OPEN AGENCY

Technical Handbook

AI Scraping, Latent-Space Training,
Cryptographic Provenance, and Creator-Side Defences

A Technical Companion to the Open Agency Website

Indian Institute of Technology Delhi
COL864 — Special Topics in AI

Open Agency is an open-source project. The website is written for a non-technical audience; this handbook is its technical companion. The text in this book is released under CC BY-SA 4.0; the accompanying website code is released under MIT. This document was prepared for COL864 (Special Topics in AI) at IIT Delhi.

This is a working document. Generative-model training and provenance standards are changing quickly. Empirical claims are cited; structural claims are stated as analysis. Corrections are welcome.

Preface

Open Agency has two parts. The website is the simple public-facing layer. It explains AI scraping, consent, and creator-side protection tools in plain language for a non-technical audience.

This handbook is the technical layer attached to the website. It is for readers who want to understand the mathematical and engineering background behind the website. We assume basic comfort with probability, gradients, vectors, and matrices. We do not assume prior expertise in machine learning, cryptography, or information theory.

The structure is simple:

> *Explain the idea. Write the formal version. Cite the source.*

Chapter 1 explains how web-scale scraping pipelines collect image–text data and why `robots.txt` and Terms-of-Service clauses are weak consent mechanisms. Chapter 2 explains latent diffusion models, denoising, CLIP conditioning, cross-attention, and why removing one image from a trained model is technically difficult. Chapter 3 explains C2PA Content Credentials as signed, tamper-evident provenance objects. Chapter 4 explains the technical ideas behind C2PA's 'Do Not Train' assertion, Glaze, and Nightshade.

The goal is not to make broad claims about AI. The goal is to be precise about one problem: how creator consent can be expressed, ignored, protected, or made technically visible in modern generative-AI pipelines.

Contents

CHAPTER 1

The Scraping Pipeline

1.1 Anatomy of a web-scale dataset

When a generative model *learns the visual world*, what it actually does is consume a list of URLs and the captions associated with them. The list is enormous and the captions are noisy, but the substrate is exactly that prosaic. The largest open dataset of this kind is **LAION-5B** (Schuhmann et al., 2022), which catalogues approximately 5×10^9 image–text pairs gathered from the open web. LAION-5B does not store the images themselves: it stores a CSV-like manifest of `(URL, caption, similarity_score, dimensions, language)` tuples, and the model trainer is responsible for fetching each image at training time.

The pipeline from a public webpage to a row in this manifest is, in idealised form:

1. A web crawler (most often Common Crawl, occasionally bespoke) walks the public web, following hyperlinks and saving HTML.
2. A parser extracts every `<img>` tag, pairing the image URL with the surrounding caption text — typically the `alt` attribute or the nearest heading.
3. The pair is filtered: a pre-trained CLIP model (Radford et al., 2021) computes the cosine similarity between the image embedding and the caption embedding, and the pair is retained only if that similarity exceeds a threshold τ. In LAION-5B, $\tau \approx 0.28$ for the English subset.
4. Surviving pairs are deduplicated, language-tagged, and indexed.

What this filtering does *not* do is check copyright, consent, or provenance. The only filter that is applied — the CLIP-similarity threshold — is a quality gate: it discards pairs where the caption does not describe the image. A copyrighted image with an accurate caption sails through. A photograph of an identifiable person with a descriptive ALT attribute sails through. The Glaze authors note, summarising the public record, that Midjourney's founder has acknowledged training on roughly 10^8 images without consent (as cited by Shan et al., 2023).

Once a manifest exists, anyone with bandwidth can recreate the dataset by re-crawling the URLs. This is what gives modern training pipelines their reach: *the dataset is a list, the list is public, and the cost of materialising it is bandwidth, not negotiation.*

1.2 The robots exclusion protocol and its limits

The standard most often invoked as 'the opt-out' for web scraping is `robots.txt`, codified in 2022 as RFC 9309 (Koster et al., 2022). The format is austere. A site publishes a file at the well-known path

`/robots.txt` containing rules of the form

```
User-agent: GPTBot
Disallow: /
```

which a compliant crawler is expected to read before fetching any URL on the host. There are four structural reasons `robots.txt` cannot serve as an AI-training opt-out, and they are worth naming carefully:

1. **Voluntary compliance.** The protocol has no enforcement mechanism. Honouring it is a courtesy, not a contract. Major labs (OpenAI, Google, Anthropic) have publicly committed to honouring named user-agents; smaller scrapers, dataset assemblers, and academic projects have no equivalent commitment.
2. **No verifiability.** The dataset is not published, and even when it is, the URLs alone cannot prove negative inclusion. There is no way for a creator to confirm that their site was excluded from a given training run.
3. **Site-level granularity.** A single `robots.txt` governs an entire host. There is no per-image, per-asset directive in the standard. A creator who wishes to share some work publicly while withholding other pieces from training cannot express that distinction.
4. **No identity binding.** The directive is keyed to a user-agent string, which is a self-declared label sent by the crawler. A non-compliant scraper can simply identify as `Mozilla/5.0` and fetch whatever it likes.

The protocol was designed in 1994 to manage server load from search engines. Treating it as the consent layer for generative-AI training is a category error.

1.3 The contractual surface

The legal argument typically advanced by platforms is that hosting clauses in their Terms of Service authorise the use of uploaded content as training data, on the theory that model training is a form of 'data processing' or 'service improvement'. Two empirical facts complicate this:

- Studies of Terms-of-Service readership consistently put the rate at which users read agreements before accepting them in the single digits.
- The clauses are typically broad enough to authorise almost any future use, including uses that did not exist when the agreement was signed. Consent under such conditions is, in the language of contract scholars, *procedural* rather than *substantive*.

The NIST AI Risk Management Framework (National Institute of Standards and Technology, 2023) captures the same observation more carefully when it distinguishes *informed* consent from *nominal* consent and treats the latter as a governance failure rather than a defence.

1.4 What "public" should and should not authorise

It is worth stating the philosophical point explicitly because the rest of the book depends on it. Posting a portrait so one's friends can see it is an act of *publication*. Donating that portrait to a model that will let strangers generate new portraits in the same person's likeness on demand is something else — closer to *reconstitution*, in the sense that the identifying features of the subject are extracted and made available to be reused without further authorisation.

The two acts share a surface (the image is online and visible) but differ at every other level: in their reach, their commercial structure, the duration of their effects, and the bargaining position of the original creator. Treating them as the same act because they share a surface is, again, a category error. Most of what follows in this book is an attempt to give creators technical tools that make the distinction visible to the systems that currently elide it.

CHAPTER 2

Latent Space and the Mathematics of Training

This chapter is the technical core of the book. The reader who is comfortable with the question *what does it mean for a diffusion model to "learn" a concept?* can skim it. The reader who is not, but who wants a precise answer to *why is removing my image from a trained model effectively impossible?*, should read it carefully. The argument has four moving parts: a representation step (the variational autoencoder), a generative process (diffusion), a conditioning mechanism (CLIP and cross-attention), and a structural fragility (the implosion phenomenon).

2.1 From pixels to latents: the variational autoencoder

A modern image model does not operate on raw pixels. The reason is computational: a 512×512 RGB image has 786,432 degrees of freedom, and a diffusion process running directly in that space is prohibitively expensive. Latent-diffusion models (Rombach et al., 2022) avoid this by first compressing the image into a much smaller *latent* representation using a **variational autoencoder** (VAE) (Kingma and Welling, 2013).

A VAE consists of a probabilistic encoder $q_\phi(z \mid x)$ and a decoder $p_\theta(x \mid z)$, where $x \in \mathbb{R}^{H \times W \times 3}$ is an image and $z \in \mathbb{R}^{h \times w \times c}$ is a low-dimensional latent (Stable Diffusion uses $h = w = 64$, $c = 4$, an $8\times$ spatial reduction). The encoder is trained to map images to a distribution over latents; the decoder is trained to reconstruct the image from a sampled latent. The learning objective is the **evidence lower bound** (ELBO):

$$\mathcal{L}_{\text{ELBO}}(x) \;=\; \mathbb{E}_{z \sim q_\phi(z \mid x)}\big[\log p_\theta(x \mid z)\big] \;-\; \mathrm{KL}\big(q_\phi(z \mid x) \,\big\|\, p(z)\big), \tag{2.1}$$

where $p(z) = \mathcal{N}(0, I)$ is a standard Gaussian prior. The first term rewards faithful reconstruction; the second term, the Kullback–Leibler divergence, regularises the encoder so that the posterior remains close to the prior. The crucial property the VAE buys us is that the latent space $\mathbb{R}^{h \times w \times c}$ is *semantically organised*: similar images sit close together, dissimilar images sit far apart, and small movements in latent space correspond to small, plausible changes in image space.

For Stable Diffusion 1.x and 2.x, the VAE is fixed during downstream training: it is a *pre-trained tokeniser* for images, much as CLIP serves as a pre-trained tokeniser for text. *For full derivations and proofs, see Kingma and Welling (2013).*

2.2 Diffusion models: learning by denoising

Diffusion models (Sohl-Dickstein et al., 2015; Ho et al., 2020) are the dominant generative architecture for images. Their training objective is unusual but has a clean intuition: *teach a network to undo a small step of noise; then chain together many small denoising steps to convert pure noise into a sample*. The full process consists of two coupled Markov chains.

2.2.1 The forward process

Given a clean datapoint x_0 (in latent-diffusion, x_0 is the encoded latent z), the forward process gradually adds Gaussian noise over T steps:

$$q(x_t \mid x_{t-1}) \;=\; \mathcal{N}\big(x_t;\, \sqrt{1-\beta_t}\, x_{t-1},\, \beta_t I\big), \qquad t = 1, \dots, T, \tag{2.2}$$

where $\{\beta_t\}_{t=1}^{T}$ is a fixed variance schedule with $0 < \beta_t < 1$. A key property, derived in Ho et al. (2020), is that this chain admits a closed-form expression for any single step:

$$q(x_t \mid x_0) \;=\; \mathcal{N}\big(x_t;\, \sqrt{\bar{\alpha}_t}\, x_0,\, (1-\bar{\alpha}_t) I\big), \qquad \bar{\alpha}_t \;=\; \prod_{s=1}^{t} (1-\beta_s). \tag{2.3}$$

Equivalently, one can write

$$x_t \;=\; \sqrt{\bar{\alpha}_t}\, x_0 \;+\; \sqrt{1-\bar{\alpha}_t}\, \epsilon, \qquad \epsilon \sim \mathcal{N}(0, I). \tag{2.4}$$

This is the reparameterisation that makes training tractable: from any x_0 one can sample a noisy x_t in a single step, with no need to simulate the chain.

2.2.2 The reverse process

The generative direction is harder: we want to sample from $q(x_{t-1} \mid x_t)$, which in general has no closed form. We approximate it with a learned Gaussian:

$$p_\theta(x_{t-1} \mid x_t) \;=\; \mathcal{N}\big(x_{t-1};\, \mu_\theta(x_t, t),\, \Sigma_\theta(x_t, t)\big). \tag{2.5}$$

In Ho et al. (2020), the variance Σ_θ is fixed and the mean μ_θ is reparameterised as a noise prediction $\epsilon_\theta(x_t, t)$. Optimising the variational bound and dropping a constant collapses the loss to

$$\mathcal{L}_{\text{simple}}(\theta) \;=\; \mathbb{E}_{t,\, x_0,\, \epsilon}\Big[\big\| \epsilon \;-\; \epsilon_\theta(x_t, t) \big\|_2^2\Big], \tag{2.6}$$

where x_t is constructed from x_0 via Eq. 2.4. The model is doing one task: *given a noisy image and a timestep, predict the noise that was added*. This is the entire training signal. *For full derivations and proofs, see Ho et al. (2020).*

A practical consequence is that training is *distributed*: every gradient step updates millions of parameters in ϵ_θ on the basis of a single batch, and a single image's contribution to the final weights is the cumulative effect of being one of many examples that shaped the noise predictor's behaviour in some neighbourhood of latent space. There is no row of a database that can be deleted.

2.3 Conditioning on language: CLIP and cross-attention

A model that only generates images would not be very useful. To make text-to-image generation possible, the diffusion process is *conditioned* on a textual prompt y. This is done in two stages: first the prompt is encoded into a vector representation, then that representation is injected into the denoising network through cross-attention.

2.3.1 The CLIP text encoder

CLIP (Radford et al., 2021) is a joint image–text encoder trained on ∼400 million image–caption pairs scraped from the web. Let f_θ be the image encoder and g_ϕ the text encoder. CLIP's training objective is contrastive: in a batch of N pairs $\{(x_i, y_i)\}_{i=1}^N$, the model is rewarded for assigning high cosine similarity to matching pairs and low similarity to mismatching pairs. The symmetric InfoNCE loss is:

$$\mathcal{L}_{\text{CLIP}} = -\frac{1}{2N}\sum_{i=1}^{N}\left[\log \frac{\exp(\langle f_\theta(x_i),\, g_\phi(y_i)\rangle/\tau)}{\sum_{j=1}^{N}\exp(\langle f_\theta(x_i),\, g_\phi(y_j)\rangle/\tau)} + \text{(symmetric)}\right], \tag{2.7}$$

where $\langle\cdot,\cdot\rangle$ is the inner product on unit-normalised embeddings and τ is a learned temperature. After training, the text encoder g_ϕ produces a sequence of token embeddings $\mathbf{c} = g_\phi(y) \in \mathbb{R}^{L\times d}$ for any prompt y.

For full derivations and proofs, see Radford et al. (2021).

2.3.2 Cross-attention in the denoising U-Net

The denoising network $\epsilon_\theta(x_t, t, \mathbf{c})$ in latent-diffusion is a U-Net augmented with cross-attention layers that ingest the text embedding $\mathbf{c}$ (Rombach et al., 2022). At each cross-attention layer, the spatial features of the image are projected to a set of *queries* Q, and the text embeddings are projected to *keys* K and *values* V:

$$Q = W_Q\,\varphi(x_t), \quad K = W_K\,\mathbf{c}, \quad V = W_V\,\mathbf{c}, \tag{2.8}$$

where $\varphi(x_t)$ is the spatial feature map at that layer and W_Q, W_K, W_V are learned projection matrices. The attention output, following Vaswani et al. (2017), is

$$\text{Attention}(Q,K,V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right)V, \tag{2.9}$$

where d_k is the per-head dimensionality. Eq. 2.9 is what *aligns* text and image: each spatial location in the image is told, softly, which text tokens it should attend to. Over the course of training, the projections W_Q, W_K, W_V specialise so that the token `"heron"` reliably activates regions of the latent that decode to bird-shaped pixels.

This is the mechanism that the implosion analysis (next section) takes apart formally. *For full derivations and proofs, see Vaswani et al. (2017) and Rombach et al. (2022).*

2.4 Why removal is intractable

Three properties of the pipeline above conspire to make image-level deletion practically impossible.

1. **Distributed encoding.** A single training image x_i contributes a single gradient step to ϵ_θ. After T training steps, ϵ_θ's parameters are a function of all T batches. There is no decomposition $\theta = \sum_i \theta^{(i)}$ that would let us subtract the contribution of x_i.
2. **Continuous training.** Production models are not trained once. Stable Diffusion versions 1.4, 1.5, and 2.1 were each fine-tuned from their predecessors; SDXL was fine-tuned from SDXL 0.9; commercial fine-tuning services apply LoRA adapters on top of base weights. Even if a removal procedure existed for one snapshot, it would have to be repeated for every descendant.
3. **Style as distributed pattern.** Style mimicry, the harm most often cited by artists (Shan et al., 2023), does not require that any single image be 'in' the model. A style is captured by the joint distribution of features across many images; a few dozen examples can be enough for fine-tuning a LoRA that reproduces an artist's idiom.

The literature on *machine unlearning* attempts to address property (1), but as Shan et al. (2024) note, current unlearning methods do not scale to generative models with billions of parameters. The honest answer to "can my image be removed from this model?" remains, in 2026, *not in any verifiable way*.

2.5 A formal view of fragility: the implosion phenomenon

A surprising recent result — and the one that connects the theory of training to the practice of defence — is that the same alignment mechanism we just described is structurally fragile. Ding et al. (2024) formalise text-to-image training as a **supervised graph alignment** problem and show that the alignment can be destroyed with a quantifiable amount of poisoned data.

The construction is roughly as follows. Let G_{img} be a graph whose vertices are visual embeddings (from the fixed VAE) and whose edges are weighted by visual similarity. Let G_{txt} be the analogous graph over textual embeddings (from CLIP). Cross-attention learns a mapping $\theta : V(G_{\text{txt}}) \to V(G_{\text{img}})$ supervised by training pairs. The authors define an **Alignment Difficulty** (AD) metric that combines image-text alignment and image-image alignment under poisoning; their main theorem shows that AD increases with the number of poisoned concepts, and that beyond a threshold, no mapping θ can satisfy the supervision. The empirical consequence: as few as a few hundred carefully-crafted poisoned samples per concept can collapse generation across that concept and *neighbouring* concepts (e.g., poisoning "dog" bleeds into "puppy", "husky").

Two implications matter for the rest of this book.

- Generative models are not robust artefacts. They are statistically tuned objects with measurable failure modes that defenders can target.
- The same property that makes scraping cheap — minimal curation, broad ingestion — makes the resulting models exposed to adversarial data injected by the very people whose work they take. This is the technical foundation for Nightshade, which we return to in Chapter 4.

For full derivations and proofs, see Ding et al. (2024).

CHAPTER 3

Cryptographic Provenance: The C2PA Standard

The previous chapter argued that, once an image is in a training run, asking for it back is structurally hopeless. The constructive question is therefore: *what kind of signal can a creator attach to a piece of content that a downstream system cannot ignore, forge, or strip without detection?* The honest answer involves cryptography. This chapter explains the primitives, the standard that combines them (C2PA), and the precise object that standard produces (a Content Credential).

3.1 The cryptographic primitives

We need three primitives: a hash function, a public-key signature scheme, and an X.509 certificate.

3.1.1 Hash functions

A cryptographic hash function $\mathrm{H} : \{0,1\}^* \to \{0,1\}^n$ maps arbitrary-length inputs to fixed-length digests (in current C2PA practice, $n = 256$ via SHA-256 or $n = 384$ via SHA-384). Three properties are demanded:

- **Pre-image resistance:** given h, finding x with $\mathrm{H}(x) = h$ is infeasible.
- **Second-preimage resistance:** given x, finding $x' \neq x$ with $\mathrm{H}(x') = \mathrm{H}(x)$ is infeasible.
- **Collision resistance:** finding any pair (x, x') with $\mathrm{H}(x) = \mathrm{H}(x')$ and $x \neq x'$ is infeasible.

The hash plays one role throughout C2PA: it produces a short fingerprint of an asset (or of structured data) such that any modification, however small, changes the fingerprint. *For full derivations and proofs, see Katz and Lindell (2014).*

3.1.2 Public-key signatures

A digital signature scheme is a triple of algorithms $(\mathrm{KeyGen}, \mathrm{Sign}, \mathrm{Verify})$:

$$(sk, pk) \leftarrow \mathrm{KeyGen}(1^\lambda), \tag{3.1}$$

$$\sigma \leftarrow \mathrm{Sign}(sk, m), \tag{3.2}$$

$$b \leftarrow \mathrm{Verify}(pk, m, \sigma) \in \{0,1\}. \tag{3.3}$$

The security definition required is **existential unforgeability under chosen-message attack** (EUF-CMA): an adversary with access to a signing oracle for messages of its choice cannot produce a

valid signature on any new message with non-negligible probability. The two schemes used in practice for C2PA are RSA-PSS (Rivest et al., 1978) and ECDSA over standardised elliptic curves; both are EUF-CMA secure under standard assumptions (Katz and Lindell, 2014). Conceptually, the public-key idea originates with Diffie and Hellman (1976).

The asymmetry that matters is: *sk is held by the signer alone; pk is published.* Anyone can verify; only the signer can produce. This is the hinge that makes provenance enforceable rather than aspirational.

3.1.3 X.509 certificates

A signature on its own proves nothing: it binds a message to a public key, but says nothing about *who* that public key belongs to. Public-key infrastructure (PKI) closes the gap by introducing X.509 certificates (Cooper et al., 2008). A certificate C is a structured record signed by a trusted Certificate Authority (CA), containing at minimum:

- A subject identifier (the entity whose key this is),
- The subject's public key pk_{subject},
- A validity period,
- A signature $\sigma_{\text{CA}} = \text{Sign}(sk_{\text{CA}}, \dots)$ over the above.

A verifier who already trusts the CA's key can validate σ_{CA} and thereby *trust* the binding from `subject` to pk_{subject}. CAs themselves can be certified by higher authorities, producing a **chain of trust** that terminates at a root key embedded out-of-band in the verifier's trust store. C2PA maintains its own published trust list as part of its Conformance Programme (Coalition for Content Provenance and Authenticity Technical Working Group, 2024, 2025).

3.2 Anatomy of a C2PA Manifest

A **C2PA Manifest** (also called a Content Credential) is the cryptographic envelope that travels with an asset. The C2PA specification (Coalition for Content Provenance and Authenticity Technical Working Group, 2024) defines it as a triple

$$M = (\mathcal{A}, c, \sigma), \tag{3.4}$$

where $\mathcal{A}$ is a set of *assertions*, c is a *claim* that binds the assertions together, and σ is the *claim signature*.

3.2.1 Assertions

Each assertion $a \in \mathcal{A}$ is a structured statement of fact that the signer is willing to attribute to themselves. The C2PA specification defines around twenty standard assertion types, including:

- Capture device and timestamp,
- Edits performed (cropping, retouching, AI generation),
- Ingredients (parent assets, components, generative inputs),
- Hard bindings (the cryptographic hash of the asset itself),
- **Training and Data Mining** — a flag indicating whether this asset may be used for AI training. This is the assertion that makes the standard relevant to the consent debate.

Assertions are serialised in CBOR (Coalition for Content Provenance and Authenticity Technical Working Group, 2024), a compact binary encoding, and each assertion is itself hashed before being included in the claim.

3.2.2 The claim and its signature

Let $h(a)$ denote the hash of assertion a. The claim c is a structured object containing the list of assertion hashes $\{h(a) : a \in \mathcal{A}\}$ together with metadata identifying the signer's certificate. The claim signature is then

$$\sigma \;=\; \mathrm{Sign}\big(sk_{\mathrm{signer}},\, \mathrm{H}(c)\big). \tag{3.5}$$

A verifier reconstructs c from the manifest, recomputes $\mathrm{H}(c)$, retrieves pk_{signer} from the embedded X.509 certificate, validates the certificate chain, and finally evaluates $\mathrm{Verify}(pk_{\mathrm{signer}}, \mathrm{H}(c), \sigma)$. Any tampering — with the asset, the assertions, or the claim — breaks the verification at exactly the step it was tampered with.

3.3 Hard bindings and soft bindings

A subtle question: what binds the manifest M to the asset A rather than to some other asset? The C2PA specification distinguishes two binding mechanisms.

3.3.1 Hard bindings

A **hard binding** is itself an assertion whose value is the cryptographic hash of the asset's bytes:

$$a_{\mathrm{hard}} \;=\; \mathrm{H}(A). \tag{3.6}$$

Because a_{hard} is included in the claim and the claim is signed (Eq. 3.5), *any modification to A breaks the manifest*. Re-encoding, cropping, recolouring, or compression all change $\mathrm{H}(A)$ and therefore invalidate σ. This is the strongest form of binding the standard offers; it is also the most fragile, in the sense that the signal is deliberately destroyed by edits.

3.3.2 Soft bindings

A **soft binding** is the standard's answer to the practical problem that social-media platforms re-encode every upload, stripping or invalidating hard-bound manifests in transit. The mechanism is a perceptual fingerprint: an invisible watermark or content-hash that survives common transformations and lets a verifier *look up* the original signed manifest in a public repository even when it is no longer attached to the file.

The C2PA specification is deliberately algorithm-agnostic about which watermarking or fingerprinting scheme is used; it standardises only the discovery API (Coalition for Content Provenance and Authenticity Technical Working Group, 2025). The soft-binding layer is the part of C2PA most likely to evolve over the next two years, and it is the part most sensitive to adversarial attacks on the watermark itself.

3.4 What this proves and what it does not

It is essential to be precise about what a verified Content Credential establishes, because the standard is often oversold in popular accounts.

- It establishes that the asset has not been altered since it was signed (hard binding intact).
- It establishes that the signer was the holder of a private key whose corresponding public key is certified by a CA on the C2PA Trust List (signature valid, certificate chain valid).
- It establishes that the assertions in the manifest are exactly those the signer attested to.

It does *not* establish:

- That the assertions are factually correct. A signer can sign false statements; the manifest binds them to the lie, not against it.
- That the absence of a manifest implies an absence of authority. Most assets in the wild have no Content Credential. C2PA is an opt-in standard.
- That a downstream consumer is required to honour the assertions. A scraper that reads a 'Do Not Train' assertion and ignores it has, today, broken no law in most jurisdictions.

The standard establishes **provenance**, not **permission**. It produces a verifiable, tamper-evident record that a creator said *no* — and that record is, in 2026, the closest thing the open web has to an enforceable opt-out signal.

CHAPTER 4

The Toolbox in Math

The previous three chapters establish a problem (scraping is unconstrained), a structural fact (training cannot be undone), and a partial answer (cryptographic provenance). This chapter introduces the active defence layer: three tools that can be deployed by individual creators today. The first, the C2PA Do-Not-Train assertion, is a re-statement of Chapter 3 in operational terms. The second, Glaze, defends against style mimicry by perturbing artwork in feature space. The third, Nightshade, exploits the implosion phenomenon of Chapter 2 to make scraping *actively costly* for a non-cooperating model trainer.

4.1 C2PA's 'Do Not Train': the cryptographic veto

In the language of Chapter 3, the Do-Not-Train signal is a single assertion a_{train} in $\mathcal{A}$ stating that the asset must not be used as training data for AI systems. The cryptographic mechanism is the same as for any other assertion: a_{train} is hashed, included in the claim c, and bound by the claim signature σ.

What distinguishes this assertion from a Terms-of-Service clause is its three-part guarantee: *tamper-evident* (any modification is detectable), *attributable* (signed by an identifiable entity through the certificate chain), and *machine-readable* (a verifier can resolve it deterministically in milliseconds). Its limitation, which we have already named, is that it is voluntary: the standard establishes a verifiable signal, not a sanction for ignoring it.

The corresponding action for a creator, then, is straightforward. Generate a key pair; obtain a certificate from a participating CA (free options now exist for individual creators); attach the assertion to outgoing assets through a compatible editor or dedicated tool. We discuss the user-facing details in the Creator's Toolbox section of the website. *For full derivations and proofs, see Coalition for Content Provenance and Authenticity Technical Working Group (2025).*

4.2 Glaze: optimisation against style mimicry

Glaze (Shan et al., 2023) is a defence against the style-mimicry attack, in which a fine-tuner takes a small number of an artist's pieces and trains a LoRA that reproduces the artist's idiom. Glaze's contribution is an imperceptible perturbation that disrupts the *style features* a mimic would extract while leaving the painting visually unchanged.

4.2.1 Threat model

Let V be a victim artist with a corpus $\{x_i\}$ of artwork. A mimic $\mathcal{M}$:

- Has access to a generic open-source text-to-image model (e.g., Stable Diffusion 1.5),
- Can scrape $\{x_i\}$ from V's public portfolio,
- Fine-tunes the generic model on $\{x_i\}$ to produce a style-specialised model.

The victim's defence is to publish, instead of $\{x_i\}$, a perturbed corpus $\{x_i + \delta_i\}$ such that fine-tuning on the perturbed pieces leads $\mathcal{M}$'s model to reproduce a *different* style.

4.2.2 The cloak optimisation

Let Φ denote a feature extractor used by mainstream text-to-image models (in practice the encoder of the VAE described in Section 2.1, since most mimics use Stable Diffusion). Let $\Omega(x, T)$ denote the output of an off-the-shelf style-transfer system applied to x, with target style T chosen to be far from the artist's true style in Φ-space. The cloak δ is the solution to

$$\min_{\delta} \ \mathrm{Dist}\Big(\Phi(x+\delta),\ \Phi\big(\Omega(x,T)\big)\Big) \quad \text{subject to} \quad \|\delta\|_p \ \leq \ p, \tag{4.1}$$

where $\mathrm{Dist}(\cdot,\cdot)$ is a feature-space distance (typically L_2) and $\|\delta\|_p$ is the perceptual budget that bounds visible distortion. The intuition behind Eq. 4.1 is direct: *move the artwork's representation in feature space toward a foreign style, while keeping pixel-level changes small enough that a human cannot see them*. A mimic who fine-tunes on cloaked pieces will learn the foreign style T rather than V's true style.

In their user study of 1,156 working artists, the Glaze authors found that 88% of respondents wanted to use such a tool, and that fine-tuning on cloaked artwork degraded the mimic's success rate to <12% in human evaluation. *For full derivations and proofs, see Shan et al. (2023).*

4.2.3 Limits

Glaze does not retroactively cloak artwork that has already been scraped. Established artists whose corpora are already in LAION-5B are protected only against future fine-tuning. The cloak is also adversarially fragile in the long run: a future feature extractor sufficiently different from Φ may not be misled by perturbations targeted at Φ. The Glaze authors are explicit about this: the system is a *first step*, deployed while legal and regulatory mechanisms catch up.

4.3 Nightshade: prompt-specific poisoning

Nightshade (Shan et al., 2024) is a more aggressive defence: rather than disrupting a single fine-tuner's run, it injects *poison samples* into the open scraping pool, designed to corrupt the base model's understanding of specific concepts.

4.3.1 The poison construction

Let p be a concept the defender wishes to protect (e.g., the name of an artist, a copyrighted character, a brand). Let t be a target concept dissimilar from p in feature space. Nightshade constructs a poison sample as a pair (x', y_p) where:

- y_p is a natural-language caption containing the concept p (e.g., `"a painting in the style of [artist p]"`),
- x' is an image that visually resembles p to a human observer, but whose feature-space embedding $\Phi(x')$ is close to that of an image of t.

The construction of x' is an optimisation closely related to Eq. 4.1:

$$x' = \underset{x' : \|x' - x\|_p \leq \rho}{\arg\min} \left\| \Phi(x') - \Phi(x_t) \right\|_2^2, \tag{4.2}$$

where x is a real image of concept p, x_t is a real image of concept t, and ρ is again a perceptual budget. The result is a *clean-label* poison: the caption is honest about p, the image looks like p to a human, but the model's internal representation for the pair points toward t.

4.3.2 Why it works

When a base model is trained or fine-tuned on a dataset that includes a small number of such pairs, the cross-attention mechanism (Section 2.3) is supervised to map the token for p toward the visual feature region of t. Because the rest of the dataset still contains genuine (x, y_p) pairs, the supervision is contradictory; per the implosion analysis of Ding et al. (2024), this contradiction degrades alignment for p and bleeds into semantically related concepts.

The empirical results in Shan et al. (2024) show that with as few as 50–300 poison samples per concept, models including Stable Diffusion XL begin generating wrong outputs for the poisoned concepts (e.g., prompting "dog" produces images of cats), and that the effect propagates to related concepts ("puppy", "husky") without explicit poisoning of those terms.

4.3.3 The strategic role of Nightshade

Nightshade is not a personal defence in the way Glaze is; an individual artist's poison samples have negligible effect on a model trained on billions of images. Its role is structural: it changes the cost–benefit calculation for non-cooperating scrapers. As Shan et al. (2024) put it, the existence of a credible poisoning capability gives content owners — studios, agencies, individual creators acting collectively — a disincentive to deploy against trainers who ignore opt-out signals. It is, in the authors' framing, *a copyright-protection tool that does not require the cooperation of the trainer*.

This is also why Nightshade matters for the consent argument even for creators who do not deploy it. Its existence recalibrates the asymmetry that Chapter 1 described: the scraper can no longer assume that ingesting unverified data is free.

For full derivations and proofs, see Shan et al. (2024).

4.4 The combined defence

The three tools sit at three different points in the consent supply chain.

- **C2PA Do-Not-Train** is a *declarative* layer: a verifiable, tamper-evident statement of intent, attached cryptographically to the asset. It works only with cooperating scrapers, but it makes the cooperation auditable.
- **Glaze** is a *personal-defensive* layer: an optimisation that protects an individual artist's style features against the specific threat of fine-tuning-based mimicry. It works without cooperation from the scraper.
- **Nightshade** is a *structural-deterrence* layer: a poisoning attack that, deployed at scale, makes scraping non-cooperating sources actively costly to the model trainer. It works specifically because cooperation has been refused.

None of the three is sufficient alone. C2PA without enforcement is a polite request; Glaze without legal pressure is a treadmill against an adversary with more compute; Nightshade without coordination is a mosquito bite. Used together — a creator who attaches signed Do-Not-Train assertions, cloaks public-facing artwork, and benefits from the existence of poison in the broader pool — they shift the leverage by a measurable amount. That measurable amount is the most honest thing this book can promise.

Afterword: The Ethics of Friction

We will close with a position that has been implicit throughout the technical handbook and which we can now state directly.

The current scraping regime is not the result of a deliberate decision that consent should not apply to AI training. It is the result of friction asymmetries: it is easier to scrape than to ask, easier to fine-tune than to license, easier to settle a class action than to reform a pipeline. None of these asymmetries are inevitable. They are engineering choices stacked on top of legal ambiguities, and engineering choices can be changed by introducing new friction in the right places.

The three layers we have described — declarative provenance, personal cloaking, structural poisoning — are best understood not as tools that solve the consent problem but as tools that *reintroduce friction*. C2PA makes it cost something to ignore a no. Glaze makes it cost something to mimic an unwilling artist. Nightshade makes it cost something to scrape from sources that have refused permission. In each case, the cost is asymmetric: it is paid by the entity that wishes to override the creator's signal, not by the creator. This is the precise inversion of the current regime.

There is a second, harder point worth making. None of these tools settle the underlying question of whether large-scale AI training on publicly visible work *should* be permitted in the first place. That question is political, not technical, and it has a different answer in Brussels than in Bengaluru than in San Francisco. The role of the technical layer is not to settle the question — it is to give the political layer time to deliberate without the data being silently and irrevocably consumed in the meantime. *This*, in our reading, is what cryptographic provenance is for: not to win the argument, but to keep the argument open.

We hope the reader closes this technical handbook with two things: a clearer picture of how a modern image model is built, and a clearer sense of which interventions are honest about their limits. The next time someone tells you that AI scraping is 'inevitable' or that a robots.txt entry is 'enough', you will know exactly which sentences to push back on, and where in the literature the pushback lives.

Open Agency, 2026

Bibliography

Coalition for Content Provenance and Authenticity Technical Working Group. C2PA specification, version 2.1. Technical report, C2PA, 2024. URL https://c2pa.org/specifications/.

Coalition for Content Provenance and Authenticity Technical Working Group. C2PA content credentials explained: Addressing common questions and updates. Technical report, C2PA, September 2025.

D. Cooper, S. Santesson, S. Farrell, S. Boeyen, R. Housley, and W. Polk. Internet X.509 public key infrastructure certificate and CRL profile. Technical Report RFC 5280, IETF, 2008.

Whitfield Diffie and Martin E. Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, 22(6):644–654, 1976.

Wenxin Ding, Cathy Y. Li, Shawn Shan, Ben Y. Zhao, and Haitao Zheng. Understanding implosion in text-to-image generative models. In *ACM Conference on Computer and Communications Security (CCS)*, 2024.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Jonathan Katz and Yehuda Lindell. *Introduction to Modern Cryptography*. CRC Press, 2nd edition, 2014.

Diederik P. Kingma and Max Welling. Auto-encoding variational Bayes. *arXiv preprint arXiv:1312.6114*, 2013.

M. Koster, G. Illyes, H. Zeller, and L. Sassman. Robots exclusion protocol. Technical Report RFC 9309, IETF, 2022.

National Institute of Standards and Technology. Artificial intelligence risk management framework (AI RMF 1.0). Technical Report NIST AI 100-1, U.S. Department of Commerce, 2023.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning (ICML)*, 2021.

Ronald L. Rivest, Adi Shamir, and Leonard Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 21(2):120–126, 1978.

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

Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, Patrick Schramowski, Srivatsa Kundurthy, Katherine Crowson, Ludwig Schmidt, Robert Kaczmarczyk, and Jenia Jitsev. LAION-5B: An open large-scale dataset for training next generation image-text models. In *NeurIPS Datasets and Benchmarks Track*, 2022.

Shawn Shan, Jenna Cryan, Emily Wenger, Haitao Zheng, Rana Hanocka, and Ben Y. Zhao. Glaze: Protecting artists from style mimicry by text-to-image models. In *USENIX Security Symposium*, 2023.

Shawn Shan, Wenxin Ding, Josephine Passananti, Stanley Wu, Haitao Zheng, and Ben Y. Zhao. Nightshade: Prompt-specific poisoning attacks on text-to-image generative models. In *IEEE Symposium on Security and Privacy (S&P)*, 2024.

Jascha Sohl-Dickstein, Eric A. Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning (ICML)*, 2015.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.