Learned Encrypter-Decrypter Models for Diffusion-Like Image Encryption: A Machine-Learning Approach Based on Stochastic Processes

Samuel Cavazos
Alshival.AI

March 28, 2026

`https://github.com/Alshival-Ai/alshicrypt-multimodal`

Abstract

This paper presents a research prototype for learned image encryption based on a fixed stochastic forward process and paired neural endpoint maps. Starting from RGBA sprite images, we corrupt RGB with a diffusion-style process, preserve alpha, and train an *Encrypter* to reproduce the encrypted transit image and a *Decrypter* to reconstruct the original. In the current implementation, RGB is modeled on a discrete 8-bit lattice and alpha is passed through unchanged.

The resulting system trains stably and fits the paired dataset with high accuracy: best dataset MAE is 0.00006691 for the Encrypter and 0.00395170 for the Decrypter. The Mew example below is a real reconstruction from the encrypted input generated by the 1000-epoch Decrypter checkpoint. At the same time, the system is not a cryptosystem. For the default schedule, the surviving pre-clipping signal coefficient after 1000 steps is only $\alpha_T = 5.071e - 12$, and 65.7% of encrypted RGB values lie exactly at 0 or 1, so the stored endpoint is highly destructive after clipping and quantization. The prototype is therefore best understood as a strong research platform for learned reconstruction under severe stochastic corruption, not yet as a formally reversible or cryptographically secure scheme.

Mew (Original) $\xrightarrow{\text{Encrypter}}$

Mew (Encrypted transit image) $\xrightarrow{\text{Decrypter}}$

Mew (1000-epoch Decrypter reconstruction)

Who's That Pokemon?

Mystery A

Mystery B

Mystery C

Three distorted sprites from the dataset are shown above. Try to guess them before checking the answer key at the end of the paper.

1 Introduction

This repository explores whether a machine-learning model pair can act like an Encrypter/Decrypter system for images after a deliberately destructive forward transform. The motivating application is not ordinary image compression; it is the stronger ambition of learning an *encryption-like* transport mechanism in which a sender applies a difficult-to-interpret transformation and a receiver learns the inverse.

One long-term application class is privacy-sensitive image transport. In a future secure version of this framework, a sender-side model could encrypt a sensitive image, such as a radiology scan, pathology slide, or other healthcare image, into a transit representation that is difficult to interpret directly, send that representation over a network, and then use a receiver-side model to decrypt it after arrival. If that workflow were paired with keyed randomness and exact authorized recovery, it could complement conventional transport protections by reducing exposure of directly viewable patient content while images move between institutions or services. In that conceptual pipeline, "encryption in transit" refers to the middle stage, where only the encrypted representation is transmitted between endpoints. The original image remains at the sender, and the decrypted reconstruction appears only after the encrypted payload reaches the receiver. The current repository does not satisfy the security, reversibility, or regulatory requirements for protected health information; here the healthcare example is a motivating application target, not a claim of deployment readiness.

Our early internal experiments focused on deterministic non-linear tensor morphisms: handcrafted non-linear spatial and channel transforms designed to scramble image tensors without explicit randomness. In practice, these morphisms were difficult to generalize. They often behaved like brittle texture-specific warps: they could fit seen patterns, but they did not produce a clean family of transformations with obvious scaling laws or analytically controlled information loss. That experience motivated a shift toward a stochastic process.

The stochastic choice was informed by diffusion and consistency-model literature. Diffusion models are attractive because the forward corruption process is mathematically tractable, the signal-to-noise tradeoff is tunable through a schedule, and a learned network can target endpoint recovery instead of inverting an arbitrary handcrafted morphism [1, 3, 4]. Our prototype adopts that philosophy in a simplified supervised form: generate paired samples with a fixed stochastic process, then learn direct maps for the forward and reverse endpoints.

Contributions. This paper makes four concrete contributions:

1. It formalizes the current stochastic forward transform and derives its closed-form endpoint distribution before clipping.

2. It explains why a stochastic path was chosen over deterministic non-linear tensor morphisms for this line of work.

3. It reports the empirical behavior of the current repository implementation, including quantitative distortion statistics, endpoint training curves, and qualitative reconstructions.

4. It identifies the precise gap between the current prototype and a true cryptographic system.

2 Background and Motivation

2.1 Images as functions and the endomorphism viewpoint

We can formalize the data domain at the pixel level. Let

$$A = \{0, \ldots, 255\}^3 \times [0, 1] \tag{1}$$

denote the implemented pixel attribute space of a single sprite sample: discrete RGB values together with a transparency value. In the current repository, alpha is preserved exactly from input to output, and for the Pokemon sprites it is effectively binary. Let Ω denote a finite coordinate grid. An image can then be viewed as a function

$$i : \Omega \to A, \tag{2}$$

which assigns an RGBA value to each coordinate.

This perspective suggests an algebraic design target for learned transport. Rather than beginning with a full image-to-image black box, we can seek an endomorphism

$$F : A \to A \tag{3}$$

that distorts pixel values while remaining invertible by construction, and then induce an image-level operator coordinatewise:

$$(\mathcal{F}i)(u) = F(i(u)), \qquad u \in \Omega. \tag{4}$$

In the idealized setting, the sender applies $\mathcal{F}$ and the receiver applies $\mathcal{F}^{-1}$, while machine-learning models are trained to approximate F and F^{-1} from data.

The same viewpoint is not limited to images. Once a medium is represented as a function from an index set into a local attribute space, the same endomorphism idea extends naturally to other modalities. For example, audio can be modeled as a function from time indices to amplitude values or short local feature vectors, and an invertible endomorphism on that value space would induce an analogous encrypt/decrypt operator for sound.

This paper studies one stochastic route toward that broader objective. In the current public prototype, RGB semantics live on the discrete 8-bit lattice even though the neural network internally processes rescaled floating-point tensors. The implemented endpoint map is diffusion-like and analytically tractable before clipping and quantization, but the stored PNG pipeline is not exactly invertible. The endomorphism framing is therefore best understood here as the conceptual target for the research program, rather than as a property already achieved by the released implementation.

2.2 Tensor algebra as the broader category-theoretic framing

The image-as-function viewpoint is useful because it exposes the image case cleanly, but the broader research program is better expressed one level up in tensor algebra. In that broader view, image and audio examples are best understood as concrete cases that help us study and understand the framework, not as the boundary of the framework itself. Instead of privileging coordinates first, we can regard a modality sample as a tensor

$$x \in V, \tag{5}$$

where V is a modality-specific tensor space. For the current repository, V is the space of image tensors with discrete 8-bit RGB values and a preserved alpha mask; in future instances, V could instead be an audio waveform space, a spectrogram space, a token-embedding tensor space for text, or another tensorized data domain. The point of the formulation is to generalize across modalities, with image and audio serving as especially intuitive cases for building understanding.

In that formulation, the encryption algorithm is modeled as a tensor morphism

$$T : V \to V. \tag{6}$$

The learned forward and reverse networks are then interpreted as learned tensor maps

$$E : V \to V, \qquad D : V \to V, \tag{7}$$

with the intended relationships

$$E(x) \approx T(x), \qquad D(E(x)) \approx x. \tag{8}$$

This is the category-theoretic version of the endomorphism viewpoint for the encryption algorithm: the same abstract pattern can be instantiated across multiple modalities as long as the data admit a tensor representation.

This tensor-space abstraction clarifies why the current image results matter beyond images themselves. The immediate contribution of the repository is an image-tensor instantiation, but the larger idea is a modality-agnostic encryption/decryption program built from endomorphisms on tensor spaces. In that sense, image and audio should be read as understanding-generating examples of the formalism, whereas the research target is a transferable framework that can be adapted to many kinds of modality.

2.3 Why stochastic corruption is attractive

Let $x_0 \in \{0, \dots, 255\}^{H \times W \times 3} \times [0, 1]^{H \times W \times 1}$ denote an RGBA image with discrete RGB values and a transparency mask. A deterministic tensor morphism seeks a map

$$z = g(x_0), \tag{9}$$

where g is intentionally hard to interpret yet still invertible or approximately invertible. The problem is that once g becomes sufficiently non-linear to hide semantics, it typically becomes difficult to characterize globally. Local Jacobians can vary sharply across the image manifold, and small changes in spatial structure may produce qualitatively different outputs. In our exploratory work this made generalization brittle.

By contrast, a stochastic process defines a family of transforms indexed by noise variables:

$$z = T_\omega(x_0), \tag{10}$$

where ω denotes a random seed or noise path. This has three advantages.

First, the endpoint distribution can often be analyzed exactly or approximately. Second, the process difficulty can be tuned continuously by a schedule rather than redesigned from scratch. Third, the same corrupted endpoint can be paired with direct learned maps, echoing the endpoint-consistency view that motivates one-step generation and denoising in consistency models [4].

2.4 Relation to consistency models

Consistency models learn functions that map points on a noise path toward a shared consistent output, allowing high-quality generation in very few steps [4]. Our method is not a consistency model in the strict sense: we do not train across multiple times with a consistency objective, nor do we solve the probability-flow ODE used in that work. The connection is conceptual rather than identical. The consistency-model literature encouraged us to prefer a stochastic path with analyzable marginals and to investigate whether direct endpoint maps can work without iterative reverse sampling.

3 Method

3.1 Data domain

The current repository uses 809 Pokemon sprite pairs as a controlled RGBA image dataset. Let

$$x_0 = (r_0, g_0, b_0, a_0) \in \{0, \dots, 255\}^{H \times W \times 3} \times [0,1]^{H \times W \times 1}. \tag{11}$$

Here the semantic RGB domain is discrete 8-bit color, while the implementation rescales RGB to floating point internally for convolutional processing. The RGB channels are treated as the signal to corrupt; the alpha channel is preserved exactly in the current implementation rather than predicted by the network.

3.2 Forward process

Define $y_t \in \mathbb{R}^{H \times W \times 3}$ as the RGB state at step t, with $y_0 = (r_0, g_0, b_0)$. Let ε_t denote independent standard Gaussian noise over the flattened RGB coordinates, so $\mathrm{vec}(\varepsilon_t) \sim \mathcal{N}(0, I_{3HW})$. The forward process implemented in the repository is

$$y_t = \sqrt{1 - \beta_t}\, y_{t-1} + \sqrt{\beta_t}\, \varepsilon_t, \tag{12}$$

for $t = 1, \dots, T$, where $T = 1000$ and the schedule is linearly interpolated from $\beta_1 = 0.0005$ to $\beta_T = 0.1$.

The stored encoded image is then

$$z = \big(Q(\mathrm{clip}(y_T, 0, 1)), a_0\big), \tag{13}$$

where clip clamps each RGB value to $[0, 1]$ and Q denotes 8-bit PNG quantization.

Implementation note. For reproducible paired supervision, the current public prototype uses a fixed pseudorandom realization during dataset generation. In other words, the *family* is stochastic, but the released dataset corresponds to one realized corruption path. This is useful for supervised learning and reproducibility, but it is weaker than a deployment setting with keyed per-message randomness.

3.3 Learned endpoint maps

Two separate neural networks are trained:

$$f_{\theta_e} : x_0 \mapsto z, \tag{14}$$

$$f_{\theta_d} : z \mapsto x_0. \tag{15}$$

In repository code and CLI flags these stages are still named 'encoder' and 'decoder' for backward compatibility, but in the paper we refer to f_{θ_e} as the *Encrypter* and f_{θ_d} as the *Decrypter*. Both are instantiated as the same compact U-Net-like CNN with 7,753,408 trainable parameters, built from three downsampling blocks, a bottleneck, and three decoder blocks with skip connections following the U-Net design pattern [2]. The network predicts three separate 256-way categorical distributions, one for each RGB channel, and the output alpha channel is copied from the input image.

Training uses a per-channel categorical objective on RGB:

$$\mathcal{L}_{\text{enc}}(\theta_e) = \frac{1}{3N} \sum_{i=1}^{N} \sum_{c \in \{R,G,B\}} \text{CE}\Big(f_{\theta_e}^{(c)}(x_0^{(i)}), z_c^{(i)} \Big), \tag{16}$$

$$\mathcal{L}_{\text{dec}}(\theta_d) = \frac{1}{3N} \sum_{i=1}^{N} \sum_{c \in \{R,G,B\}} \text{CE}\Big(f_{\theta_d}^{(c)}(z^{(i)}), x_{0,c}^{(i)} \Big). \tag{17}$$

Here CE denotes cross-entropy over the 256 discrete RGB values for one channel. The repository still reports dataset mean absolute error (MAE) after converting predicted RGB logits back to discrete 8-bit values and composing them with the preserved alpha channel.

4 Mathematical Analysis

4.1 Closed-form endpoint distribution

Equation (12) is a discrete-time linear Gaussian process on RGB pixels. Define

$$\alpha_t = \prod_{s=1}^{t} \sqrt{1 - \beta_s}. \tag{18}$$

Proposition 1. *Conditioned on y_0, the pre-clipping endpoint distribution is*

$$\text{vec}(y_t) \mid y_0 \sim \mathcal{N}\big(\alpha_t \, \text{vec}(y_0), \; (1 - \alpha_t^2) I_{3HW} \big). \tag{19}$$

Proof. Unrolling Equation (12) yields

$$y_t = \alpha_t y_0 + \sum_{k=1}^{t} \left(\sqrt{\beta_k} \prod_{s=k+1}^{t} \sqrt{1 - \beta_s} \right) \varepsilon_k. \tag{20}$$

The mean conditioned on y_0 is therefore $\alpha_t y_0$. Since the ε_k are independent standard Gaussians over the flattened RGB coordinates, the covariance is the sum of the squared coefficients:

$$\sum_{k=1}^{t} \beta_k \prod_{s=k+1}^{t} (1 - \beta_s) = 1 - \prod_{s=1}^{t} (1 - \beta_s) = 1 - \alpha_t^2. \tag{21}$$

Thus $\text{vec}(y_t) \mid y_0$ is Gaussian with the claimed mean and covariance. □

4.2 Why the chosen schedule is so destructive

For the default schedule, the final attenuation coefficient is

$$\alpha_T = 5.071e-12, \qquad \alpha_T^2 = 2.571e-23. \tag{22}$$

This means the linear signal energy surviving in RGB at step T is effectively zero before clipping. The forward process therefore behaves almost like pure Gaussian noise at the endpoint, after which clipping and 8-bit quantization collapse many distinct continuous states into the same stored PNG values.

Figure 2 makes this explicit. The linear schedule increases noise steadily, while the surviving signal coefficient decays superlinearly in log scale.

4.3 Implications for invertibility

The current forward operator is not injective once clipping and quantization are applied. Even if two distinct source images $x_0 \neq x_0'$ induce different continuous distributions before quantization, the map in Equation (13) collapses broad regions of RGB space to the same 8-bit values. Exact inversion is therefore impossible in general.

This is the central mathematical distinction between our prototype and a cryptosystem. A cryptosystem can be computationally hard to invert without a key while still being exactly invertible with the key. The present process is instead *information-destroying*. Recovery must therefore come from learned image priors and dataset regularity, not from formal reversibility.

4.4 Role of alpha preservation

The alpha channel is preserved exactly in the dataset generator, so the raw alpha-channel MAE between original and encoded images is 0.000000. This matters for interpretation. A zero-error alpha channel lowers aggregate MAE while leaving RGB corruption unchanged. Consequently, RGB-only MAE is the more informative statistic when discussing semantic recovery.

5 Experimental Setup

The public experiment uses the repository's current configuration:

- Dataset size: 809 paired images.
- Image resolution during training: 120×120 RGBA.
- Forward schedule: 1000 steps, linear β_t from 0.0005 to 0.1.
- Architecture: U-Net-like CNN with 7,753,408 trainable parameters.
- Objective: per-channel cross-entropy on discrete RGB values, with alpha preserved exactly.
- Training logs used here: continued Encrypter and Decrypter runs up to 1000 epochs from the repository's 'models/paper_eval' directory.

These are training-set results, not holdout results. They are useful for understanding whether the endpoint maps can fit the current paired transformation, but they should not be interpreted as evidence of security or strong out-of-distribution generalization.

6 Results

6.1 Forward-process statistics

The encoded RGB endpoint is heavily saturated and nearly decorrelated from the source RGB. Across the current dataset:

- 65.7% of encoded RGB values are exactly 0 or 1 after clipping and quantization.
- The flattened RGB correlation between source and encoded images is -0.0006.
- The alpha channel is unchanged.

This validates the intended behavior of the stochastic transform: the visible endpoint is hard to interpret directly. It also explains why exact inversion is mathematically impossible in the current design.

Metric	Direct baseline	Encrypter	Decrypter
All-channel MAE	0.25615245	0.00006691	0.00395170
RGB-only MAE	0.34153659	0.00008921	0.00526893
RGB PSNR (dB)	5.75	41.80	31.52

Table 1: Dataset-level endpoint metrics on the current paired dataset. All-channel MAE is lower than RGB-only MAE because alpha is preserved exactly by the forward generator.

6.2 Training behavior

Both endpoint networks reach useful error levels quickly, although the discrete-RGB objective is noisier than the earlier continuous-regression baseline. Figure 3 shows steady overall improvement together with the expected epoch-to-epoch oscillations from categorical training on a small dataset. Extending training beyond the initial short run helped materially. The Encrypter best MAE improved from about 0.00628 at 60 epochs to 0.000080 by 200 epochs and 0.0000669 by 1000 epochs. The Decrypter improved even more meaningfully, from about 0.0282 in the earlier short run to 0.0223 by 200 epochs and 0.00395 by 1000 epochs.

6.3 Quantitative endpoint accuracy

Table 1 summarizes the main quantitative results. The direct baseline compares the original image to the encoded image without a learned model. Both learned models improve substantially over that baseline.

6.4 Qualitative behavior

Figure 4 illustrates the current visual behavior. The encrypted targets are visually noisy and semantically obscured. The Encrypter should therefore be judged by how closely it matches that distorted transit target, not by similarity to the original image. The learned Encrypter now tracks the distorted target closely, including much of its clipped color distribution. After the 1000-epoch continuation, the Decrypter also recovers much stronger color fidelity and edge structure than before, and the Mew example in the abstract is now a genuine high-quality reconstruction from the encrypted input rather than a weak toy output.

Figure 5 reinforces the same conclusion. The encoded RGB distribution is sharply concentrated at clipped values, and the learned models reduce error against the direct baseline but do not recover a truly invertible channel.

7 Discussion

7.1 What succeeded

The stochastic reformulation solved several problems that appeared in the earlier deterministic-morphism stage of this project.

First, it gave the experiment a mathematically transparent forward path. Second, it produced a stable supervised learning problem with direct paired targets. Third, it enabled one-step learned endpoint maps that train cleanly with a simple convolutional architecture. These are meaningful successes for a research prototype.

7.2 What failed, or remains unresolved

Three limitations are structural rather than incidental.

The process is not reversible. Because clipping and quantization destroy information, the Decrypter is solving an image-reconstruction problem under a strong prior, not decrypting a formally invertible ciphertext.

The current implementation leaks alpha. Preserving alpha means one quarter of the channels are already solved. This lowers aggregate error and exposes silhouette information directly.

The public prototype fixes a single realized corruption path. That choice is appropriate for reproducible training pairs, but it is not a secure keyed scheme. A production-grade system would need secret per-message randomness or a keyed pseudorandom generator driving the corruption schedule.

The current implementation is limited to a discrete color lattice. The present image pipeline models RGB on the 8-bit lattice $\{0, \ldots, 255\}^3$ and learns three categorical channel distributions rather than a truly continuous color field. That choice matches the stored PNG endpoint and works well for the current sprite dataset, but it is still a limitation of the present system. Extending the framework to continuous color scales would likely require a different output parameterization, such as continuous regression with stronger calibration, mixture-density style heads, discretizations at much finer resolution, or latent continuous color models. It would also likely require substantially more and more varied training data than the current Pokemon sprite corpus, because continuous color recovery is a richer estimation problem than recovering values on a fixed 8-bit lattice.

7.3 Why the stochastic choice was still the right research move

Despite those limitations, the stochastic choice was justified. It gave us an analyzable corruption operator, a well-conditioned training target, and a direct way to study the gap between *learned reconstruction under noise* and *cryptographic reversibility*. Deterministic non-linear tensor morphisms did not provide that level of control or transparency in our exploratory work.

The tensor-algebra framing in Figure 1 also helps explain why this remains a useful research direction despite the present limitations. Even though the current public implementation is only an image-tensor prototype and not yet an invertible cryptosystem, it already serves as a concrete testbed for learned tensor morphisms, learned inverse maps, and modality-general principles that can later be carried into audio, text, and other data domains.

8 Future Work

Several directions follow naturally from the present findings:

1. Replace the fixed realized noise path with keyed per-sample stochasticity.
2. Remove or separately encrypt alpha to avoid silhouette leakage.
3. Study invertible or volume-preserving transforms when exact recoverability is required.

4. Add holdout and cross-domain evaluations to distinguish memorization, generalization, and true prior-based reconstruction.

5. Explore multi-time training objectives inspired more directly by consistency models instead of only endpoint supervision.

6. Investigate continuous-color extensions beyond the current discrete RGB lattice, likely with richer output heads and larger, more diverse training datasets.

9 Conclusion

This project demonstrates that a diffusion-inspired stochastic corruption process is a practical and mathematically legible replacement for brittle deterministic tensor morphisms in early machine-learning-for-encryption research. The resulting endpoint networks do learn the paired transform on the present dataset, and the training behavior is stable. At the same time, the mathematics makes clear why the current system is not yet encryption in the cryptographic sense: the forward process nearly eliminates RGB signal before clipping, the stored endpoint is not injective, and recovery depends on learned priors rather than exact reversibility.

That is still a useful result. The prototype succeeds as a research platform for studying stochastic transport, one-step learned reconstruction, and the boundary between image priors and secure invertible encoding. The next stage should preserve the mathematical clarity of the stochastic framework while tightening the design toward keyed randomness and formally reversible transforms.

Who's That Pokemon? Answer Key

Mystery A: Roselia	Mystery B: Vivillon	Mystery C: Claydol
		
Original	Original	Original
		
1000-epoch Decrypter output	1000-epoch Decrypter output	1000-epoch Decrypter output

References

[1] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, volume 33, pages 6840–6851, 2020. URL `https://proceedings.neurips.cc/paper/2020/hash/4c5bcfec8584af0d967f1ab10179ca4b-Abstract.html`.

[2] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Inter-*

vention – MICCAI 2015, pages 234–241. Springer, 2015. URL https://link.springer.com/chapter/10.1007/978-3-319-24574-4_28.

[3] Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=PxTIG12RRHS.

[4] Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 32211–32252, 2023. URL https://proceedings.mlr.press/v202/song23a.html.

Tensor Algebra for the Encryption Algorithm:
Extension to Image, Audio, Text, and Other Modalities

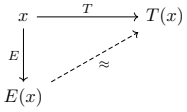
Alshicrypt Multimodal

March 27, 2026

Objects

- V: tensor space for a modality
- $x \in V$: data tensor
- $T : V \to V$: tensor morphism / endomorphism
- $E : V \to V$: learned Encrypter
- $D : V \to V$: learned Decrypter

Tensor-Morphism Picture



This expresses the intended learned relationship

$$E(x) \approx T(x).$$

Learned Reconstruction Picture

$$x \xrightarrow{E} E(x) \xrightarrow{D} D(E(x))$$

with the intended reconstruction relation

$$D(E(x)) \approx x.$$

Interpretation

The object x is a full tensor, and the morphism T acts on that tensor as a whole. This is the tensor-algebra viewpoint: we treat the modality object itself as a tensor in a tensor space V and study endomorphisms

$$T : V \to V.$$

In the learned system, the Encrypter E is trained to approximate the forward tensor morphism, while the Decrypter D is trained to map the transformed tensor back toward the original.

1

Figure 1: Tensor-algebra view of the encryption algorithm. A modality sample is treated as a tensor $x \in V$, the intended encryption transform is a tensor morphism $T : V \to V$, and the learned Encrypter E and Decrypter D are trained to approximate the forward and reverse tensor maps. The current repository instantiates this picture for image tensors; image and audio cases are especially useful for intuition, but the abstraction is meant to generalize to text and other tensor-representable modalities as well.

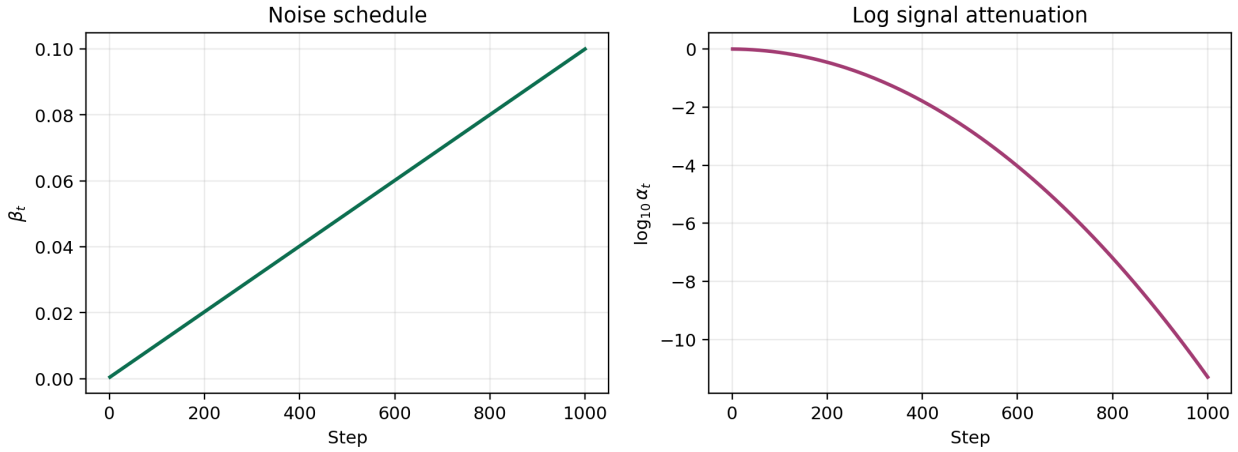


Figure 2: The default corruption schedule used in the repository. Left: the linear β_t schedule. Right: the log attenuation coefficient $\log_{10} \alpha_t$. By the final step, the RGB signal contribution is effectively annihilated.

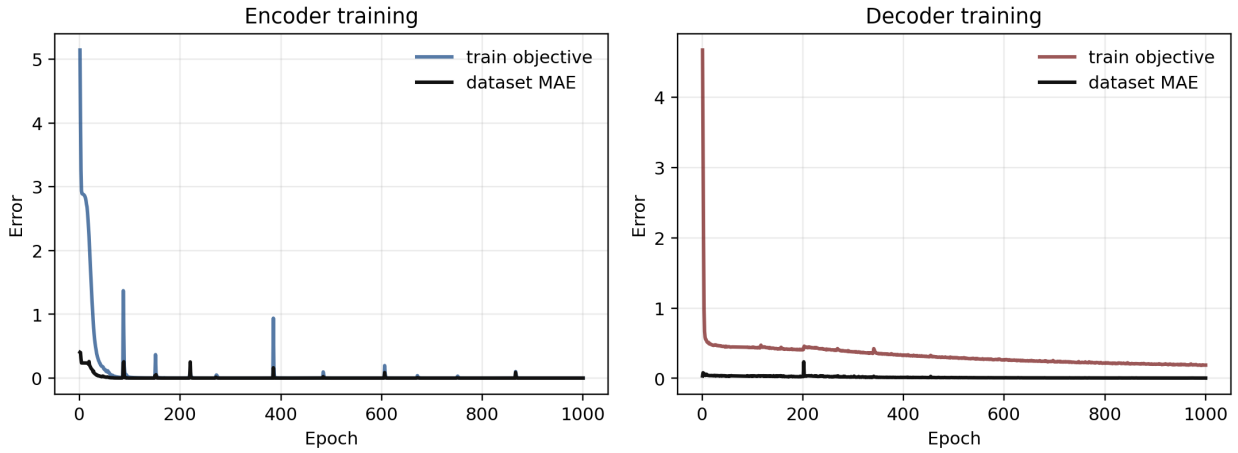


Figure 3: Training dynamics for the Encrypter and Decrypter endpoint networks. Both optimize stably, supporting the claim that the stochastic pairing setup is learnable even with a relatively small convolutional model.

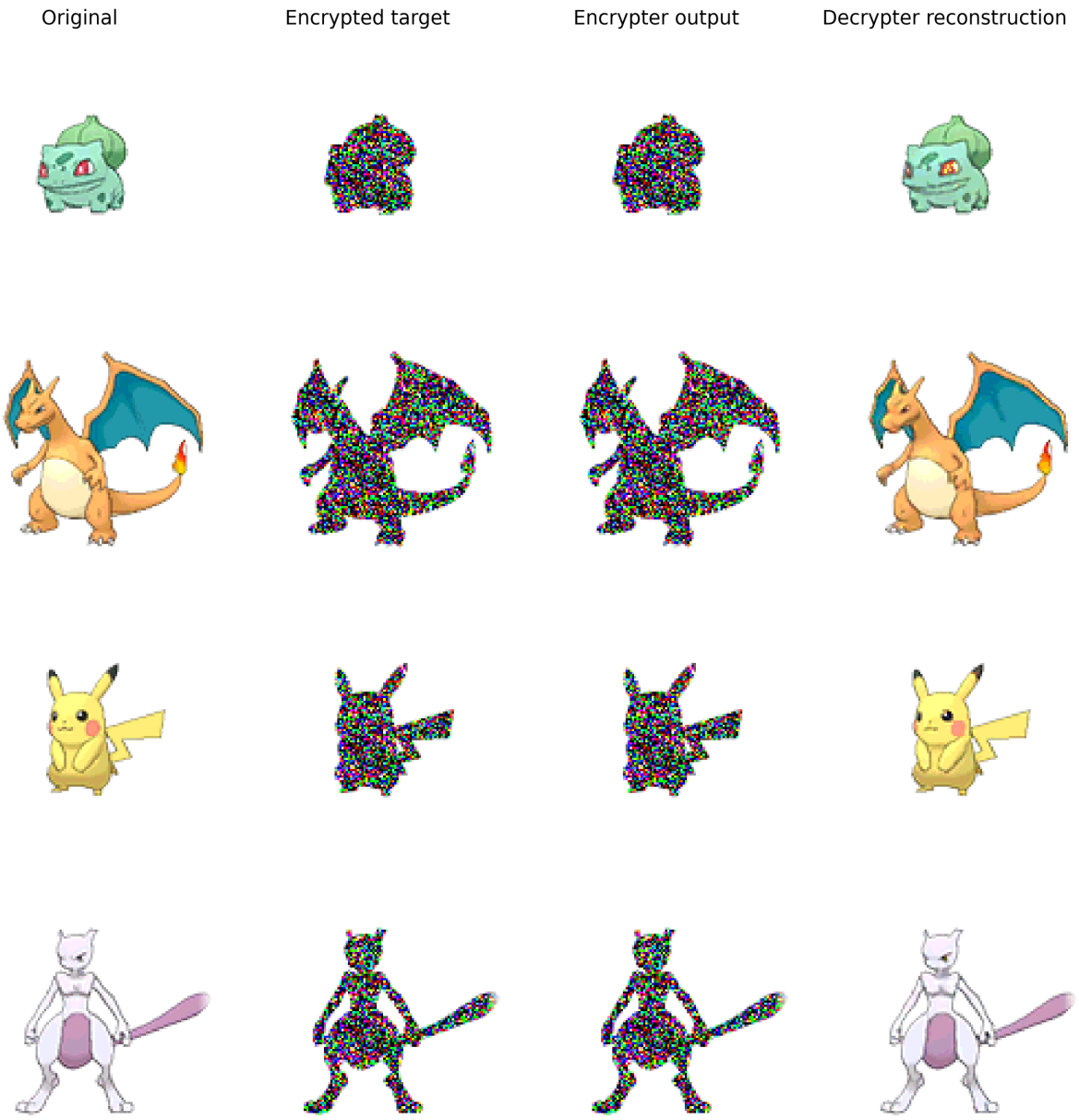


Figure 4: Qualitative examples from the current repository checkpoints: bulbasaur, charizard, pikachu, and mewtwo. Left-to-right within each example, the Encrypter output is supposed to resemble the distorted encrypted target rather than the original image. The final column is the Decrypter reconstruction from that encrypted input. After the longer 1000-epoch training continuation, the paper checkpoints recover geometry, shading, and much more of the original sprite color distribution, although they are still learned reconstructions rather than exact inverses.

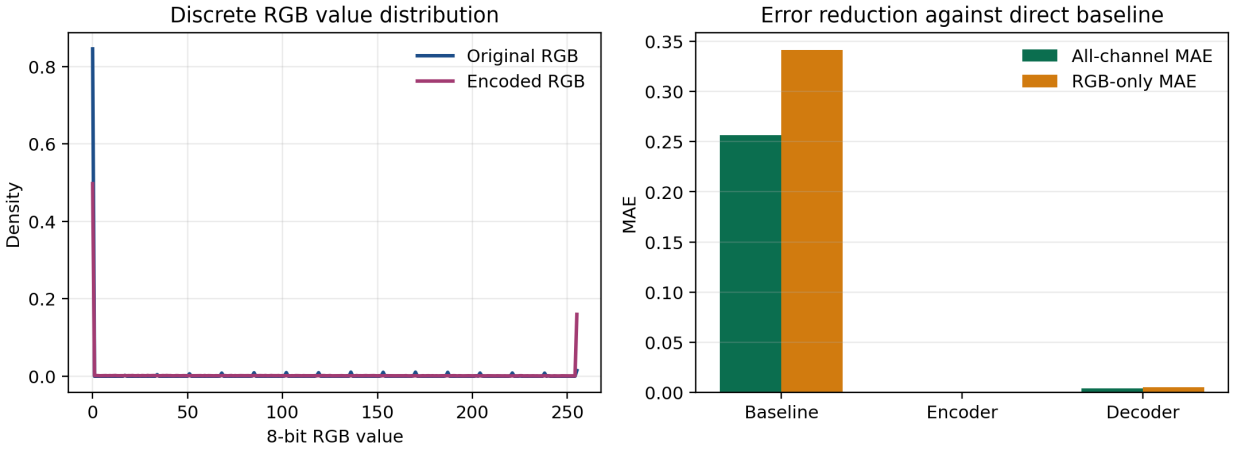


Figure 5: Left: original and encrypted RGB intensity distributions. Right: error reduction relative to the direct baseline. The current Decrypter improves over the baseline substantially, but the result is still learned reconstruction rather than evidence of full reversibility.