

# Koopcon: A new approach towards smarter and less complex learning

Vahid Jebraeeli

*Electrical and Computer Engineering Department  
North Carolina State University  
Raleigh, USA  
vjebrae@ncsu.edu*

Derya Cansever

*Electrical and Computer Engineering Department  
North Carolina State University  
Raleigh, USA  
dhcansev@ncsu.edu*

Bo Jiang

*Electrical and Computer Engineering Department  
North Carolina State University  
Raleigh, USA  
bjiang8@ncsu.edu*

Hamid Krim

*Electrical and Computer Engineering Department  
North Carolina State University  
Raleigh, USA  
ahk@ncsu.edu*

**Abstract**—In the era of big data, the sheer volume and complexity of datasets pose significant challenges in machine learning, particularly in image processing tasks. This paper introduces an innovative Autoencoder-based Dataset Condensation Model backed by Koopman operator theory that effectively packs large datasets into compact, information-rich representations. Inspired by the predictive coding mechanisms of the human brain, our model leverages a novel approach to encode and reconstruct data, maintaining essential features and label distributions. The condensation process utilizes an autoencoder neural network architecture, coupled with Optimal Transport theory and Wasserstein distance, to minimize the distributional discrepancies between the original and synthesized datasets. We present a two-stage implementation strategy: first, condensing the large dataset into a smaller synthesized subset; second, evaluating the synthesized data by training a classifier and comparing its performance with a classifier trained on an equivalent subset of the original data. Our experimental results demonstrate that the classifiers trained on condensed data exhibit comparable performance to those trained on the original datasets, thus affirming the efficacy of our condensation model. This work not only contributes to the reduction of computational resources but also paves the way for efficient data handling in constrained environments, marking a significant step forward in data-efficient machine learning.<sup>1</sup>

**Index Terms**—Condensation, Autoencoder, Optimal Transport, Wasserstein Distance

## I. INTRODUCTION

The large amounts of data required for training a Deep Learning network for computer vision or imaging problems is increasingly arising as a challenging issue. This paper introduces an innovative method, deeply rooted in the principles of predictive coding [1] observed in the human brain. Predictive coding in the brain is an inspiration to our condensation strategy in that an inference performance difference adapts the new compressed data to the original large data. Our approach aims to emulate the brain's efficiency in processing and exploiting vast quantities of sensory information. By creating compact, predictive representations from complex stimuli, our model parallels the brain's capacity to differentially distinguish

expected and observed data. This enables a significant reduction in the volume of data required for effective training and analysis. Our approach distills large-scale image datasets into condensed, information-rich synthesized counterparts, adeptly balancing the richness of information with computational efficiency. In addressing this data condensation issue, the work of Zhao et al. in [2] was an early contribution data-efficient learning strategies in machine learning. Their approach entails optimizing the synthesized data so that the gradients of a corresponding neural network, when trained on these synthesized datasets, to closely match those obtained from training on original larger datasets. The resulting condensed datasets encapsulate the core characteristics and diversity of extensive data requirements typically associated with such tasks. They demonstrate the effectiveness of their method across various neural network architectures, with condensed datasets that are both compact and informationally dense, thus reducing computational and memory gains.

In another twist on the theme, Zhao et al. [3] developed an efficient method for dataset condensation that synthesizes informative samples whose feature distributions those of original training images in various sampled embedding spaces. This approach significantly reduced synthesis cost and exhibits comparable or superior performance in various settings, including larger datasets and more sophisticated neural architectures. Their method marks a notable advancement by providing a practical solution for dataset condensation foregoing complex bi-level optimization, thereby easing the computational burden in training processes.

Cazenavette et al. in [4] introduce a technique for dataset condensation using the notion of long-range training dynamics by exploiting "expert trajectories" of sequences of model parameters obtained during training on a full dataset. By matching segments of these expert trajectories with those derived from models trained on synthesized data, their approach effectively captures the essential learning dynamics necessary for training the latter. This proposed approach achieved better performance and a promising scaling potential to higher

<sup>1</sup>Thanks to the generous support of ARO grant W911NF-23-2-0041.

resolution data.

We introduce here an Autoencoder-based Model

for condensing large datasets into compact, information-rich representations. Inspired by predictive coding [1] which conjectured to be central to brain cognitive functionality, we employ a combination of Optimal Transport [9] theory and Wasserstein distance [8] to minimize discrepancies between the original and synthesized datasets. We demonstrate the condensation viability of this novel approach, by conceptually projecting the nonlinear data features onto a space with a learned so-called Koopman operator, and subsequently exploiting the latter's properties for a systematic realization.

The paper's structure is as follows: Section II provides background on Koopman Operator Theory [5] and its application in deep learning [6]. Section III introduces the Koopcon model, explaining its formulation, design, and how it integrates elements like autoencoder architecture, self-attention mechanisms [10], and optimal transport theory [9]. Section IV presents our experimental approach, demonstrating the model's effectiveness in maintaining classifier performance with condensed datasets and comparing it against existing methods on standard datasets. The final section, V, concludes the paper, summarizing key findings and outlining potential future directions.

## II. RELATED BACKGROUND

### A. Koopman Operator Theory

Koopman operator theory offers a rich and elegant framework for analyzing nonlinear dynamical systems by transforming them into a linear context. First proposed by Koopman in [5], and further developed in control systems [20], and deep learning applications [6], the theory facilitates the study of complex systems using linear operators on function spaces, regardless of the nonlinearity in the state space.

*Theorem 1 (Koopman Operator Linearity):* Given a nonlinear dynamical system with state evolution defined by  $\vec{x}_{t+1} = f(\vec{x}_t)$ , where  $\vec{x}_t$  (the system state at time  $t$ )  $\in \mathcal{M} \subseteq \mathbb{R}^n$  and  $f: \mathcal{M} \rightarrow \mathcal{M}$ , the Koopman operator  $\mathcal{K}: \mathcal{H} \rightarrow \mathcal{H}$  acts linearly on observable functions  $g: \mathcal{M} \rightarrow \mathbb{R}$  in the Hilbert space  $\mathcal{H}$ , such that:

$$(\mathcal{K}g)(x_t) = g(f(x_t)) = g(x_{t+1}) \quad (1)$$

The theorem emphasizes that the Koopman operator advances observables  $g$  linearly in time according to the system's dynamics. The eigenfunctions  $\psi$  of the Koopman operator satisfy the linear eigenvalue equation  $\mathcal{K}\psi(x) = \lambda\psi(x)$ , with  $\lambda$  as the eigenvalue, indicating a scaled or rotated evolution of the eigenfunction, with its structure preserved over time.

### B. Deep Koopman Operator

Expanding upon Koopman's theory, recent advancements in deep learning [6], [20] have facilitated the approximation of the Koopman operator using neural networks, allowing for practical applications in a variety of complex systems.

*Theorem 2 (Deep Koopman Learning):* Observations  $X = [x_1, x_2, \dots, x_t]$  and their time-evolved states  $X' =$

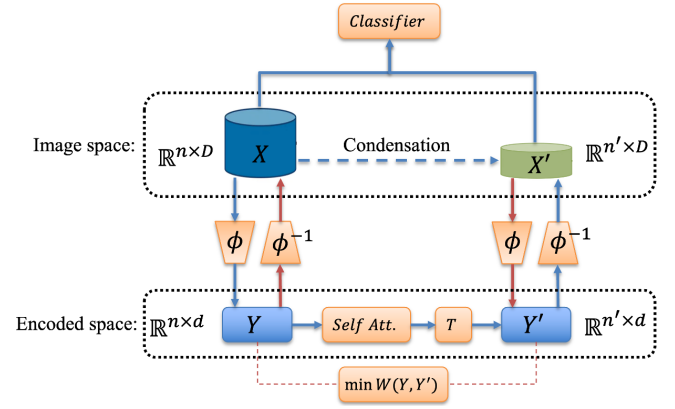


Fig. 1. Koopcon model architecture

$[x_2, x_3, \dots, x_{t+1}]$  of a dynamical system can be utilized to learn a neural network approximation of the Koopman eigenfunctions  $\phi(\cdot)$  and the linear dynamics embodied by a matrix  $T$  in state space  $\mathcal{Y}$ , by minimizing the loss function:

$$\min_{\phi, T} \|\hat{X}' - X'\|^2, \quad (2)$$

where  $\hat{X}'$  are the predicted future states generated by the learned dynamics.

We adopt an autoencoder to capture the K-eigenfunction  $\phi(\cdot)$  and the linear dynamics matrix  $T$ . The encoder maps the input data into a latent space representing the Koopman observables, and the decoder reconstructs the state space from these observables. The linear evolution in the latent space is governed by the learned matrix  $T$ , analogous to the Koopman matrix  $K$ , facilitating the prediction of future system states. As described in [6], DLKoopman provides a bridge between non-linear dynamics and linear predictive models.

### C. Notations

For clarity and consistency throughout this paper, the following notations are adopted, where capital boldface lower case letters respectively denote matrices and vectors, and subscripts denote vector elements while superscripts indicate an alternative copy in a sequence or a transformation.

- $X$ : The original high-dimensional and large-scale dataset, where each element  $x_i$  represents an individual data point with associated features.
- $Y$ : The latent representation of  $X$  obtained after encoding through the autoencoder's encoder network  $\phi$ .
- $X'$ : The condensed dataset synthesized from  $X$ , which is smaller in size but designed to retain the essential information of the original dataset.
- $Y'$ : The condensed latent representation of  $Y$ , which is the result of applying the condensation process within the latent space.
- $\phi$  and  $\phi^{-1}$ : Encoder  $\phi$  maps input data  $X$  to latent representation  $Y$ , and decoder  $\phi^{-1}$  maps  $Y$  back to reconstructed data  $\hat{X}$  or  $X'$ .

- $f_c$ : The classifier function trained on the reconstructed condensed dataset  $X'$ .
- $\mathcal{W}$ : The Wasserstein distance used to measure distributional discrepancies in the condensation process.

#### D. Optimal Transport

Data condensation may be abstractly interpreted as a generalized compression/dimension reduction in that a number of data entities (e.g., images) are compressed into one entity, while methodically aggregating all associated characteristic features. In so doing, we seek to quantitatively track this task by comparing the resulting distributions to that prior, using a measure derived from optimal transport theory. This approach draws upon recent advancements (like in [11] and [12]), integrating principles of optimal transport to enhance the fidelity and efficiency of data condensation, thereby preserving essential information while achieving significant reductions in data volume. Integral to our model is the minimization of the cost  $c$  for transforming the encoded latent representation of original data with a probability density function  $p(Y)$  to closely match that of condensed version of data  $p(Y')$ . This is articulated through the Optimal Transport Loss:

$$\mathcal{L}_{O.T.} = \min_{\pi \in \Pi(p(Y), p(Y'))} \mathbb{E}_{(Y, Y') \sim \pi} [c(Y, Y')] \quad (3)$$

Here,  $\pi$  represents a coupling between the distributions  $p(Y)$  and  $p(Y')$ , with  $c(Y, Y')$  denoting the dissimilarity measure between  $Y$  and  $Y'$ .

#### E. Problem Formulation

The dataset condensation problem involves transforming a large-scale training set  $X$  into a smaller synthetic set  $X'$ . Formally,  $X = \{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_n, y_n)\}$  comprises  $n$  image and label pairs. The condensed set  $X' = \{(\mathbf{x}'_1, y_1), \dots, (\mathbf{x}'_{n'}, y_{n'})\}$  yields  $n'$  synthetic image and label pairs. The principle objective is to seek CNN-parameterized models of  $X$  and  $X'$  which similarly perform on unseen testing data, analytically stated as:

$$\mathbb{E}_{\mathbf{x} \sim P_X} [\ell(\Psi_{\theta^X}(\mathbf{x}), y)] \simeq \mathbb{E}_{\mathbf{x} \sim P_{X'}} [\ell(\Psi_{\theta^{X'}}(\mathbf{x}), y)],$$

where  $P_X$  represents the real data distribution,  $\ell$  the loss function (e.g., cross-entropy), and  $\Psi$  a deep neural network parameterized by  $\theta$ .

### III. METHODOLOGY

The driving intuition behind our proposed solution to data Koopman-condensation (Koopcon) lies in the projection of characteristic features  $Y$  in data at hand, and in their systematic, class-consistent and distributionally-balanced packing  $Y'$  prior to reconstruction ( $X'$ ).

**Claim:** A near-optimal and inference-driven dimension-reduction  $Y'$  of a data-set  $X$ , can be achieved by an optimal transport in a Koopman-data space.

As depicted in Figure 1, is engineered to condense a voluminous, high-dimensional dataset  $X$  into the compact yet informatively dense set  $X'$  is crafted to retain the crucial

attributes of the original dataset exploiting the Koopman linear evolution of non-linear dynamics.

The latent representation  $Y \in \mathbb{R}^{n \times d}$  is the result of  $\phi : \mathbb{R}^{n \times D} \rightarrow \mathbb{R}^{n \times d}$  with  $d < D$ , and is followed by a linear transformation to capture the intrinsic data dynamics.

**Reconstruction Loss ( $\mathcal{L}_{re}$ ):** The auto-encoder parametrization denoted by  $\phi$  and  $\theta$  for the encoder and decoder respectively follows the standard optimization loss between the input distribution and that of the output written as

$$\mathcal{L}_{re}(\phi, \theta; X) = \mathbb{E}_{Y \sim q_\phi(Y|X)} [-\log p_\theta(X|Y)] + \text{KL}[q_\phi(Y|X) || p(Y)], \quad (4)$$

where the first term is the expected negative log-likelihood, and the second term is the Kullback-Leibler divergence between the encoded distribution  $q_\phi(Y|X)$  and a prior distribution  $p(Y)$ . To selectively exploit the most relevant features and further refine the intrinsic linear evolution of the nonlinear dynamics, we precede the linear transformation  $T(\cdot)$  with a self-attention transformation on  $Y$  to yield  $Y' \in \mathbb{R}^{n' \times d}$ .

In the condensation phase,  $Y'$  crafted to be of lower dimension ( $n'$ ) while still reflecting the original dataset's distribution. This process is guided by the minimization of the Wasserstein distance  $\mathcal{W}(Y, Y')$ , ensuring that the condensed data  $Y'$  maintains the distributional integrity of  $Y$ .

**Wasserstein Distance ( $\mathcal{L}_W$ ):** The concept of Wasserstein distance [8] arises as a specialized form of the Optimal Transport Loss [9], where it specifically measures the cost to align the distribution of the encoded data  $Y$  with that of the condensed representation  $Y'$ . The Wasserstein distance, therefore, quantifies the minimal "effort" required to morph the distribution  $p_Y$  into  $p_{Y'}$ , making it a natural measure for the effectiveness of dataset condensation processes.

The Wasserstein distance can be expressed as:

$$\mathcal{L}_W(p_Y, p_{Y'}) = \min_{\pi \in \Pi(p_Y, p_{Y'})} \iint c(Y, Y') \pi(p_Y, p_{Y'}) dY dY' \quad (5)$$

In this formulation,  $\pi$  corresponds to the optimal transport that associates the distributions  $p(Y)$  and  $p(Y')$ . By minimizing the Wasserstein distance  $\mathcal{L}_W$ , we aim to ensure that the condensed dataset  $Y'$  not only statistically resembles the original dataset  $Y$  but also preserves its geometric and topological properties, crucial for maintaining the fidelity of the condensed data for subsequent learning tasks that depend on the intricate relationships within the data's manifold structure.

The condensed representation  $Y'$  is subsequently mapped back into the high-dimensional image space using the same decoder function  $\phi^{-1}$  that was initially used for encoding. This results in the condensed dataset  $X'$ , where  $X' \in \mathbb{R}^{n' \times D}$ . The utilization of the same autoencoder for both encoding and decoding stages ensures that the condensed data  $X'$  is a plausible output of the autoencoder, retaining the structure and distributional properties of the original dataset.

A classifier  $f_c$  is then trained on the reconstructed condensed dataset  $X'$ , which is equipped to predict the output labels  $\hat{y}$  as if it were trained on the original dataset  $X$ . This

process allows the classifier to benefit from the distilled information within  $X'$ , enabling efficient training with significantly reduced computational resources.

**Classification Loss ( $\mathcal{L}_{ce}$ ):** Central to our model is the classification loss, which serves as a form of implicit feedback information. It evaluates the discrepancy between the predicted labels obtained from the classifier and the true labels, guiding the latent representation towards maintaining label consistency. Crucially, this process involves both the original and synthesized condensed images ( $X$  and  $X'$ ), which are merged and passed through the classifier to ensure comprehensive learning. The Cross-Entropy (CE) loss metric is utilized for this purpose:

$$\mathcal{L}_{ce}(f_c, \tilde{X}, y) = - \sum_i y_i \log(f_c(\tilde{X}_i)) \quad (6)$$

Here,  $f_c$  represents the classifier function,  $\tilde{X}$  is the combined set of original and reconstructed data,  $y$  is the vector of true labels, and  $\tilde{X}_i$  refers to the  $i$ -th data instance in the merged dataset. This loss component is instrumental in ensuring that the condensed dataset encapsulates not only the structural attributes of the original data but also its label characteristics, thus preserving essential discriminative features and preventing the dilution of categorical information during the dataset condensation process.

The Koopcon model leverages the computational efficiency of linear dynamics in the encoded space and the cognitive economy of the brain's predictive coding strategy [1]. It presents a significant advancement in creating data-efficient learning strategies, allowing for scalable training on extensive datasets while maintaining performance parity with models trained on the full dataset.

**Covariance Loss ( $\mathcal{L}_{cov}$ ):** To foster a more diverse and representative condensed dataset, we introduce a covariance loss term into the overall loss function. This term serves as a regularizer, promoting the capture of distinct features within the latent representations  $Y$ . When examining the encoded versions of all  $Y_i$ s against the encoded version of representative  $Y'_i$ s, the necessity for such a loss term becomes apparent. In scenarios without the covariance loss, the representatives tend to cluster together, leading to a less diversified representation. Conversely, the inclusion of covariance loss encourages a more scattered distribution of representatives  $Y'_i$ s, thereby enhancing the diversity within the dataset.

The mathematical definition of Covariance Loss is given by:

$$\mathcal{L}_{cov}(Y') = \|\text{Cov}(Y') - I\|_F^2, \quad (7)$$

where  $\text{Cov}(Y')$  denotes the covariance matrix of the latent representation  $Y'$ ,  $I$  is the identity matrix, and  $\|\cdot\|_F$  represents the Frobenius norm. By minimizing  $\mathcal{L}_{cov}$ , the model is encouraged to produce features that are uncorrelated, thereby increasing the informativeness and variability of the synthesized samples. This discourages feature redundancy, which is instrumental in avoiding overfitting and improving the model's ability to generalize from synthesized representatives to unseen

data. Thus, the Covariance Loss plays a pivotal role in ensuring that the condensed dataset is not only a compressed version of the original data but also a functionally diverse subset that retains the original's rich feature set.

This yields a weighted sum of these losses including a covariate spread constraint  $\mathcal{L}_{cov}$  to ensure that the synthesized samples are diverse and persistent for a good cover of the distribution, as:

$$\mathcal{L}_{total} = \alpha_0 \mathcal{L}_{re} + \alpha_1 \mathcal{L}_{ce} + \alpha_2 \mathcal{L}_{\mathcal{W}} + \alpha_3 \mathcal{L}_{cov} \quad (8)$$

where  $\alpha_0, \alpha_1, \alpha_2$ , and  $\alpha_3$  are hyperparameters that balance the different components of the loss function.

In summary, the model synthesizes a dataset  $X'$  that, when used to train a machine learning model, aims to achieve performance comparable to using the original dataset  $X$ . The losses guide the model to learn a latent representation that is both accurate to the original data and informative for the learning task. Find detailed algorithms for train and test of our model in algorithms 1 and 2 respectively.

---

#### Algorithm 1 Koopcon Training Algorithm

---

```

1: Given:
2:  $X \in \mathbb{R}^{n \times D}$ , original dataset with  $n$  samples
3:  $\phi$ : Encoder mapping  $\mathbb{R}^D \rightarrow \mathbb{R}^d$ 
4:  $\phi^{-1}$ : Decoder mapping  $\mathbb{R}^d \rightarrow \mathbb{R}^D$ 
5:  $n'$ : Target number of synthesized samples
6:  $\alpha_0, \alpha_1, \alpha_2, \alpha_3$ : Weights for loss components
7:  $N$ : Number of training epochs
8:  $M$ : Number of classes of data
9: Initialize: Parameters of Autoencoder ( $\phi, \phi^{-1}$ ), Classifier  $f_c$ , Linear transformation  $T$ , Self attention  $SA$ 
10: for epoch = 1 to  $N$  do
11:   for class = 1 to  $M$  do
12:      $Y \leftarrow \phi(X)$ 
13:      $Y' \leftarrow T(SA(Y))$ 
14:      $X' \leftarrow \phi^{-1}(Y')$ 
15:      $L_{re} \leftarrow \|X' - X\|^2$ 
16:      $\hat{Y} \leftarrow f_c(X \oplus X'), (\oplus: \text{concatenation})$ 
17:      $L_{ce} \leftarrow - \sum_i y_i \cdot \log(\hat{Y}), (y_i: \text{vector of true labels})$ 
18:      $\mathcal{L}_{\mathcal{W}} \leftarrow \mathcal{W}(Y, Y'), (\mathcal{W}: \text{Wasserstein Distance})$ 
19:      $\mathcal{L}_{cov}(Y') \leftarrow \|\text{Cov}(Y') - I\|_F^2$ 
20:      $L_{total} \leftarrow \alpha_0 L_{re} + \alpha_1 L_{ce} + \alpha_2 \mathcal{L}_{\mathcal{W}} + \alpha_3 \mathcal{L}_{cov}$ 
21:     Update Parameters

```

---

## IV. EXPERIMENTS AND RESULTS

### A. Stages of Implementation

The stages of implementation are illustrated in Figure 2, which outlines the two-phase process of dataset condensation and subsequent evaluation.

#### 1) First Stage (Condensation)

**I. Input (Real Big Dataset):** We begin with a large dataset  $X$ , consisting of pairs  $(x_i, y_i)$  where  $x_i$  represents the features (e.g., images) and  $y_i$  the corresponding labels. The dataset has  $n$  such pairs, and labels range over  $C$  different classes, from 0 to  $C - 1$ .

## Algorithm 2 Koopcon Testing Algorithm

```

1: Given:
2:  $X_{\text{train}}, X_{\text{test}}$ , real train and test data
3:  $\phi$ : Encoder mapping  $\mathbb{R}^D \rightarrow \mathbb{R}^d$ 
4:  $\phi^{-1}$ : Decoder mapping  $\mathbb{R}^d \rightarrow \mathbb{R}^D$ 
5: Linear transformation  $T$  and Self-Attention  $SA$ 
6: Classifiers  $f_c$ 
7:  $N$ : Number of training epochs
8:  $M$ : Number of classes
9: Initialize: Load trained parameters for Autoencoder ( $\phi, \phi^{-1}$ ), Linear Transformation  $T$ , and Self-Attention  $SA$ 
10: for epoch = 1 to  $N$  do
11:   for class = 1 to  $M$  do
12:      $Y \leftarrow \phi(X_{\text{train}})$ 
13:      $Y' \leftarrow T(SA(Y))$ 
14:      $X' \leftarrow \phi^{-1}(Y')$ 
15:      $f_c^{\text{synth}}$ : Train classifier  $f_c$  on  $X'$ 
16:      $f_c^{\text{real}}$ : Train classifier  $f_c$  on  $X_{\text{train}}$ 
17: Evaluate: Test  $f_c^{\text{synth}}$  and  $f_c^{\text{real}}$  on  $X_{\text{test}}$ 
18: Compare Performance: Calculate and report accuracy for both classifiers

```

**II. Dataset Condensation Process:** This large dataset  $X$  undergoes a condensation process to produce a much smaller, synthesized dataset  $X'$ . This condensed dataset contains pairs  $(x'_i, y_i)$ , where  $x'_i$  are the synthesized features (condensed representations) and  $y_i$  are the corresponding labels. There are  $n'$  pairs in  $X'$ , and it maintains the same range of labels as the original dataset.

### 2) Second Stage (Evaluation)

**I. Training with Synthetic Data (Small Dataset):** The synthesized dataset  $X'$  is then used to train a classifier. The classifier learns to predict labels based on the condensed feature set provided by  $X'$ .

**II. Training with Real Data (Big Dataset):** In parallel, we train the same type of classifier on a subset of the large dataset  $X$ . This subset is selected to have the same number of examples  $n'$  as the  $X'$  to make a fair comparison.

**III. Comparison of Test Performance:** After both classifiers are trained, their performance is evaluated on a test set. The goal is to demonstrate that the classifier trained on the synthesized dataset  $X'$  performs similarly to that of the real dataset  $X$ , despite  $X'$  being significantly smaller in size.

The underlying hypothesis is that if the synthesized dataset  $X'$  is a good condensation of  $X$ , then the classifier trained on  $X'$  should generalize almost as well as the classifier trained on  $X$  when evaluated on unseen data. This would show that  $X'$  successfully captures the core information from the larger dataset  $X$ , enabling effective training with much less data.

## B. Results

The tables provide a comparative overview of classification accuracies achieved by various dataset condensation models across standard datasets CIFAR10 [13], FashionMNIST [14], and MNIST [15]. Each model's performance is evaluated

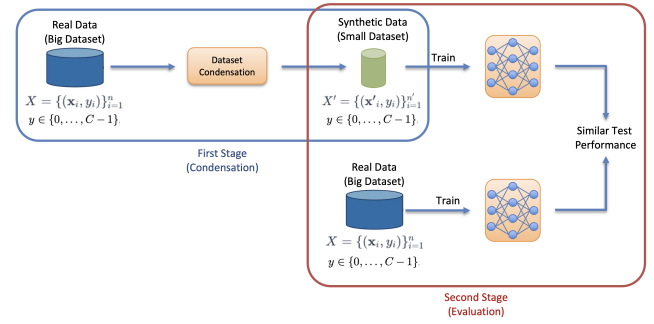


Fig. 2. Stages of Implementation and Evaluation of a Condensation Model, from initial large dataset to synthetic dataset generation, and subsequent training and performance comparison.

based on the number of images per class used during training, ranging from a single image to fifty images. The reported accuracies are the mean values derived from 10 separate experiments, with the upper and lower accuracy bounds presented within the tables for each dataset. In all experiments, we are starting with 5000 Img/Cls and ConvNet is used as a classifier.

**Table I** outlines the accuracy on the MNIST dataset. It compares our proposed model against several other models, including Gradient Matching (GM) [2], Distribution Matching (DM) [3], Matching Training Trajectory (MTT) [4], Kernel Inducing Points (KIP) [7], and the results from using the entire dataset for training. The accuracy of the proposed model consistently increases with the number of images per class, suggesting that it can effectively utilize additional data. Notably, the proposed model outperforms the other methods in each category, inching closer to the full dataset's performance as the number of images increases. The "Img/Cls" columns show the target number of compressed data per class. Starting with 5000 Img/Cls and compressing to 10 Img/Cls for MNIST, for instance in 2nd row.

**Table II** shows the accuracies on the FashionMNIST and CIFAR10 datasets. It follows a similar pattern to Table I, where the proposed model generally surpasses the competing methods across varying images per class. The proposed model's performance on both FashionMNIST and CIFAR10 datasets shows a significant improvement over other methods, especially as the number of images per class grows.

TABLE I  
CLASSIFICATION ACCURACY ON MNIST DATASET

Img/Cls	GM [2]	DM [3]	MTT [4]	KIP [7]	Ours	Whole Dataset
1	91.7±0.5	89.7±0.6	91.4±0.9	90.1±0.1	<b>95.5±0.5</b>	
10	97.4±0.2	97.5±0.1	97.3±0.1	97.5±0.0	<b>98.2±0.1</b>	99.6±0.0
50	98.8±0.2	98.6±0.1	98.5±0.1	98.3±0.1	<b>99.4±0.0</b>	

**Table III** showcases a comparison of generalizability for various dataset condensation models by evaluating their performance across different architectures. The classification accuracies are reported for each condensation model when used to

TABLE II  
CLASSIFICATION ACCURACY ON FASHIONMNIST AND CIFAR10 DATASETS

Img/Cls	FashionMNIST						CIFAR10					
	GM [2]	DM [3]	MTT [4]	KIP [7]	Ours	Whole Dataset	GM [2]	DM [3]	MTT [4]	KIP [7]	Ours	Whole Dataset
1	70.5±0.6	71.5±0.5	75.3±0.9	73.5±0.5	<b>76.0±0.9</b>		28.3±0.5	26.0±0.8	46.3±0.8	49.9±0.2	<b>51.4±0.7</b>	
10	82.3±0.4	83.6±0.2	87.2±0.3	86.8±0.1	<b>87.5±0.4</b>	93.5±0.1	44.9±0.5	48.9±0.6	65.3±0.7	62.7±0.3	<b>67.7±0.5</b>	84.8±0.1
50	83.6±0.4	88.2±0.1	88.3±0.1	88.0±0.1	<b>88.9±0.0</b>		53.9±0.5	63.0±0.4	71.6±0.2	68.6±0.2	<b>73.2±0.3</b>	

train four distinct classifying networks: ConvNet [16], ResNet-18 [17], VGG-11 [18], and AlexNet [19].

The comparison reveals how well each condensation approach can adapt to different architectures, which is indicative of its ability to capture the essential features of the original dataset and generalize from it. Notably, the proposed model demonstrates competitive accuracy across all classifying networks, suggesting that it produces a synthetic dataset that effectively generalizes and maintains the integrity of the data's underlying structure, regardless of the classifier used. This trait is particularly valuable in machine learning, where the ability to perform well across various architectures is a hallmark of a robust condensation technique. The structural variations of autoencoder architectures employed in our study include Shallow, Medium, and Deep. Each architecture represents a different level of model complexity. The Shallow autoencoder consists of 5 convolutional layers, ascending to the Medium autoencoder with 7 convolutional layers, and peaking with the Deep architecture, which comprises 9 convolutional layers. These designs are purposefully crafted to evaluate the impact of depth on the model's ability to encode and reconstruct image data, with the hypothesis that deeper networks may capture more abstract features but could also be prone to overfitting.

TABLE III  
COMPARATIVE GENERALIZATION PERFORMANCE OF CONDENSATION MODELS ACROSS DIFFERENT CLASSIFIERS

Condensation Architecture	Classifying Network			
	ConvNet [16]	ResNet [17]	VGG [18]	AlexNet [19]
GM [2]	53.2±0.8	42.1±0.7	46.3±1.3	34.0±2.3
DM [3]	49.2±0.8	36.8±1.2	41.2±1.8	34.9±1.1
MTT [4]	<b>64.4±0.9</b>	49.2±1.1	46.6±2.0	34.2±2.6
KIP [7]	62.7±0.3	49.0±1.2	30.1±1.5	57.2±0.4
Ours	<b>65.0±0.5</b>	<b>60.7±1.0</b>	<b>59.5±1.5</b>	<b>61.5±0.9</b>

Figure 3 presents the empirical results of our experiments, graphing the test accuracy achieved by each autoencoder architecture on the different dataset. The x-axis plots the number of images per class (Img/Cls) used during training, serving as a measure of dataset size and richness. The y-axis quantifies the test accuracy, providing a clear performance metric for each model. The lines for each architecture variant—Shallow, Medium, and Deep—converge towards the accuracy obtained when the entire dataset is leveraged for training, depicted by the 'Whole Dataset' line. These results are instrumental in understanding how the depth of an autoencoder affects its capacity for dataset condensation and subsequent classification



Fig. 3. Test accuracy comparison between our model with different depth of Autoencoder, GM and DM Architectures in Different datasets

performance, with the aim to optimize the trade-off between model complexity and generalizability.

Figures 3 highlights the balance between autoencoder complexity and the effectiveness of our dataset condensation model. It suggests that increasing the autoencoder's depth enhances feature capture but also indicate that more complexity doesn't always yield better generalization. Our findings underscore the importance of optimizing autoencoder depth to produce a synthesized dataset that is both compact and representative of the original, striking a crucial balance for efficient model training.

## V. CONCLUSION AND FUTURE WORKS

In this paper, we presented Koopcon, an innovative Autoencoder-based Dataset Condensation Model, adept at transforming large datasets into smaller, information-rich counterparts. Using an OT-theoretic criterion, we effectively secured a minimization of Wasserstein distance in Koopman space between the distributions of the original and that of condensed data. Through extensive experiments, we demonstrated that classifiers trained on these condensed datasets achieve performance comparable to those trained on the entire dataset, thus significantly reducing computational resources without compromising data integrity. Koopcon marks a pivotal advancement in data-efficient machine learning, offering feature-driven solutions in constrained environments.

Our future work will explore expansive synthesis as a complement to dataset condensation. This approach will start with a condensed but information-rich dataset and aim to expand it, generating a more diverse and informative dataset. The goal is to enhance the small dataset systematically while preserving its essential characteristics and information content. Such an expanded dataset could improve the training of machine learning models by providing a richer set of data points, potentially enhancing model robustness and generalization while managing computational costs effectively.



## REFERENCES

- [1] Friston, Karl, and Stefan Kiebel. "Predictive coding under the free-energy principle." *Philosophical transactions of the Royal Society B: Biological sciences* 364.1521 (2009): 1211-1221.
- [2] Zhao, Bo, Konda Reddy Mopuri, and Hakan Bilen. "Dataset condensation with gradient matching." *arXiv preprint arXiv:2006.05929* (2020).
- [3] Zhao, Bo, and Hakan Bilen. "Dataset condensation with distribution matching." *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. (2023).
- [4] Cazenavette, George, et al. "Dataset distillation by matching training trajectories." *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. (2022).
- [5] Koopman BO. *Hamiltonian Systems and Transformation in Hilbert Space*. *Proc Natl Acad Sci U S A*. 1931 May;17(5):315-8.
- [6] Dey, Sourya, and Eric William Davis. "DLKoopman: A deep learning software package for Koopman theory." *Learning for Dynamics and Control Conference*. PMLR, (2023).
- [7] Nguyen, Timothy, et al. "Dataset distillation with infinitely wide convolutional networks." *Advances in Neural Information Processing Systems* 34 (2021): 5186-5198.
- [8] Vaserstein, Leonid Nisonovich. "Markov processes over denumerable products of spaces, describing large systems of automata." *Problemy Peredachi Informatsii* 5.3 (1969): 64-72.
- [9] Villani, Cédric. *Optimal transport: old and new*. Vol. 338. Berlin: springer, 2009.
- [10] Vaswani, Ashish, et al. "Attention is all you need." *Advances in neural information processing systems* 30 (2017).
- [11] S. Roheda, A. Panahi and H. Krim, "Fast Optimal Transport for Latent Domain Adaptation," 2023 IEEE International Conference on Image Processing (ICIP), Kuala Lumpur, Malaysia, (2023): 1810-1814
- [12] B. Jiang, H. Krim, T. Wu and D. Cansever, "Implicit Bayes Adaptation: A Collaborative Transport Approach," ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Rhodes Island, Greece, (2023): 1-5
- [13] Krizhevsky, Alex, and Geoffrey Hinton. "Learning multiple layers of features from tiny images." (2009): 7.
- [14] Xiao, Han, Kashif Rasul, and Roland Vollgraf. "Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms." *arXiv preprint arXiv:1708.07747* (2017).
- [15] Deng, Li. "The mnist database of handwritten digit images for machine learning research [best of the web]." *IEEE signal processing magazine* 29.6 (2012): 141-142.
- [16] Gidaris, Spyros, and Nikos Komodakis. "Dynamic few-shot visual learning without forgetting." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018.
- [17] He, Kaiming, et al. "Deep residual learning for image recognition." *Proceedings of the IEEE conference on computer vision and pattern recognition*. (2016).
- [18] Simonyan, Karen, and Andrew Zisserman. "Very deep convolutional networks for large-scale image recognition." *arXiv preprint arXiv:1409.1556* (2014).
- [19] Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. "Imagenet classification with deep convolutional neural networks." *Advances in neural information processing systems* 25 (2012).
- [20] Kaiser, Eurika, J. Nathan Kutz, and Steven L. Brunton. "Data-driven discovery of Koopman eigenfunctions for control." *Machine Learning: Science and Technology* 2.3 (2021): 035023.