WHITE PAPER

Tensorlock: Direct Neural Network Training on Encrypted Datasets

Executive summary · October 2026

Overview

Most high-value data remain unavailable for model development, as data owners seek to protect privacy, commercial value, and intellectual property.

Tensorlock is an encryption method that lets an artificial neural network (ANN) train and perform inference directly on encrypted data. No example is decrypted before any machine-learning operation. Standard network architectures operate on the encrypted vectors with little modification.

This summary describes the core method, its measured performance, and its security analysis, as reported in the initial Tensorlock manuscript.

How the method works

Tensorlock converts each plaintext vector into an encrypted vector of higher dimension. It applies four transformations in sequence:

1. **Padding** adds dimensions to each vector. The added values can be random, sampled from the source data, or derived from the structure of the source data. Padding conceals the original dimension and enlarges the encrypted space.
2. **Perturbation** alters coordinate values within set bounds. It adds noise and preserves the utility of the data for learning.
3. **Fixed index shuffling** applies one secret permutation of coordinates to every example.
4. **Random orthogonal transformation** rotates the vectors into a secret coordinate basis with a random orthogonal matrix.

The same secret transformation applies to all training, test, and inference data in a project.

The orthogonal transformation preserves inner products and distances among the padded vectors. A network needs these geometric relationships to learn. The transformation conceals the original coordinate basis and the positions of the plaintext dimensions.

Training performance

Networks trained on Tensorlock-encrypted data reach accuracy close to that of networks trained on plaintext. The initial Tensorlock manuscript reports the following validation accuracies for standard fully connected networks.

Dataset	Absolute accuracy difference
MNIST	0.0012
Fashion-MNIST	0.0032
CIFAR-10	0.0227
EMNIST Digits	0.0002
EMNIST Letters	0.0028

These results use random padding and perturbation. With advanced padding and perturbation, the differences ranged from 0.0005 to 0.0257.

Training time on MNIST rose by only 1.62% with encrypted data. Reduced storage precision did not lower accuracy: 16-bit ciphertext performed as well as 64-bit ciphertext.

Threat model

Tensorlock operates within a defined threat model. A data owner encrypts a dataset and sends the ciphertext and labels to a second party for training. An

attacker may obtain the encrypted dataset and its labels. The attacker does not hold the plaintext training examples.

At inference, the data owner operates an inference oracle. The oracle accepts a plaintext example, encrypts it privately, runs the trained network, and returns only the predicted label. It never discloses the encrypted vector.

Security analysis

Preserved and hidden information. Tensorlock intentionally preserves the geometry of the padded dataset. It hides the coordinate system. Infinitely many candidate datasets, related by orthogonal transformation, are equally consistent with the observed ciphertext.

Padding. Padding adds a second ambiguity. The attacker must determine which hidden coordinates held the plaintext signal before the orthogonal transformation. The attacker observes the encrypted dimension D and does not know the plaintext dimension.

Inference oracle. The oracle returns labels alone. It therefore supplies no equations that involve the secret transformation.

Coordinate mixing. Each encrypted coordinate combines many plaintext and padding coordinates. The distribution of each encrypted coordinate consequently loses the structure of the plaintext coordinates. It approaches a Gaussian distribution as the encrypted dimension increases.

Modeled covariance attacks

Orthogonal transformation preserves the eigenvalues of the covariance matrix. An attacker with strong reference knowledge of the plaintext covariance could exploit this property. The initial Tensorlock manuscript therefore models attacks in which the attacker holds an exact or near-exact reference dataset.

With an exact reference and a known plaintext dimension, some recovery of examples was possible. Recovery degraded sharply when the reference differed

slightly from the source. With MNIST as the reference for EMNIST Digits data, the paired mean absolute error (MAE) rose from 0.082 to 0.241. That error is similar to the error between unrelated examples of the same class, so no example-specific recovery remained.

Advanced padding and nonlinear perturbation weakened the attack further. In one MNIST experiment with an exact reference and a known plaintext dimension, the paired MAE rose from 0.082 to 0.140. Correctly identified padding coordinates fell from 392 of 392 to 195 of 392.

When the plaintext dimension was unknown, the tested attacks achieved no meaningful recovery against the advanced transformations.

Positioning

Other approaches to private machine learning include federated learning, fully homomorphic encryption, trusted execution environments, differential privacy, and earlier data-encoding methods. Tensorlock differs from these in four practical respects:

- Standard networks train directly on the encrypted data, with minimal changes to architecture.
- Accuracy and training time stay close to those of plaintext training.
- Security parameters are configurable, including the extent and type of padding and perturbation.
- One encryption method is applied to training, test, and inference data.

The last point has commercial importance. A trained network accepts only inputs encrypted with the data owner's key. The data owner therefore controls the use of the network at inference. Encryption protects the privacy of the data and also its economic value.

Limitations

The security analysis in the Tensorlock manuscript applies to a specific threat model and to the attacks tested. Tensorlock does not claim information-theoretic privacy or general ciphertext indistinguishability. The reported experiments in the initial Tensorlock manuscript use fully connected networks on image-classification datasets, but Tensorlock can be adapted to use with other network structures like CNNs, LLMs, and generative networks (see white papers on tensorlock.ai).

Conclusion

Tensorlock preserves the dataset geometry that a neural network needs and conceals the original coordinate representation. Networks trained on encrypted data reach accuracy close to plaintext results, at nearly the same training cost.

Within the stated threat model, the modeled attacks recovered examples only with exact or near-exact knowledge of the source covariance. Tensorlock offers a practical way to use high-value datasets for machine learning without disclosing them in plaintext.