Generative Expansion of Small Datasets: An Expansive Graph Approach

Vahid Jebraeeli
ECE Department
NC State University
Raleigh, USA
vjebrae@ncsu.edu

Bo Jiang
Neuroscience Department
Washington University
St. Louis, USA
bo.j@wustl.edu

Hamid Krim
ECE Department
NC State University
Raleigh, USA
ahk@ncsu.edu

Derya Cansever
ECE Department
NC State University
Raleigh, USA
dhcansev@ncsu.edu

Abstract—**Limited data availability in machine learning significantly impacts performance and generalization. Traditional augmentation methods enhance moderately sufficient datasets. GANs struggle with convergence when generating diverse samples. Diffusion models, while effective, have high computational costs. We introduce an Expansive Synthesis model generating large-scale, information-rich datasets from minimal samples. It uses expander graph mappings and feature interpolation to preserve data distribution and feature relationships. The model leverages neural networks' non-linear latent space, captured by a Koopman operator, to create a linear feature space for dataset expansion. An autoencoder with self-attention layers and optimal transport refines distributional consistency. We validate by comparing classifiers trained on generated data to those trained on original datasets. Results show comparable performance, demonstrating the model's potential to augment training data effectively. This work advances data generation, addressing scarcity in machine learning applications.** [1]

Index Terms—**Dimension Expansion, Koopman Operator Theory, Optimal Transport, Expander Graph Mapping**

I. INTRODUCTION

Data scarcity in machine learning limits performance and generalizability across various fields. Current augmentation techniques and GANs lack diversity and stability. Our "Expansive Synthesis" addresses this shortfall by providing a framework to create extensive datasets from limited data. This approach improves data sufficiency for effective training. It builds on "dimension expansion" [5], "expander graphs" [14], self-attention [18], and optimal transport [19] for well-grounded synthesis. Data augmentation in deep learning uses geometric and color transformations to enhance the generalization capabilities of CNNs. Random Oversampling (ROS) and Synthetic Minority Over-sampling Technique (SMOTE) Chawla et al. [1] address class imbalances but have limitations. GANs Goodfellow et al. [7] produce high-quality synthetic data, with improvements from DCGANs Radford et al. [15] and CycleGANs Zhu et al. [20]. Neural Style Transfer Gatys et al. [6] and Fast Style Transfer [10] add artistic dimensions. However, GANs face instability and diversity issues. Meta-learning approaches like Neural Architecture Search (NAS) and AutoAugment [2] automate augmentation strategies but are complex and better suited for smaller-scale tasks.

Motivated by data-starved environments encountered in practice, we seek a Neural Network's capacity to extrapolate from latent space representations of minimal data. Towards lifting the limited data augmentation strategies for severely small datasets, we propose a dimension expansion-inspired synthesis we refer to as "Expansive Synthesis" to generate large-scale, highly training-efficient datasets from small samples. Utilizing regularization techniques like self-attention [18], our approach leverages dimension expansion and its expander graph

perspective [5] to exploit features for robust data synthesis, facilitating model training. This approach is in some sense the dual of our recent data condensation [9] work. We describe a framework where extracted and attention-driven features define a Koopman space, and optimally combine nonlinear components for dimension expansion while preserving data distribution via optimal transport. Section 2 contextualizes the deep learning of a Koopman Operator[11, 4]. Section 3 details the Expansive Synthesis model. Section 4 presents experimental validation, demonstrating the effectiveness in generating expansive datasets and on evaluating classifier performance. Section 5 concludes, summarizing findings and outlining future directions.

II. RELATED BACKGROUND

A Koopman operator[11] enables a linear analysis of nonlinear dynamical systems. We adopt a deep learning model estimate of a Koopman operator whose generated data distribution is preserved by an Optimal Transport (OT) transformation.

A. Koopman Feature Space

Koopman theory offers an elegant and simplified framework for analyzing nonlinear dynamical systems in a linear space of functions of state variables[11]. Building on this theory, a deep learning model was proposed for approximating a Koopman operator, thereby providing flexible application in data-driven problems[4].

B. Data Distribution Preservation

It is critical to preserve the distribution of a synthesized expanded dataset to safeguard its utility for further training (or other) purposes. To that end, Wasserstein distance is used for regularization, as illustrated in Fig. 1(b). Additional details will be discussed in the Methodology section.

III. METHODOLOGY

Pursuing our objective primarily entails,

Claim: *A near-optimal and feature consistent expansion X' of a dataset X can be achieved in a Koopman-data space using Expander Graph Mapping, with proper distribution and feature refinement regularizations.*

Fig. 1 illustrates our proposed method to expand a minimal dataset X into the a larger and feature-rich data set X', with key attributes preserved by a Koopman-representation linear evolution of non-linear dynamics [11]. We first obtain a latent representation $Y \in \mathbb{R}^{n \times d}$ via $\phi : \mathbb{R}^{n \times D} \to \mathbb{R}^{n \times d}$ ($d < D$). Multi-head Spatial Self-Attention captures discriminative features, while Expander Graph Mapping [4] generates diverse datapoints, maintaining original distribution via Wasserstein distance and Covariance loss. To pursue the dimension expansion of small but diverse data sets, we first define a functional space where a meaningful preservation of non-linear features and characteristic of training data at hand.
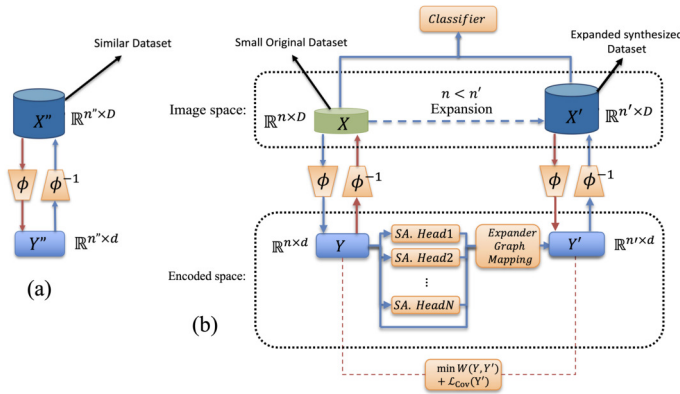
Fig. 1. Overall architecture of Expansive Synthesis model. **(a)** Pretraining phase using a similar larger dataset X'' to learn general features, which are encoded and decoded to produce Y''. **(b)** Fine-tuning phase on the smaller minimal sample dataset X to adapt the model's weights, followed by the expansion of X to generate the synthesized dataset X' using expander graph mapping and self-attention mechanisms. The expanded dataset X' is then used to train a classifier.
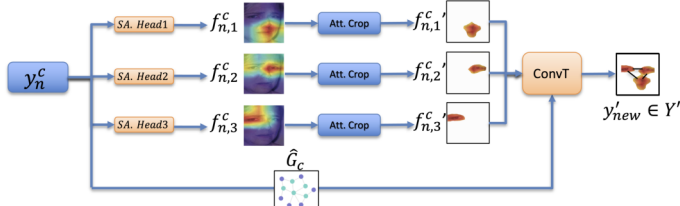

Fig. 2. Architecture of Expander Graph Mapping

To that end, as illustrated on the left-hand side of Fig. 1(b), we aim to glean a Koopman-like characterization of data across all classes of interest as defined above. A training data sample as an aggregate of extracted features (possibly refined by an attention mechanism), may be interpreted as a structured set of graph nodes, benefiting of an expander graph diversification strategy to achieve a potential dimension expansion[5] facilitating our proposed approach as discussed next.

A. Autoencoder:Koopman Feature Space

Our goal of a Data "Expansive Synthesis" is initiated by pre-training a convolutional Autoencoder (AE) uses a dataset X'' analogous to (and distinct from) the target smaller dataset X. This phase equips the AE with understanding of general features in a typically sufficient dataset. The AE's architecture has multiple convolutional layers, each with a set of $cout-number$ filters and associated parameters $W_1, W_2, ..., W_{cout}$, optimized to capture and encode essential data characteristics [7, 13]. With W_i as the ith filter parameters, a corresponding convolutional transformation is defined as $\text{conv}(X''; W_i)$. Intrinsically meant for dimension reduction, the AE encodes X'' to Y'', to retain key features as a compact and informative representation by minimizing a reconstruction loss. As shown in Fig. 1(a), the encoding is carried out by a mapping ϕ, with a corresponding decoding using ϕ^{-1}. In adapting the pre-trained AE to a Koopman-Autoencoder (KAE), we optimize the pre-trained weights W to capture X data characteristics. Our model introduces scaling (γ) and shifting (β) parameters to fine-tune the KAE' performance, to yield the following convolution operation during fine-tuning,

$$\text{conv}(X; W) \cdot \gamma + \beta =$$
$$\text{conv}(X; \{\gamma_1 W_1 + \beta_1, \ldots, \gamma_{cout} W_{cout} + \beta_{cout}\}). \quad (1)$$

Here, γ_i adjusts each filter's activation strength, controlling input feature sensitivity [8]. Higher γ_i enhances filter response

to specific features; β_i modifies activation threshold. These filter-specific parameters enable precise adaptation to the smaller dataset X. Fine-tuning maintains generalization from the larger similar dataset while adapting to X's distinct features. This refined AE generates the expanded synthesized dataset X' (Fig. 1(b), right), transitioning from encoded representation Y' to image space via ϕ^{-1}.

B. Feature Refinement: A Self-Attention Mechanism

We address the incompatibility of longitudinal learning of attention maps across token sequences[18] with images. As shown in Fig. 1(b), our proposed spatial multi-head SA mechanism to improve the resulting capture of spatial dependencies, entails the partitioning of encoded data into patches, and the proximal positioning of flattened features in a stacked representation (with no positional encoding). This in fact enhances the spatial self-attention across all dimensions to focus on interacting features throughout a given image.

C. Expander Graph Mapping

We invoke the notion of Expander Graph Mapping[2] to pursue a dimension expansion and realize our Expansive Synthesis (ES) model. The ES follows a systematic selection of a proper feature set in the encoded space Y which are structurally combined to generate a new and larger dataset $\mathbf{Y}'$ with new datapoints as row vectors y'_{new}.

Proposition: *A composition map of a learned attention transformation and of an Expander Graph Mapping achieve a dimension expansion of $Y \in \mathbb{R}^{n \times d}$ to $Y' \in \mathbb{R}^{n' \times d}$ where $n' > n$ is of $\mathcal{O}(n^L)$, with L as the uniform feature-size selected for each datapoint in $Y \in \mathbb{R}^{n \times d}$.*

In order to formalize this dimension expansion process, we define it as a composition of two maps. Let $\mathcal{A} : \mathbb{R}^{n \times d} \to \mathbb{R}^{n \times d \times L}$ and $\mathcal{E}_G : \mathbb{R}^{n \times d \times L} \to \mathbb{R}^{n' \times d}$ be two maps. The dimension expansion can be then expressed as:

$$(\mathcal{E}_G \circ \mathcal{A})(Y) = \mathcal{E}_G(\mathcal{A}(Y)) = Y' \in \mathbb{R}^{n' \times d}, \quad \forall Y \in \mathbb{R}^{n \times d} \quad (2)$$

The first map, $\mathcal{A}$, represents the application of self-attention to extract L important features from each datapoint. The second map, $\mathcal{E}_G$, represents the Expander Graph Mapping that uses these features to generate new datapoints, resulting in the expanded dataset. Each row vector y_i in $Y \in \mathbb{R}^{n \times d}, i = 1, \cdots, n/c$, as a datapoint may thus be described by a graph $G_{y_i}^c = (V_{y_i}^c, E_{y_i}^c)$. To uniformize the graph size of each datapoint y_i, and robustify its representation, we highlight the most relevant features by L self-attention heads $SA.\text{Head}_j$ to glean feature maps $\{f_{i,j}^c\}$, where c denotes class, i datapoint index ($i \in \{1, \ldots, n/c\}$), and j feature index ($j \in \{1, \ldots, L\}$). Each size-standardized graph $G_{y_i}^c \in \mathcal{G}^c$ (set of all graphs for class c) is trained to maximize its Laplacian $\mathcal{L}$ spectral gap $\lambda_2 - \lambda_1$ to yield a well connected representative class code-graph $\hat{G}^c$[4].[3] Self-attention calculates feature importance, capturing dependencies in y_i:

$$f_{i,j}^c = SA.\text{Head}_j(y_i),$$
$$\forall j \in \{1, \ldots, L\}, \forall i \in \{1, \ldots, n/c\}, \forall c \in \{1, \ldots, C\}. \quad (3)$$

---

[2]An expander graph is a well connected sparse graph, with every not-too large subset of vertices has a large boundary (hence with a capacity of diverse connections for new generation).

[3]For $G = (V, E)$ with $|V| = L$, $\mathcal{L} = D - A$, where D_{ii} is vertex i's degree, and A is the adjacency matrix. $\mathcal{L}$'s eigenvalues are $\lambda_1, \lambda_2, \ldots, \lambda_L$, with $\lambda_1 = 0$ for connected graphs.

This step corresponds to $\mathcal{A} : \mathbb{R}^{n \times d} \to \mathbb{R}^{n \times d \times L}$. To avoid including nuisance and non-characteristic background information $f^c i, j$, we proceed to first crop: $f^{c'} i, j = \text{Att. Crop}(f^c i, j)$ prior to transferring them into Y' using ConvT (Conversion Transformation), guided by $\hat{G}^c = (V_{\hat{G}^c}, E_{\hat{G}^c})$. The dimension expansion constructively generates a new datapoint y'new by selecting one component from each column of the matrix of features $F = [f i j^{c'}] \in \mathbb{R}^{n/c \times L}$ as vertices of $\hat{G}^c$. More generally, we can express this as $\hat{G}^c = (V_{new}^c, E_{\hat{G}^c})$ where $V_{new}^c = S \odot F$. Here, $S \in \mathbb{R}^{n/c \times L}$ is a selection matrix with exactly one 1 in each column and zeros elsewhere, and $\odot$ denotes the Hadamard (element-wise) product. To introduce variability in the selection process, we can define $S = P^i M$, where $P^i \in \mathbb{R}^{n/c \times n/c}$ is a permutation matrix that shuffles the rows, and $M \in \mathbb{R}^{n/c \times L}$ is a fixed matrix with exactly one 1 in each column (in different rows) and zeros elsewhere. The index i in P^i allows for different permutations in each iteration for new datapoint generation. Eq. 4 integrates selected features with the code-graph, ensuring accurate feature placement:

$$y'_{\text{new}} = \sum_{j=1}^{L} \text{ConvT}(f_{i,j}^{c'}, \hat{G}^c), \tag{4}$$

This step corresponds to $\mathcal{E}_G : \mathbb{R}^{n \times d \times L} \to \mathbb{R}^{n' \times d}$. The "ConvT" implements the mapping the enhanced features $f_{i,j}^{c'}$ to nodes $V_{\hat{G}^c}$ of Expansive Graphs of $\hat{G}^c$, while ensuring $v' \in V_{y'_i}$ inherit $G_{y_i}^c$ good connectivity properties [4, 19]. $\hat{G}^c = (V_{\hat{G}^c}, E_{\hat{G}^c})$ maintains robust connectivity with minimal edges. Embedding $f_{i,j}^{c'}$ through $\hat{G}^c$, preserves $\lambda_{\hat{G}_{y_{i,k}}} \approx \lambda_{G_{y'_{i,k}}}$, while maintaining structural and training-valued robustness properties. One additionally notes that L as the number of features used for generating a new datapoint, from the minimal n samples, results in an Expander Graph Mapping yielding $n^L - n$ unique combinations. n^L represents the total possible combinations excluding n original samples in the dataset. As we elaborate next, robust consistency and diversity of the generative data Y'_{new} relative to the original small dataset are computationally preserved by regularization of the overall loss.□

D. Loss Objective Functions and Regularization

The generative ES flow in Fig. 1(b) is sought by regularizing global losses with other local constraints. Global losses include data reconstruction and classification, while local regularizing constraints account for distributional consistency (Wasserstain distance), diversity, and feature refinement.

1) Total Loss ($\mathcal{L}_{total}$):

Claim: *The optimal expansion of a dataset X to X' can be achieved by minimizing a total loss function $\mathcal{L}_{total}$, composed of four terms working in concert:*

$$\mathcal{L}_{\text{total}} = \alpha_0 \mathcal{L}_{re} + \alpha_1 \mathcal{L}_{ce} + \alpha_2 \mathcal{L}_{\mathcal{W}} + \alpha_3 \mathcal{L}_{cov} \tag{5}$$

where $\mathcal{L}_{re}$ is the reconstruction loss ensuring accuracy of the expanded dataset, $\mathcal{L}_{ce}$ is the classification loss maintaining label consistency, $\mathcal{L}_{\mathcal{W}}$ is the Wasserstein distance preserving distributional consistency, $\mathcal{L}_{cov}$ is the covariance loss promoting diversity in the expanded dataset, and $\alpha_0, \alpha_1, \alpha_2$, and α_3 are balancing hyperparameters. These terms jointly ensure that the synthesized samples are diverse, persistent, and maintain a good cover of the original data distribution while preserving essential discriminative features.

2) Reconstruction Loss ($\mathcal{L}_{re}$):

The AE parametrization denoted by ϕ and ϕ^{-1} for encoder and decoder respectively, follows the standard optimization loss between the input and output distribution written as:

$$\mathcal{L}_{re}(\phi, \theta; X) = \mathbb{E}_{Y \sim q_\phi(Y|X)}[-\log p_\theta(X \mid Y)] + \text{KL}[q_\phi(Y \mid X) \| p(Y)], \tag{6}$$

where the first term is the expected negative log-likelihood, and the second term is the KL divergence between the encoded distribution $q_\phi(Y|X)$ and a prior distribution $p(Y)$ [7]. To selectively exploit the most relevant features and further refine the intrinsic linear evolution of the nonlinear dynamics, we induce a composition map of a multi-head attention transformation on Y followed by an expander graph mapping to yield $Y' \in \mathbb{R}^{n' \times d}$ [18]. The expansion phase entails crafting Y' assume a higher dimension (n') while reflecting the original dataset distribution. The latter is guided by the minimization of the Wasserstein distance [17] $\mathcal{W}(Y, Y')$, ensuring the safeguard of the distributional integrity of Y.

3) Distributional Consistency($\mathcal{L}_{\mathcal{W}}$):

Wasserstein distance, a key expression in Optimal Transport Loss [19], measures the cost to align the distribution of encoded data Y with expanded representation Y'. It quantifies the minimal effort to morph p_Y into $p_{Y'}$, measuring dataset expansion effectiveness [16]. The Wasserstein distance [17] is:

$$\mathcal{L}_{\mathcal{W}}(p_Y, p_{Y'}) = \min_{\pi \in \Pi(p_Y, p_{Y'})} \iint c(Y, Y') \pi(p_Y, p_{Y'}) \, dY \, dY' \tag{7}$$

In our formulation, π is the OT plan associating $p(Y)$ and $p(Y')$. Minimizing $\mathcal{L}_{\mathcal{W}}$ ensures Y' statistically resemble Y and preserve its geometric and topological properties, crucial to maintaining data distributional fidelity for subsequent learning tasks dependent on the data' manifold structure [19]. Y' is mapped back to a high-dimensional image space using ϕ^{-1}, resulting in $X' \in \mathbb{R}^{n' \times D}$. Note that using the same AE for encoding and decoding ensures X' be a plausible AE output, retaining the original dataset's structure and distributional properties. A classifier Cl-trained on X' predicts labels $\hat{l}$ as if trained on X, benefiting from X''s distilled information and enabling efficient training with reduced data.

4) Classification Loss ($\mathcal{L}_{ce}$):

The classification loss provides an implicit feedback, measuring discrepancy between predicted and true labels to maintain label consistency. It uses both original (X) and synthesized (X') images, merged and passed through the classifier. The Cross-Entropy (CE) loss is used:

$$\mathcal{L}_{ce}(Cl, \tilde{X}, l) = -\sum_i l_i \log(Cl(\tilde{X}_i)) \tag{8}$$

Here, Cl is the classifier, $\tilde{X}$ the combined set of original and reconstructed data, l the vector of true labels, and $\tilde{X}_i$ the i-th data instance in the merged dataset [7]. This loss component ensures the expanded dataset encapsulate the original data's structural attributes and label characteristics, while preserving essential discriminative features and preventing categorical information dilution in the course of expansion. It enables scalable training on large datasets while maintaining performance comparable to full-dataset training.

5) Expansion Diversity Loss ($\mathcal{L}_{cov}$):

Including covariance constraint term aims at promoting diverse and representative expanded datasets. This regularizer encourages distinct features in latent representations Y to enhance spread and space exploring persistence of generated data representation. Mathematical definition of Cov. Loss is:

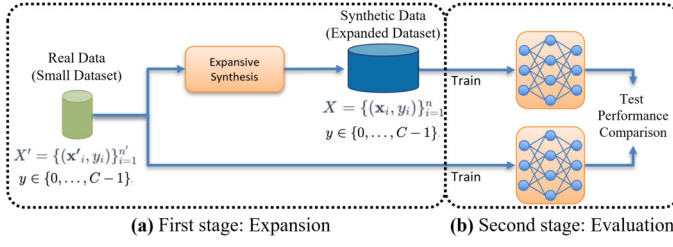$$\mathcal{L}_{cov}(Y') = \|\text{Cov}(Y') - I\|_F^2, \tag{9}$$

(a) First stage: Expansion   **(b)** Second stage: Evaluation

Fig. 3. Stages of Implementation and evaluation of a expansion model

where $\mathrm{Cov}(Y')$ is the covariance matrix of Y', I is the identity matrix, and $\|\cdot\|_F$ is the Frobenius norm.□

Algorithm 1 Expansive Syn. Train Algorithm

1: **Given:**
2:   $X \in \mathbb{R}^{n \times D}$, original dataset with n samples
3:   AE: ϕ (Encoder mapping $\mathbb{R}^D \to \mathbb{R}^d$) and ϕ^{-1} (Decoder mapping $\mathbb{R}^d \to \mathbb{R}^D$)
4:   MHSSA: Multi-head Spatial Self-attention
5:   EGM: Expander Graph Mapping
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:** Pretrained Parameters of AE (ϕ, ϕ^{-1}), Classifier Cl
10: **for** epoch $= 1$ **to** N **do**
11:    **for** class $= 1$ **to** M **do**
12:      $Y \leftarrow \phi(X)$
13:      $Y_{SA} \leftarrow \mathrm{MHSSA}(Y)$
14:      $Y' \leftarrow \mathrm{EGM}(Y_{SA})$
15:      $X' \leftarrow \phi^{-1}(Y')$
16:      $L_{re} \leftarrow \|X' - X\|^2$
17:      $\hat{Y} \leftarrow Cl(X \oplus X')$, ($\oplus$: concatenation)
18:      $L_{ce} \leftarrow -\sum_i l_i \cdot \log(\hat{Y})$, ($l_i$: true labels)
19:      $\mathcal{L}_\mathcal{W} \leftarrow \mathcal{W}(Y, Y')$, ($\mathcal{W}$: Wasserstein Distance)
20:      $\mathcal{L}_{cov}(Y') \leftarrow \left\|\mathrm{Cov}(Y') - I\right\|_F^2$
21:      $L_{total} \leftarrow \alpha_0 L_{re} + \alpha_1 L_{ce} + \alpha_2 \mathcal{L}_\mathcal{W} + \alpha_3 \mathcal{L}_{cov}$
22:      Update Parameters

IV. EXPERIMENTS AND RESULTS

A. Stages of Implementation

Fig. 3 illustrates a two-phase process: expansion and evaluation.

1) First Stage (Expansion):

We start with a small dataset X of n pairs (x_i, l_i), where x_i is input data and l_i are labels ranging from 10 to c. X is expanded to create X', a larger synthesized dataset of n' pairs (x_i', l_i), where x_i' are synthesized datapoints (expanded version) and l_i are labels, maintaining the original label range.

2) Second Stage (Evaluation):

The synthesized dataset X' is used to train a classifier. We train the same type of classifier on the corresponding small dataset X. The performance of both classifiers is evaluated on a test set, and as we show, the goal of training for inference on X' outperforms that carried out on X and its augmented version (using current techniques) is achieved. The hypothesis is that if X' is a good expansion of X, the X'-trained classifier should generalize almost as well as the X-trained one on unseen data, demonstrating that X' captures X's core information, enabling effective training with less data.

B. Results and Comparisons

The effectiveness in generating large-scale, high-performing datasets from minimal samples is demonstrated by comparing classifiers trained on these synthesized datasets against those trained on original and also on classically augmented datasets. Tables I compare classification accuracies across MNIST [3],

TABLE I
CLASSIFICATION ACCURACY RESULTS (%) INITIATED WITH 10 AND 100 IMAGES PER CLASS (IMG/CLS) POST EXPANSION FOR OUR EXPANSIVE SYNTHESIS MODEL WITH DIFFERENT SETTINGS ON DIFFERENT DATASETS. ACRONYMS: RWA (ROW-WISE ATTENTION), SA (SELF-ATTENTION), SC (SKIP CONNECTIONS), MHSA (MULTI-HEAD SELF-ATTENTION), LIN. INT. (LINEAR INTERPOLATION), EGM (EXPANDER GRAPH MAPPING).

Dataset / Exp. Setting	Original data	Expanded data (initiated with 10 Img/Cls & E. R. = 10)				
		RWA + No SC + Lin. Int.	SA + No SC + Lin. Int.	SA + SC + Lin. Int.	MHSA + SC + Lin. Int.	MHSA + SC + EGM
MNIST	52.3	63.2	65.1	68.6	69.1	**69.5**
CIFAR-10	41.5	52.1	54.2	58.7	60.1	**63.1**
CIFAR-100	30.1	42.3	44.7	49	52.3	**55.1**

Dataset / Exp. Setting	Original data	Expanded data (initiated with 100 Img/Cls & E. R. = 10)				
		RWA + No SC + Lin. Int.	SA + No SC + Lin. Int.	SA + SC + Lin. Int.	MHSA + SC + Lin. Int.	MHSA + SC + EGM
MNIST	64.9	74.7	76.1	80.4	83.1	**84.1**
CIFAR-10	52	62.9	64.7	69.8	73.9	**74.4**
CIFAR-100	38.3	51.2	54.2	59.1	63.5	**63.9**

CIFAR-10, and CIFAR-100 [12]. Classifiers trained on 10 original images achieve the lowest accuracies. Expansive synthesis with row-wise self-attention improves accuracy by an average 20% over the baseline. Spatial attention further enhances performance by 6%, while adding skip connections yields an additional 8% improvement. Multi-head spatial self-attention boosts results by 4%, and incorporating Expander Graph Mapping (EGM) achieves the best results with a 6% enhancement over the previous configuration. The ablation study demonstrates that each module incrementally improves performance, with multi-head self-attention and EGM optimizing feature representation. Table II compares classification accuracy for models initialized with 10 or 100 images per class across MNIST, CIFAR-10, and CIFAR-100. The expansive synthesis model consistently outperforms traditional augmentation. On MNIST, accuracy improves from 52.3% to 54.7% with 10 images per class and from 64.9% to 66.8% with 100 images per class. Similar improvements are seen for CIFAR-10 and CIFAR-100. These results demonstrate our model's effectiveness in enhancing dataset quality and classifier performance.

TABLE II
COMPARISON OF CLASSIFICATION ACC. (%) FOR MODELS INITIATED WITH 10 AND 100 IMAGES PER CLASS (IMG/CLS) ACROSS DIFFERENT DATASETS. THE TABLE COMPARES THE PERFORMANCE OF CLASSIFIERS TRAINED ON ORIGINAL DATA, CLASSICALLY AUGMENTED DATA, AND DATA EXPANDED USING OUR ES METHOD, ALL WITH THE SAME EXPANSION RATIO (ER=10).

Dataset / Exp. Setting	Initiated with 10 Img/Cls (E. R. = 10)			Initiated with 100 Img/Cls (E. R. = 10)		
	Original data	Classically Augmented	MHSA + SC + EGM	Original data	Classically Augmented	MHSA + SC + EGM
MNIST	52.3	54.7	**69.5**	64.9	66.8	**84.1**
CIFAR-10	41.5	46.7	**63.1**	52	57.4	**74.4**
CIFAR-100	30.1	32.8	**55.1**	38.3	41.5	**63.9**

V. CONCLUSION AND FUTURE WORKS

In this paper, we introduced the Expansive Synthesis model for generating large-scale, high-performing datasets from minimal samples. It uses a Koopman feature space within an AE to transform non-linear latent space features into a linear space. Composing a multi-head spatial self-attention mechanism for improving feature extraction, with Expander graph mappings to secure feature connectivity during expansion, together with proper regularization such as Optimal transport preserving distribution integrity yields a principled, effective enrichment of small datasets destined for training to learn. We also plan to extend our approach to high-resolution data in future work. This is an effective alternative solution to synthetic data augmentation which has been shown and even GANs which demand large amounts of data for training and are still susceptible to instability.

REFERENCES

[1] Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. Smote: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 16:321–357, 2002.

[2] Ekin D. Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V. Le. Autoaugment: Learning augmentation strategies from data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[3] Li Deng. The mnist database of handwritten digit images for machine learning research [best of the web]. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012.

[4] Sourya Dey and Eric William Davis. Dlkoopman: A deep learning software package for koopman theory. In *Learning for Dynamics and Control Conference. PMLR*, 2023.

[5] Michael A. Forbes and Venkatesan Guruswami. Dimension expanders via rank condensers. *CoRR*, abs/1411.7455, 2014. URL http://arxiv.org/abs/1411.7455.

[6] Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge. A neural algorithm of artistic style. *arXiv preprint arXiv:1508.06576*, 2015.

[7] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems 27 (NIPS 2014)*, 2014.

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[9] Vahid Jebraeeli, Bo Jiang, Derya Cansever, and Hamid Krim. Koopcon: A new approach towards smarter and less complex learning. *arXiv preprint arXiv:2405.13866 (Accepted in the IEEE International Conference on Image Processing (ICIP))*, 2024.

[10] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2016.

[11] Bernard O. Koopman. Hamiltonian systems and transformation in hilbert space. *Proceedings of the National Academy of Sciences*, 17(5):315–318, 1931.

[12] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. Technical report, Technical Report, 2009.

[13] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems 25 (NIPS 2012)*, 2012.

[14] Gregory A Margulis. Explicit constructions of expanding graphs. *Problemy Peredachi Informatsii*, 9(4):71–80, 1973.

[15] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

[16] Siddharth Roheda, Ashkan Panahi, and Hamid Krim. Fast optimal transport for latent domain adaptation. In *2023 IEEE International Conference on Image Processing (ICIP)*, pages 1810–1814. IEEE, 2023.

[17] Leonid Nisonovich Vaserstein. Markov processes over denumerable products of spaces, describing large systems of automata. *Problemy Peredachi Informatsii*, 5(3):64–72, 1969.

[18] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems 30 (NIPS 2017)*, 2017.

[19] Cédric Villani. *Optimal transport: old and new*, volume 338. Springer, 2009.

[20] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017.