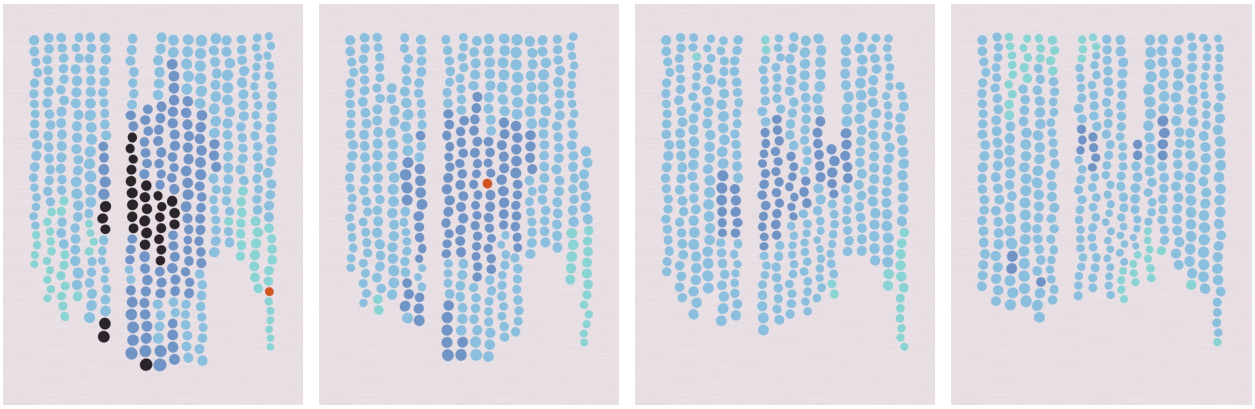
Abstract

`essentialism` is a generative artwork of 44 editions whose program is stored on the Tezos blockchain. The program carries two of my photographs as 28×38 grids of 16 gray levels. Each token draws one of them, blended with noise, at a weight set by its edition number: 100% at edition 1, falling by equal steps to 12% at edition 44. The token prints that weight as its "memory". I measured how much of the photograph survives into the drawing, over 1,000 seeds at every edition. The photograph's median correlation with the generator's internal darkness field stays above 0.94 through edition 21, and more than half of its total decline falls between editions 34 and 44; a closed form with two measured spreads reproduces that curve to within 0.021. In the ink of the drawn dots, the median rank correlation with the photograph falls from 0.82 at edition 1 to 0.14 at edition 44 for the tree (the sky runs 0.84 to 0.13), against a control near zero. The ledger, meanwhile, kept every write: every superseded version of the program, including a first version whose header claimed an algorithm that was not mine, and the record each mint wrote, whose render matches the platform's own image of every minted token at a gray-level correlation of 0.98 to 0.99. This is a study of one work by its author, and its measurements describe what the program draws at every edition, thirty of which are not yet minted, not what a viewer recognizes. Code, data and this paper's source rebuild from one command; they are archived at `https://doi.org/10.5281/zenodo.23333265` and maintained at `https://github.com/TheBenMeadows/essentialism-preprint`.

1 The work

`essentialism` [Meadows, 2026b] was published on 31 July 2026 as generator 407 on bootloader [objkt.com, 2026], a platform on Tezos [Goodman, 2014] that stores a generative program in a contract's storage and renders each token in the browser from that program and a per-token seed. The edition is 44 tokens at 8 tez; fourteen are minted as of 3 November 2026. Each token is a 750×1000 SVG: columns of dots dropped down a sheet of paper, colored from a small library of inks.

Each token draws one of two photographs, chosen by its seed: "Blue Tree" [Meadows, 2023a] 87% of the time and "Natural Gradient" [Meadows, 2023b] 13%. The tree was removed after a storm in December 2023, three months after I minted the photograph on Ethereum, so the picture has outlived its subject. The sky is an unedited photograph whose full-resolution file sits on Arweave [Williams et al., 2019]. The bootloader runtime allows no external files, so each photograph was reduced to a 28×38 grid of 16 gray levels and written into the code as a string. A token's *edition*



ed. 1, memory 100% ed. 15, memory 71% ed. 30, memory 41% ed. 44, memory 12%

Figure 1: One seed at four edition numbers: edition 1 (token 5364), with the seed `0x975fa1a8...4e98` from its artifact as the mint wrote it at block 14305273 under generator version 1, rendered here with version 6, then the same seed at editions 15, 30 and 44. Traits: field density, three inks, source "blue tree", one ember; density is how many columns are drawn, and the ember is the single orange dot, drawn at editions 1 and 15. The dark mass at edition 1 is the trunk and crown of the tree. Token 5364's mint-time artifact, rendered in a headless browser, matches the platform's own image of the token at a gray-level correlation of 0.98 (Section 5); the version-6 render shown here was not itself compared. Of the traits, only the memory factor and the ember change across the row; the composition shifts because dot sizes and run lengths follow the field defined in Section 2. Rendered by `repro/figures.mjs`, rasterized with Inkscape 1.4.4.

number e is its position in the mint order, 1 to 44. The program blends the photograph with noise at a weight $f(e)$ that is 1 at edition 1 and falls by equal steps to 0.12 at edition 44, and prints $f(e)$ as a percentage in the token's trait line, labelled "memory". I call f the memory factor and the percentage the printed memory. The dots take their ink from what is left of the photograph at that weight. Figure 1 shows one seed at four editions.

The column structure is not mine. In November 2022 I worked through Andy Haskell's tutorial on converting images to mosaics in p5.js [Haskell, 2022], randomized the dot size, pointed it at my own photographs, and saved nine test outputs under the name Essentialism without releasing them. In the tutorial's version each dot asks a photograph for its color; here each dot asks a grid the edition number weighs down by equal steps. Before writing this paper I had not checked how much of a photograph the printed memory leaves in the drawing. Sections 2 and 3 say how the program works and how I measured it, Section 4 what the editions remember, Section 5 what the ledger remembers, Section 6 where that sits among other ways of keeping and losing, and Section 7 what can be rebuilt and what the measurement cannot say.

2 Construction

The generator is 10,098 bytes of JavaScript (generator version 6, SHA-256 `33d85a12...c248`; the full digest is in Section 7). It is stored URI-encoded, 15,560 bytes, and runs in bootloader's SVG-JS runtime, which provides an SVG root, a seeded pseudo-random generator `rnd()` built from splitmix64 [Steele et al., 2014] feeding sfc32 [Doty-Humphrey], and the token's `iterationNumber`, which is its edition number. Line numbers below refer to the stored code.

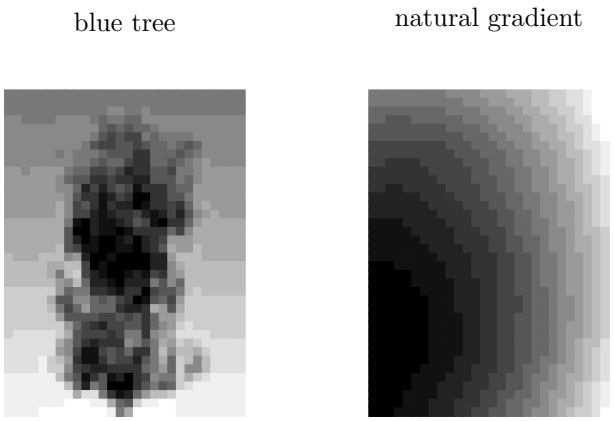


Figure 2: The two photographs as the ledger holds them: 28×38 cells, 16 gray levels, read from the generator's string literals by `repro/field.mjs`. This is the most of either photograph any edition can recall.

The grids. Each photograph was converted to grayscale, center-cropped to the grid's 28×38 aspect (close to the sheet's 3×4) and resampled to 28×38 cells, autocontrasted with a 1% cutoff, and quantized to 16 levels, written one hexadecimal character per cell: 1,064 bytes per photograph, carrying at most 532 bytes of information. Both strings are literals in the code (lines 25 and 26). Storing the two cost 0.532 tez: 2,128 bytes at the 250 mutez (0.00025 tez) Tezos charges per byte of contract storage. The grids correlate with each other at $r = 0.10$ over cells. Figure 2 shows both.

The darkness field. Let $P(x, y) \in [0, 1]$ be the bilinear interpolation of the token's grid at canvas position (x, y) on a canvas of width W and height H, where 1 is the lightest level, and let $n(x, y)$ be two octaves of value noise [Perlin, 1985] over a 16×16 lattice drawn from the first 256 outputs of `rnd()` (lines 35 to 42). The generator draws from a darkness field

$$D(x, y) \;=\; f\,(1 - P(x, y)) \;+\; 0.85\,(1 - f)\,n\big(3.1\tfrac{x}{W} + 5.2,\; 3.1\tfrac{y}{H}\big)\,, \tag{1}$$

with memory factor

$$f(e) \;=\; \max\!\left(0.12,\; 1 - 0.88\,\frac{\min(e, 44) - 1}{43}\right) \tag{2}$$

for edition number e (line 69). Over $e = 1, \ldots, 44$ the floor never binds: f falls by $0.88/43 \approx 0.0205$ per edition from 1 to exactly 0.12. Platform previews use $f = 1$.

Columns and dots. The canvas is divided into columns of width $3r_0$, where r_0 is a base dot radius drawn per token; the factor 3 and the column-of-dots loop are Haskell's [Haskell, 2022]. Whether a column is drawn, and how far down it runs, depends on a noise value per column and on D at the column's center. Down each column the generator places dots of radius $r = r_0\,(0.72 + 0.56\,n')(0.8 + 0.5\,D)$, where n' is a second noise value drawn along the column, so dark regions of the darkness field get larger dots and longer runs. Each dot's ink is chosen from D at its center: the token's k inks are sorted from light to dark and indexed by $\lfloor kD \rfloor$; below $D = 0.1$ the dot is usually bone, the paper's own off-white, and above 0.88 usually carbon, the darkest ink. With probability 0.58 a dot repeats the previous dot's ink when D has changed by less than 0.3 since that dot (line 129), which turns per-dot color noise into runs that break where the picture changes. On night paper, a dark sheet drawn for 8% of tokens, the k inks are indexed in reverse,

trait	value	weight in code	observed (20,000 seeds)
density	air / field / full	50 / 38 / 12%	49.7 / 38.1 / 12.3%
inks	1 / 2 / 3 / 4	4 / 56 / 30 / 10%	3.8 / 56.2 / 29.9 / 10.1%
source	blue tree / natural gradient	87 / 13%	86.8 / 13.2%
paper	night (carbon)	8%	8.1%
ember	present	55%	54.8%
memory	edition-determined	Eq. (2)	n/a

Table 1: Seeded traits. Density is how many columns are drawn (air the fewest, full the most). Observed shares from `repro/census.mjs` at edition 1; the largest deviation is 1.44 standard errors (one ink). The five non-night papers (blush, ivory, rose, fog, overhead blue at 28/22/20/12/10%) are not reported in the trait line and were not counted.

and bone, used below $D = 0.1$ on every sheet, also takes carbon's place above 0.88, so carbon is not used. A static analogue is the column-structured ordered dither of print halftoning [Ulichney, 1987], with the dither threshold replaced by a field whose photographic weight falls with the edition number.

Inks and traits. The five inks are colors I had already used in earlier on-chain works and listed in a palette file before this project began: pink and pale blue from "together", two blues from "overhead", and a pale blue labeled "mr. blue sky" whose source work I can no longer identify. Bone and carbon come from earlier noise studies, and the single ember dot (#e0521b), one orange dot that 55% of tokens carry, comes from the fourth 2022 test output. Table 1 lists the seeded traits with the weights written in the code and their frequencies over 20,000 seeds; every observed share is within 1.5 standard errors of its weight. Memory is the only trait not drawn from the seed. Each token writes its traits, including the printed memory, into the SVG's `desc` element (line 153), which is the trait line.

3 Measuring what survives

The printed memory is $f(e)$ and nothing more. How much of the photograph a finished edition still shows depends on everything between Eq. (1) and the drawn dots, so I measured it at two stages, the darkness field and the ink.

Two measures. For the field, I rebuilt n for each seed from the first 256 draws of its PRNG, evaluated Eq. (1) on a 75×100 lattice, and computed the Pearson correlation between D and the photograph's darkness $1 - P$, for both photographs on every seed (`repro/field.mjs`). For the drawing, I rendered each (seed, edition) pair with the stored generator, took every dot except the ember, and computed the Spearman rank correlation [Spearman, 1904] between the dot's ink darkness (one minus Rec. 601 luma) and the photograph's darkness at the dot's center (`repro/measure.mjs`). Centers are taken before the texture filter, an SVG displacement filter that roughens each dot's edge and moves it by at most 3 pixels along each axis, about a ninth of a grid cell (up to about 4.2 pixels, a sixth of a cell, on the diagonal). A token's ink darkness takes at most $k + 2$ values (k inks, bone and carbon), so ties alone keep the rank correlation below 1 even at full memory. Night-paper tokens invert the ink order by design, so their correlations are sign-flipped. Neither measure is an information bound: both grids sit in the code, and with the seed's noise

edition	f	field, tree	dots, tree	(IQR)	single-ink %, tree	dots, sky	control, tree
1	1.000	1.000	0.824	(0.741, 0.858)	1.0	0.845	0.067
9	0.836	0.996	0.793	(0.690, 0.835)	2.5	0.814	0.075
15	0.713	0.981	0.753	(0.636, 0.807)	5.5	0.770	0.101
22	0.570	0.938	0.693	(0.545, 0.764)	6.3	0.704	0.060
30	0.407	0.816	0.540	(0.349, 0.659)	8.8	0.528	0.014
34	0.325	0.699	0.433	(0.205, 0.594)	11.0	0.409	0.009
44	0.120	0.267	0.136	(0.000, 0.338)	12.7	0.128	0.000

Table 2: Selected rows of Figure 3. f is the memory factor, which the token prints as a percentage. Field: median Pearson correlation of D with the photograph. Dots: median Spearman correlation of drawn ink darkness with the photograph at each dot, over all renders, with single-ink renders (share given) scored 0. Control: the swapped-photograph render correlated with the tree. Full tables in `data/curve.csv` and `data/field.csv`.

any edition's field can be inverted exactly, $1 - P = (D - 0.85(1 - f)\, n)/f$. The measures say how strongly the photograph shows in the field and in the drawing, which is a different question from whether anyone could see it.

The swapped-photograph control. For each (seed, edition) I also render the same program with only the photograph swapped. The source draw still consumes its random number and every trait except the ember is drawn before the dots, so those traits come out the same (checked on 600 pairs; `renderSwapped` in `repro/runtime.mjs` leaves the stored code unmodified). Inside the dot loop, dot sizes, run lengths and the random draws now follow the other photograph. I then correlate the swapped render's ink with the photograph the real token used. What survives that swap is what the two photographs share and whatever the drawing process adds on its own, so the control is a baseline for a correlation reached without the token's own photograph driving it.

Seeds and pooling. Seeds are $1{,}000{,}003 + 7{,}919\, i$ for $i = 1, \ldots, 1000$, giving 44,000 renders: at every edition, 872 renders from tree tokens and 128 from sky tokens. Every trait drawn before the dot loop is the same at all 44 editions, because the PRNG stream depends on the seed alone; only the printed memory, which is set by the edition number, and the ember, which is drawn after the dots, can differ. So each point on a field curve is a median over the 1,000 seeds at that edition, each point on a dot curve a median over that photograph's 872 or 128 renders, and the 44 points of one seed are one trajectory, not 44 independent draws.

Single-ink renders. In 3,291 renders (7.5%) every dot has the same ink and the rank correlation is undefined. I score these 0, since their ink carries no rank information about the photograph, so every render counts. For tree tokens their share rises from 1.0% at edition 1 to 13.3% at edition 42 (12.7% at edition 44); dropping them instead would raise the edition-44 median from 0.14 to 0.19.

4 What the editions remember

Figure 3 and Table 2 give the curves. The photograph holds and then drops. Between editions 1 and 22 the field's correlation with the tree falls by 0.06; between editions 34 and 44 it falls by

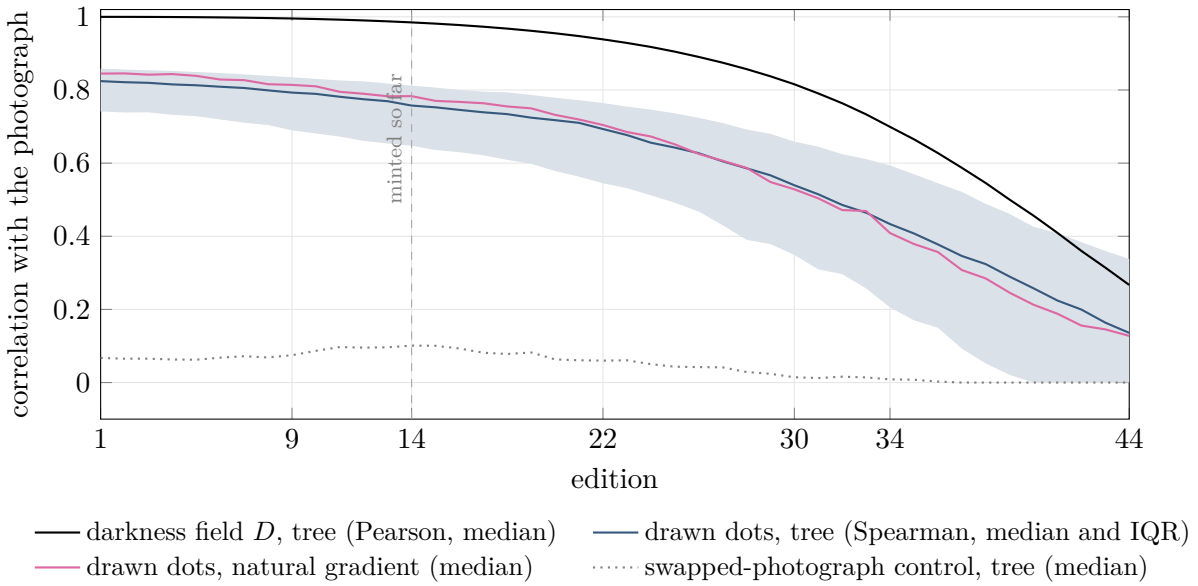


Figure 3: How much of the photograph survives, by edition. The field curve is a Pearson correlation over the whole canvas; the dot curves are Spearman correlations over drawn dots, so the two share a shape but not a scale. Field medians over all 1,000 seeds; dot medians over that photograph's renders, every render counted (872 tree and 128 sky renders per edition); the band is the tree's interquartile range (IQR).

0.43, out of a total loss of 0.73. The drawn dots follow the same shape less sharply: a fall of 0.13 between editions 1 and 22 and 0.30 between 34 and 44.

The closed form. Equation (3) says why. If the noise is uncorrelated with the photograph, Eq. (1) gives

$$\rho_{\text{field}}(f) \;\approx\; \frac{f\,\sigma_p}{\sqrt{f^2\sigma_p^2 + 0.85^2(1-f)^2\sigma_n^2}}, \tag{3}$$

where σ_p is the standard deviation of the photograph's darkness over the canvas and σ_n that of the noise. Measured (`repro/sigma.mjs`): $\sigma_p = 0.2515$ for the tree and 0.2601 for the sky; the median σ_n over 1,000 seeds is 0.1486. While $f\sigma_p$ is large next to $0.85(1-f)\sigma_n$ the noise barely moves the ranking of dark and light; the photograph loses its hold where the two terms of Eq. (1) have equal spread, $f\sigma_p = 0.85(1-f)\sigma_n$, which is $f^* = 0.334$ for the tree and 0.327 for the sky. Eq. (2) reaches those values at edition 33.5 and 33.9, where Eq. (3) gives $1/\sqrt{2} \approx 0.71$. Over all 44 editions the closed form is within 0.010 of the tree's measured curve and within 0.021 of the sky's, and the measurement equals it at edition 1 and sits above it at every later edition; the gap is the covariance between each seed's noise and the photograph, which is zero only on average over seeds (`review/probes-r2.mjs`). Equation (3) is the attenuation of a correlation by additive noise, the identity behind Spearman's correction for attenuation [Spearman, 1904], and it says where the drop falls for this mixture and these two spreads: around edition 34.

What the traits do. At edition 1 the field correlation is exactly 1, and the drawn dots reach a median of 0.82 (tree) and 0.84 (sky). Part of the gap between the two is the tie ceiling and part is the drawing: sparse columns, run persistence, and dots placed where D is dark. Table 3 gives the

trait	value	median drawn correlation, edition 1	renders
density	air / field / full	0.77 / 0.84 / 0.87	499 / 385 / 116
inks	1 / 2 / 3 / 4	0.42 / 0.82 / 0.84 / 0.87	35 / 536 / 313 / 116
paper	night	0.80	85
all		0.83	1,000

Table 3: The drawn measure at edition 1 by trait, both photographs pooled, from `data/renders.csv`. Night-paper medians are after the sign flip.

edition-1 median by trait. Denser drawings and larger palettes keep more; one-ink tokens, which have only bone, carbon and their one ink to rank, keep least. The quantizer also loses levels as the run goes on: carbon needs $D > 0.88$, and from edition 27 no carbon dot appears on light paper in any of the 1,000 seeds, so by the time the steep part of the decline begins, the drawing has one ink level fewer to express it with.

The end of the run. At edition 44 the median drawn correlation is 0.14 for the tree; 62.5% of the 872 tree renders are positive, and 629 exceed their own swapped-photograph control ($z = 13$ against an even split). The tree's medians fall in order; individual renders do not follow one: a random edition-44 tree render scores above a random edition-34 render 26% of the time, and adjacent editions are close to an even split (edition 23 above edition 22: 48%). In a set of 44 tokens actually minted, a later token can easily show more of the photograph than an earlier one. The 12% floor leaves a trace a statistic can find.

The control is near zero but not exactly: its median by edition ranges from 0.000 to 0.101 for tree tokens and to 0.104 for sky tokens, consistent with the two grids' shared large-scale structure (cell correlation 0.10). From edition 37 the tree control's median is exactly 0, because the single-ink control renders, scored 0, fall across the middle of the distribution; the paired comparison above is the better test at those editions.

Ink, size and presence. A second measure aggregates every channel the drawing has: for each of the 1,064 grid cells, the sum over dots centered in it of dot area times the dot's darkness above the paper, correlated (Pearson) with the grid, with the same control (`repro/extra.mjs`). It is much weaker, because most cells hold no dot: 0.20 for the tree at edition 1 against 0.11 for its control, and 0.14 against 0.13 at edition 44. At the five editions where it was computed (1, 9, 22, 34 and 44), the median area score of single-ink renders is below the control's median for tree tokens and at most 0.015 above it for sky tokens, so scoring their drawn measure 0 loses little. The drawn measure separates the photograph from its control more clearly than the area measure does, and it is the one reported above.

The printed memory against the measurement. Each token prints f, rounded to a whole percent, as its memory. Read against the median drawn measure, the printed memory lies above the median through edition 11 (84% at edition 9 against a tree median of 0.79). At edition 12 the tree's memory factor lies 0.0002 above its median, and only rounding it to the printed 77% puts the label below. From edition 13 the printed memory lies below the median for the tree, and from edition 12 for the sky; the one exception is the sky at edition 42, where the median (0.156) falls just below the printed 16%. At edition 44 the token prints 12%, against 0.14 for the tree's dots and 0.27 for its field. A weight and a correlation are different quantities; the trait line invites the

comparison, and these are medians, so an individual render can sit on either side of its label at any edition.

The collection so far. Fourteen of the 44 editions are minted as of 3 November 2026. The most recent is edition 14, with $f = 0.734$, printed as 73%. At that edition number the field's median correlation is 0.985 for tree tokens and 0.986 for sky tokens, and the median drawn correlation is 0.76 for tree tokens and 0.78 for sky tokens. These are medians over seeds, not measurements of the fourteen minted tokens, whose own seeds are listed in `data/provenance.csv` (Section 5). The steep part of the curve lies in editions not yet minted, and whether the piece completes its forgetting depends on whether anyone collects them.

5 What the ledger remembers

Superseded versions. bootloader keeps each generator as one key in its contract's storage, and a version update overwrites that key's current value. The generator's storage key has been written 21 times: once to create it, five times to publish versions 2 to 6, once to open the sale, and fourteen times to record a mint. All 21 writes are listed in `data/generator-history.csv`. Version 1, published at 22:10:28 UTC on 31 July, opened with the comment "in november 2022 i wrote a p5.js sketch called '365 days of tbm' ... this is that algorithm, rewritten to run entirely on-chain." Edition 1 was minted under it four minutes later. That evening I asked for a token-level comparison of my 2022 sketch with Haskell's tutorial: the function that draws the columns was 95% identical to his by token count, and two helpers were identical character for character. Version 4 (03:03 UTC, 1 August) rewrote the header to credit him, versions 5 and 6 added a corrected description, and the public working log says what happened [Meadows, 2026c]. Every earlier value remains in the block that wrote it, and the TzKT indexer still returns all six versions [Baking Bad, 2026]; `repro/verify-chain.mjs` retrieves them. Version 1 survives because archive nodes and indexers keep history, and the false claim stays on that record next to its correction.

Seeds and artifacts. For each token the ledger keeps an edition number, the entropy the platform's random-number contract set at mint, and every version of the token's metadata, including the *artifact*: the stored wrapper that declares the seed the program runs from. For all fourteen minted tokens that seed is the stored entropy with the bits of each byte reversed. The contract converts the 32 entropy bytes to a number by testing each byte's bits from the lowest up and shifting them in, which reads every byte backwards (an entropy byte `0x01` becomes `0x80` in the seed); the platform's published artifact template then seeds its generator exactly as `repro/runtime.mjs` does (splitmix64 into sfc32). Rendering each token's mint-time artifact and comparing it with the platform's own image gives a gray-level correlation of 0.98 to 0.99 for all fourteen (`data/provenance.csv`, `review/thumb-match.mjs`): the platform shows what the chain stores, and every minted token's image is reproducible from public data.

Edition 1 carries a second record of my own making. I asked the platform three times to regenerate edition 1's artifact, after the updates to versions 2, 3 and 6. Each time it stored the seed as 32 raw bytes instead of a number, so the artifact now on chain is not valid UTF-8, is not a program, and draws nothing. The same happens to every token whose holder regenerates it, as recently as 14 October 2026 on another artist's token. The version the mint wrote remains in the block that wrote it (level 14305273) and is the one the platform's thumbnail and viewer show. I

reported the raw-bytes bug on 31 July 2026 [Meadows, 2026a]; it was unresolved when this version was written.

What the record answers. The ledger answers what was stored, when, and by which operation: it dates version 1's header and the rewrite that credits Haskell, and it fixes the seed each mint wrote, from which every minted token's image renders. It does not answer why. The comparison that led to version 4 is in the working log [Meadows, 2026c], not on chain, and nothing on chain marks edition 1's current artifact as unrenderable; that is found only by reading the stored bytes.

6 Keeping and forgetting

The work assigns permanence to the ledger and loss to the editions, and three months of running it tested both halves. The ledger kept every version of the program, the mint-time artifact of every token, and the claim I would have chosen to forget. The editions lose the photograph on a schedule written into the program. A later version could rewrite that schedule, as versions 2 to 6 rewrote other things, but no update can change the edition number a minted token was given, and until the generator is updated again the current program with both grids is part of the live state every node holds. Of what differs between tokens, the seed gives the traits and the noise, and the edition number alone carries a schedule, so that is where the forgetting goes, and Figure 3 is its measure.

That schedule is the reverse of forgetting in a rememberer. Ebbinghaus measured his own relearning savings and found the steepest loss in the first hour [Ebbinghaus, 1885]; a replication reproduced the curve [Murre and Dros, 2015], and a reanalysis found forgetting best described by a power function of time, whose rate of loss falls with time [Wixted and Ebbesen, 1991]. Figure 3 falls late instead, because the schedule runs on a counter and the drop comes where the noise spread overtakes the photograph's. A retention curve and Figure 3 do not measure the same thing: a retention curve follows one rememberer through time, while each edition's level here is set by its edition number, not by elapsed time.

Scheduled loss has company. Opałka's count is the nearest painted sequence: from 1972 each canvas was painted on a ground about 1% whiter than the one before [Opałka, 1965], and Metzger's manifesto of 1959 proposed works made to destroy themselves [Metzger, 1959]. Gibson's poem about his father in *Agrippa (a book of the dead)*, made with Ashbaugh and Begos, ran from a diskette that scrolled it up the screen once and then made it seem to vanish behind an encryption-like effect; Ashbaugh's printed images were meant to fade too, but there was no technical way to do it [Gibson and Ashbaugh, 1992]. The text the diskette withdrew survived all the same: the poem was leaked onto the Internet early on. Lucier rerecorded his voice in a room until the room's resonances replaced the words [Lucier, 1970], and Basinski let decaying tape loops play until their coating fell away [Basinski, 2002]. In Jim Campbell's *Portrait of My Father*, a photograph of his father is visible for an instant and then disappears, over and over, at the rate of the artist's heartbeat as recorded through one night of sleep [Campbell, 1994]; the photograph returns with every beat, where an edition here keeps one fixed share of its photograph. In Frampton's *(nostalgia)*, photographs from his years as a still photographer burn to ash one at a time on an electric burner, and he told Scott MacDonald that they "can be resurrected by rewinding the film" [Frampton, 1971], as the photograph each edition here loses stays in the stored code. Oscar Muñoz's *Narcisos*, begun in 1994, screen-prints his portrait in charcoal dust on water, and the image distorts as the water evaporates and the dust settles to the bottom of the tray [Muñoz, 1994]; in *Re/trato* he paints his portrait in water on a concrete sidewalk and keeps painting as it evaporates [Muñoz, 2003]; in both

the portrait fades while it is being made or shown, where each edition here is one fixed stage of the fading.

So does code on a ledger: generators that live entirely on chain go back at least to Larva Labs' Autoglyphs, whose drawing code sits in its Ethereum contract [Larva Labs, 2019]. On ledgers, Friend's *Lifeforms* die unless their holder gives them away within 90 days [Friend, 2021]; in Reas's *923 EMPTY ROOMS* the mint number selects the composition [Reas, 2023], and in Gerrard's *World Flag* release order follows each country's greenhouse-gas ranking [Gerrard, 2023]. The closest precedent for a programmed decline is 0xDEAFBEEF's *Entropy* series, whose contract increments a counter on every transfer and whose on-chain program degrades the output as that counter rises [0xDEAFBEEF, 2021]; the artist's own note calls the process artificial, "in that no information is actually lost", since the program and every state remain on chain. Mathcastles' *Terraforms* was written to decay by the clock: its contract computes whole years of decay from the block's timestamp and hands them to every token's renderer, but each token that enters its dreaming mode postpones the start by ten years, and 500 of them cancel it [Mathcastles, 2021]. Its holders passed that count (the contract's counter read 1,914 on 17 November 2026), so it does not decay. In *Entropy* the schedule runs on ownership changes and in *Terraforms* it would have run on time; here it runs on mint order, and I know of no on-chain edition keyed to mint order that fades, though I did not survey the field systematically.

What the ledger kept is the other half. A ledger extended only by appending blocks [Nakamoto, 2008, Wood, 2014, Goodman, 2014] offers no operation that removes a past state; a contract can overwrite the current value of a key, but every earlier value stays in the block that wrote it, as long as someone keeps the history. Not every node does: in rolling mode a Tezos node keeps only recent blocks, and in full mode it keeps every block but prunes old state [Octez developers, 2026]. Mayer-Schönberger [2009] argues that digital storage made remembering the default and forgetting the exception. Chun [2008] argues that a conflation of memory with storage underlies and undermines digital media's archival promise, since memory, "with its constant degeneration, does not equal storage". Finck sets out why an append-only ledger cannot comply with a right to erasure in any straightforward way [Finck, 2018], and the redactable blockchains of Ateniese et al. [2017] are a proposal to add the mutability an append-only ledger lacks. The record of version 1 is a small instance of what that means from the author's side.

7 Reproducibility and limits

Everything in Sections 2 to 4 is regenerated by `make` in the paper's directory. The package is archived on Zenodo at `https://doi.org/10.5281/zenodo.23333265` and maintained at `https://github.com/TheBenMeadows/essentialism-preprint`.

The chain side. `repro/essentialism.chain.js` is the generator, byte-identical to version 6 on Tezos (contract `KT1CB4MYiAViCuXWBU961x7LjQXGeA8SnQwt`, big map 771404, key 407), SHA-256 `33d85a12cf2a8404bccda1e31d6b3fd72c361673f995cc58b0857eabb6d5c248`. `repro/verify-chain.mjs` fetches the stored code through TzKT, checks the digest, and writes the version history; `repro/provenance.mjs` fetches every token's metadata versions. `make verify` runs those two. `review/thumb-match.mjs` renders each stored artifact that carries a valid seed in headless Chromium (Playwright 1.62.0) and correlates it with the platform's thumbnail: each minted token's mint-time artifact matches the platform's image at a gray-level correlation of 0.98 to 0.99 (Section 5); rerun `repro/provenance.mjs` after it. These three are the only steps that need the network.

The local side. `repro/runtime.mjs` stands in for the bootloader runtime (splitmix64 into

sfc32, and a minimal DOM that serializes SVG). For seed 1,007,922 at edition 1 its output is byte-identical to that of the renderer used during development, and its PRNG matches the platform's published artifact template line for line. The measurement scripts (`repro/census.mjs`, `field.mjs`, `sigma.mjs`, `measure.mjs`, `extra.mjs`, `plotdata.mjs`, `stats.mjs`, `figures.mjs`) are deterministic with seeds fixed in the scripts; the full measurement takes under a minute with Node.js 26.11 on an Apple-silicon Mac. `data/` holds every CSV the paper reads, the raw TzKT responses, the measured numbers quoted in the text (`data/stats.txt`) and `SHA256SUMS`.

Two things are not reproducible from this package. The photographs' full-resolution files are not included; the published images are linked in the references, and `repro/embed-photo.py` is the script that produced the grids, but the published images may not match the exact crop and export I used, so reapplying it will not reproduce the stored strings byte for byte. The stored strings are the authority. And the texture in Figure 1 depends on Inkscape's SVG filter implementation; browsers render the same geometry with slightly different grain.

The measurement is statistical. A rank correlation between ink darkness and a grid measures association; it does not say that a person would recognize a tree, and I ran no viewer study. Harmon [1973] found that a portrait of 16×16 squares at 8 or 16 gray levels was close to the coarsest from which a face could still be identified, and Harmon and Julesz [1973] found that noise in the band next to such a portrait's own spatial frequencies suppresses recognition considerably more than the high-frequency noise the blocks introduce. The grids here are 28×38 cells at 16 levels, but they hold a tree and a sky, not faces, and neither result says how recognizable an edition is. A perceptual measure such as SSIM [Wang et al., 2004] on a rasterized render would answer a nearer question; I used rank correlation because it needs no rasterizer and no choice of scale. The field and drawn measures use different statistics on different supports (Pearson over the whole canvas, Spearman over the dots), so they are comparable in shape, not in level. The control is not zero, so not every part of a reported correlation is due to the photograph the token sampled, and the control does not say how much is. The seeds are an arithmetic sequence rather than a random sample of the 64-bit seed space; the trait frequencies in Table 1 match the coded weights, which is consistent with an unbiased sequence but does not prove one. The 44,000 renders are 1,000 seed trajectories through the 44 editions, and the sky's drawn-dot curves rest on 128 of them (the field curves use all 1,000 for both photographs). Thirty of the 44 editions do not exist yet. Every statement here about editions 15 to 44 is about what the generator will produce for them, not about tokens anyone holds, and the part of the curve that carries most of the forgetting is in that range.

Acknowledgements

The column structure of this work comes from Andy Haskell's tutorial [Haskell, 2022], credited in the on-chain code since generator version 4.

Use of AI tools

The port of my 2022 sketch to bootloader's runtime (the runtime translation, noise function, grid embedding, fade arithmetic and texture filters), the measurement and audit scripts, and the text of this paper were drafted with Claude (Anthropic) in Claude Code, directed by me. The photographs, the ink library, the premise of a fading memory, the edition count and the corrections in Section 5 are my decisions, and I take responsibility for all content. Drafts were reviewed by other language

models through the Devin, Cursor and Kagi command-line tools; every measurable defect a reviewer reported was reproduced or refuted by a script before it changed the paper, and the probe and audit scripts that check the measured numbers and the chain facts in the text against the data are in the repository. Every reference was checked against the work itself or a published description of it.

References

0xDEAFBEEF. Series 3: Entropy. On-chain audiovisual series, Ethereum; a counter incremented on each transfer degrades the output, 2021. `https://www.deafbeef.com/series/3`.

Giuseppe Ateniese, Bernardo Magri, Daniele Venturi, and Ewerton Andrade. Redactable blockchain – or – rewriting history in Bitcoin and friends. In *2017 IEEE European Symposium on Security and Privacy (EuroS&P)*, pages 111–126, 2017. doi: 10.1109/EuroSP.2017.37.

Baking Bad. TzKT: Tezos blockchain indexer and API. `https://api.tzkt.io`, 2026. Accessed 2026-10-12.

William Basinski. The Disintegration Loops. Audio recording, 2062, 2002. Discogs release 75994.

Jim Campbell. Portrait of My Father. Custom electronics, glass, photograph, LCD material, 1994–95, from the series Memory Works; collection of the San José Museum of Art. `https://www.jimcampbell.tv/portfolio/photo-of-my-mother-portrait-of-my-father`, 1994.

Wendy Hui Kyong Chun. The enduring ephemeral, or the future is a memory. *Critical Inquiry*, 35 (1):148–171, 2008. doi: 10.1086/595632.

Chris Doty-Humphrey. PractRand: a C++ library of pseudo-random number generators and statistical tests, including the sfc32 generator. `https://pracrand.sourceforge.net/`. Software.

Hermann Ebbinghaus. *Über das Gedächtnis: Untersuchungen zur experimentellen Psychologie*. Duncker & Humblot, Leipzig, 1885.

Michèle Finck. Blockchains and data protection in the European Union. *European Data Protection Law Review*, 4(1):17–35, 2018. doi: 10.21552/edpl/2018/1/6.

Hollis Frampton. (nostalgia). 16mm film, black and white, sound, 36 min; first part of *Hapax Legomena*. Distributor record: `https://lux.org.uk/work/002299-nostalgia/`; National Film Preservation Board essay: `https://www.loc.gov/static/programs/national-film-preservation-board/documents/nostalgia.pdf`, 1971.

Sarah Friend. Lifeforms. Tokens on Polygon, contract `0x8916edd9b39783d85303ecc6613917ddd735d88d`. `https://lifeforms.supply/`, 2021.

John Gerrard. World Flag. Art Blocks and Pace, contract `0xea698596b6009a622c3ed00dd5a8b5d1cae4fc36`, project 6, 2023.

William Gibson and Dennis Ashbaugh. Agrippa (a book of the dead). Artist's book with diskette, published by Kevin Begos, Jr. Documentation and emulation: The Agrippa Files, University of California, Santa Barbara, `https://agrippa.english.ucsb.edu/`, 1992.

L. M. Goodman. Tezos: A self-amending crypto-ledger. https://github.com/tezos/tezos-papers, 2014. White paper; source white_paper_original.tex.

Leon D. Harmon. The recognition of faces. *Scientific American*, 229(5):70–83, 1973. doi: 10.1038/scientificamerican1173-70.

Leon D. Harmon and Bela Julesz. Masking in visual recognition: Effects of two-dimensional filtered noise. *Science*, 180(4091):1194–1197, 1973. doi: 10.1126/science.180.4091.1194.

Andy Haskell. Convert images to mosaics in p5.js. https://dev.to/andyhaskell/convert-images-to-mosaics-in-p5js-2dlc, January 2022. Tutorial; finished sketch at https://editor.p5js.org/andyhaskell/sketches/fwaWSmAh_.

Larva Labs. Autoglyphs. Ethereum, contract 0xd4e4078ca3495de5b1d4db434bebc5a986197782. https://www.larvalabs.com/autoglyphs, 2019.

Alvin Lucier. I Am Sitting in a Room, 1970. Sound work for voice and electromagnetic tape. Recording: Lovely Music VR 1013 (LP), 1981.

Mathcastles. Terraforms. Ethereum, contract 0x4E1f41613c9084FdB9E34E11fAE9412427480e56; verified source at https://repo.sourcify.dev/1/0x4E1f41613c9084FdB9E34E11fAE9412427480e56, 2021.

Viktor Mayer-Schönberger. *Delete: The Virtue of Forgetting in the Digital Age*. Princeton University Press, Princeton, NJ, 2009. ISBN 9780691138619.

Benjamin Meadows. Blue Tree. Photograph, 1 of 1, minted on Ethereum 2023-09-06. Contract 0x93ad4670D8Dd9FC3987EB296315827C3D138ADeA, token 5. https://eth.blockscout.com/token/0x93ad4670D8Dd9FC3987EB296315827C3D138ADeA/instance/5, 2023a.

Benjamin Meadows. Natural Gradient. Photograph, 1 of 1, minted on Ethereum 2023-09-17, metadata on Arweave. Contract 0x93ad4670D8Dd9FC3987EB296315827C3D138ADeA, token 6. https://eth.blockscout.com/token/0x93ad4670D8Dd9FC3987EB296315827C3D138ADeA/instance/6, 2023b.

Benjamin Meadows. Seed handling: regenerate_token stores raw bytes in artifactUri (invalid JS/UTF-8); stored artifact seed differs from frontend render seed. https://github.com/objkt-com/bootloader-monorepo/issues/30, 2026a. GitHub issue, opened 2026-07-31.

Benjamin Meadows. essentialism. Generative work, bootloader svg-js generator 407, contract KT1CB4MYiAViCuXWBU961x7LjQXGeA8SnQwt on Tezos. https://bootloader.art/generator/svg-js/407, 2026b. Published 2026-07-31; generator version 6 since 2026-08-01.

Benjamin Meadows. essentialism: the working log. https://thebenmeadows.com/blog/essentialism-the-working-log/, 2026c. Published 2026-07-31, corrected 2026-08-26.

Gustav Metzger. Auto-destructive art. Manifesto, London, 1959. Dated 4 November 1959.

Oscar Muñoz. Narcisos. Series begun 1994; charcoal dust screen-printed on water. One example: coal dust over glass over Plexiglas, 1995, Metropolitan Museum of Art 2019.184, https://www.metmuseum.org/art/collection/search/827985, 1994.

Oscar Muñoz. Re/trato. Video, 28 min. Museum of Contemporary Art, Los Angeles, 2004.95, https://www.moca.org/artworks/retrato; described at High Line Art, https://www.thehighline.org/art/projects/oscarmunoz/, 2003.

Jaap M. J. Murre and Joeri Dros. Replication and analysis of Ebbinghaus' forgetting curve. *PLoS ONE*, 10(7):e0120644, 2015. doi: 10.1371/journal.pone.0120644.

Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system. https://bitcoin.org/bitcoin.pdf, 2008.

objkt.com. bootloader: on-chain generative art platform on Tezos. https://bootloader.art; source: https://github.com/objkt-com/bootloader-monorepo, 2026. Accessed 2026-10-12.

Octez developers. History modes. Octez and Protocol documentation, https://octez.tezos.com/docs/user/history_modes.html, 2026. Accessed 2026-10-12.

Roman Opałka. OPALKA 1965/1 – ∞. Series of numbered paintings, 1965–2011. Description of the progressive whitening of the grounds: Christie's lot essay, https://www.christies.com/en/lot/lot-6100726, 1965.

Ken Perlin. An image synthesizer. *ACM SIGGRAPH Computer Graphics*, 19(3):287–296, 1985. doi: 10.1145/325165.325247.

Casey Reas. 923 EMPTY ROOMS. Art Blocks and Bright Moments, contract 0x145789247973c5d612bf121e9e4eef84b63eb707, project 1. https://www.artblocks.io/collection/923-empty-rooms-by-casey-reas, 2023.

Charles Spearman. The proof and measurement of association between two things. *The American Journal of Psychology*, 15(1):72–101, 1904. doi: 10.2307/1412159.

Guy L. Steele, Doug Lea, and Christine H. Flood. Fast splittable pseudorandom number generators. In *Proceedings of the 2014 ACM International Conference on Object Oriented Programming Systems Languages & Applications (OOPSLA)*, pages 453–472, 2014. doi: 10.1145/2660193.2660195.

Robert Ulichney. *Digital Halftoning*. MIT Press, Cambridge, MA, 1987.

Zhou Wang, Alan C. Bovik, Hamid R. Sheikh, and Eero P. Simoncelli. Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4): 600–612, 2004. doi: 10.1109/TIP.2003.819861.

Sam Williams, Viktor Diordiiev, Lev Berman, India Raybould, and Ivan Uemlianin. Arweave: A protocol for economically sustainable information permanence. https://www.arweave.org/yellow-paper.pdf, 2019. Yellow paper, draft 1.

John T. Wixted and Ebbe B. Ebbesen. On the form of forgetting. *Psychological Science*, 2(6): 409–415, 1991. doi: 10.1111/j.1467-9280.1991.tb00175.x.

Gavin Wood. Ethereum: A secure decentralised generalised transaction ledger. https://ethereum.github.io/yellowpaper/paper.pdf, 2014. Yellow paper, first published 2014; the URL serves the current revision.