The Dawn of KAN in Image-to-Image (I2I) Translation: Integrating Kolmogorov-Arnold Networks with GANs for Unpaired I2I Translation

Arpan Mahara*, Naphtali D. Rishe†, Liangdong Deng‡
Knight Foundation School of Computing and Information Sciences
Florida International University
Miami, FL 33199
*amaha038@fiu.edu, †rishen@cs.fiu.edu, †liadeng@cs.fiu.edu

***Abstract*—Image-to-Image translation in Generative Artificial Intelligence (Generative AI) has been a central focus of research, with applications spanning healthcare, remote sensing, physics, chemistry, photography, and more. Among the numerous methodologies, Generative Adversarial Networks (GANs) with contrastive learning have been particularly successful. This study aims to demonstrate that the Kolmogorov-Arnold Network (KAN) can effectively replace the Multi-layer Perceptron (MLP) method in generative AI, particularly in the subdomain of image-to-image translation, to achieve better generative quality. Our novel approach replaces the two-layer MLP with a two-layer KAN in the existing Contrastive Unpaired Image-to-Image Translation (CUT) model, developing the KAN-CUT model. This substitution favors the generation of more informative features in low-dimensional vector representations, which contrastive learning can utilize more effectively to produce high-quality images in the target domain. Extensive experiments, detailed in the results section, demonstrate the applicability of KAN in conjunction with contrastive learning and GANs in Generative AI, particularly for image-to-image translation. This work suggests that KAN could be a valuable component in the broader generative AI domain.**

***Index Terms*—Generative AI, Image-to-Image translation, Generative Adversarial Networks (GANs), Contrastive Learning, Multi-layer Perceptron, Kolmogorov-Arnold Networks (KANs), PatchNCE Loss**

I. Introduction

Generative AI is prevalent across various research fields, including images, texts, videos, and more. It has been developed to achieve generative outcomes such as text-to-text [1], text-to-image [2], image-to-text [3], text-to-video [4], video-to-video [5], and image-to-image generation or translation [6]. The present study primarily focuses on image-to-image translation, a subdomain of Generative AI.

The popularity and success of Generative AI can be largely attributed to the flexibility and expressiveness of Multi-layer Perceptrons (MLPs) [7]–[9]. Despite their significant contributions to the field of Deep Learning (a branch of Generative AI), MLPs have certain limitations, such as the inability to optimize univariate functions effectively and their lower accuracy compared to splines in low-dimensional spaces.

Recently, the Kolmogorov-Arnold Network (KAN) [10], based on the Kolmogorov-Arnold representation theorem [11], has been proposed as a potential replacement for MLPs. KAN combines the strengths of MLPs and splines, offering improved accuracy and interpretability. The authors have demonstrated KAN's better performance compared to MLPs in small-scale AI applications and suggested its potential in broader applications.

Image-to-image translation is a well-researched domain where images from domain A are translated to domain B using generative mechanisms to achieve various outcomes such as facial attribute manipulation [12], medical image analysis [13], and geospatial analysis [14]. Among various mechanisms like VAEs [15] and diffusion models [16], GANs have been widely used to achieve image-to-image translation. Initially, the Pix2Pix model [6] utilized paired images of domains A and B in a supervised manner for image-to-image translation. However, due to the impracticality of obtaining paired images, the CycleGAN model [17] was introduced to perform translation in an unsupervised manner. Several other promising GANs equipped with unsupervised principles, such as GCGAN [18], CUT [19], DCLGAN [20], and StarGAN [21], have since been proposed. Among these, CUT is particularly notable for its accuracy, computational efficiency, and time complexity due to its uni-directional training. This model combines Generative Adversarial training with mutual information maximization using contrastive learning to generate high-quality images in the target domain.

A noteworthy aspect of this mechanism is its use of contrastive learning, inspired by the SIMCLR [22] framework, where features processed from different layers of the generator are fed into a two-layer MLP to enhance feature representation. This process allows for high-quality image generation in the target domain.

Understanding the importance of KAN, to demonstrate and potentially prove the applicability of KAN in Generative AI, specifically in image-to-image translation, we first propose a novel customization of KAN and construct an efficient two-layer KAN, which we then use to replace the two-layer MLP in CUT, giving rise to the KAN-CUT model. Our overall contributions in the present study are:

- Reformulated the KAN architecture to improve efficiency by avoiding the expansion of the input tensor and remov-

ing additional entropy regularization for L1 normalization.
- Enhanced the KAN layer by implementing an activation function that concatenates the basis function and spline function, replacing the original addition operation, and supplemented with additional processing using Gated Linear Units (GLU).
- Innovatively replaced the two-layer MLP in the CUT model with our efficient two-layer KAN, creating the KAN-CUT model for unpaired image-to-image translation.
- This study marks the first integration of KAN in the image-to-image translation domain (a subdomain of Generative AI), potentially paving the way for numerous applications of KAN in different Generative AI domains to achieve better outcomes.

II. Related Work

In unpaired image-to-image translation within the GANs domain, CycleGAN [17] was a foundational work where images from domains A and B were trained on the cyclic principle. This principle dictates that if b is an image generated by generator G with a as an input, then we should be able to obtain the original image a when we feed the generated image b to another generator F. The ultimate goal is to obtain a mapping of each instance from A and B in the absence of paired images, so that a previously unseen instance image from domain A can be accurately translated to another instance of an image in domain B. While the model was very successful and is still being applied in various research endeavors, it has limitations such as being restrictive, computationally expensive, and having high time complexity due to its cyclic nature.

To address limitations related to time complexity and computational expense, several different works in GANs have been presented that make the training process single-directional, thus eliminating the need for auxiliary generators and discriminators. Notable examples include GCGAN [18] and CUT [19]. GCGAN leverages geometric consistency to impose the structural correspondence between the input and output images, ensuring that the generated image maintains the geometrical structure of the input image.

Contrastive Learning, a methodology that favors achieving numerous Machine Learning (ML) and Deep Learning tasks such as classification, segmentation, simulation, and generation, operates with a self-supervised mechanism in the absence of supervised data. SIMCLR, a contrastive framework presented in [22], was a remarkable work in Contrastive Learning based on image data that obtained better results than previous semi-supervised and self-supervised tasks, and comparable performance to well-known supervised models like ResNet50 [23] in the ImageNet dataset. In their work, the authors showcased that simple data augmentation and applying contrastive learning to the patches of augmented images can help determine similar features in embedding space, and maximizing mutual information can lead to the final goal without the need for human-labeled data. More importantly, the authors of SIMCLR [22] found that instead of directly applying contrastive learning to features or vectors in lower dimensions, processing the features with a two-layer MLP before applying contrastive learning yielded better results.

Inspired by SIMCLR, [19] proposed the CUT model, which utilized contrastive learning at the patch level, integrated with GAN in a uni-directional fashion, to address the limitations of CycleGAN. This model successfully achieved better performance based on quantitative evaluation along with lower time complexity. The most important procedure in this method is that during training, patches drawn from the input image and target domain image are processed through network layers and propagated to a two-layer MLP to obtain enriched features or vectors, following [22]. In the obtained enriched vectors, contrastive learning is carried out, and maximization of information among corresponding patches is done to maintain qualitative and selective image generation. It can be seen that the two-layer MLP is a crucial component that helps generate enriched vectors or features used for image-to-image translation.

Some subsequent works focused on improving the quality of generation while still incorporating bi-directional training with the need for two generators and two discriminators but with additional changes in the principle. Examples of these works include DualGAN [24] and DCLGAN [20]. DualGAN introduced a dual learning mechanism inspired by natural language translation. It utilizes a primary GAN to convert images from the input domain to the output domain and a secondary GAN to reverse this conversion. This architecture ensures consistent and accurate mappings between domains by using reconstruction loss, which enhances the quality and robustness of the generated images. Similarly, DCLGAN [20] shows that reverse mapping can be achieved without depending on the generated images, leading to non-restrictive mapping, unlike the mechanism of CycleGAN [17]. It aligns with CycleGAN [17] by using two generators and two discriminators and bi-directional training, additionally applying contrastive learning, similar to CUT [19], by maximizing mutual information among patches in both directions. DCLGAN can be considered an upgraded version of CUT. These GANs, successful in unpaired image-to-image translation, were mainly constructed to function between two domains. However, there are scenarios where generation is needed among multiple domains, such as generating images from various attributes or styles. StarGAN [21] was proposed to achieve multi-domain translation using a single generator and discriminator. StarGAN [21] was proposed to achieve translation across multiple domains using a unified generator and discriminator. StarGAN acquires mappings among various domains by conditioning the generator on domain labels, allowing it to flexibly translate an input image to any desired target domain.

KANs [10], recently proposed, have been applied to various Machine Learning and Deep Learning tasks, such as image classification [25], image segmentation and generation [26], time-series analysis [27], and graph-based learning [28]. Even

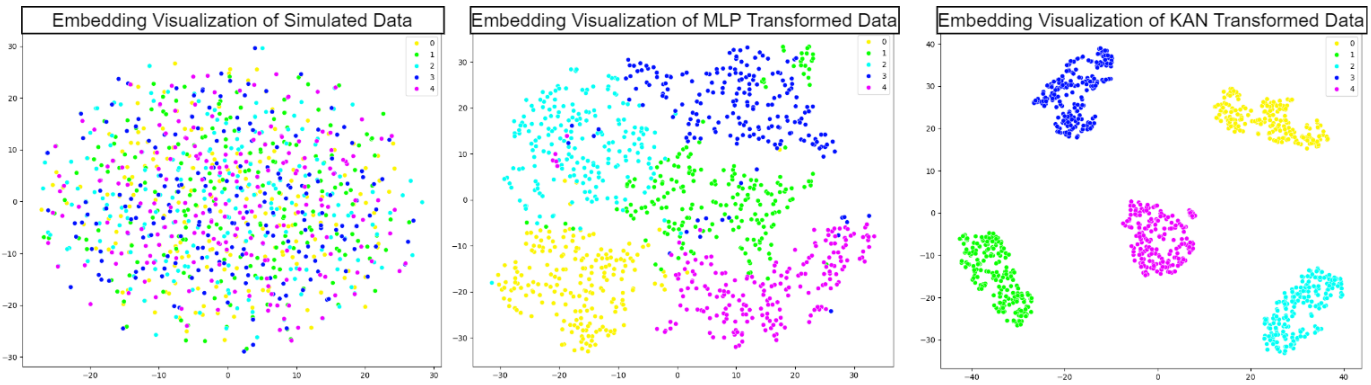

Fig. 1. Comparison of Embedding Visualizations of Simulated Data with Transformation by Simple MLP and KAN Models

though [26] presents a generative model called Diffusion U-KAN, it only depicts generating images based on training limited to one domain. Due to the lack of work in the generation or translation of images from one domain to another, we propose to integrate KAN into the existing CUT model [19] and present the new KAN-CUT model, capable of generating higher quality images across two domains. The code is available at **https://github.com/amaha7984/KAN-CUT**.

III. Proposed Approach

In this section, we first detail the architecture of the Kolmogorov-Arnold Network (KAN), followed by our novel changes and enhancements to the architecture. We then proceed to integrate the customized efficient KAN into the domain of image-to-image translation by replacing the two-layer Multi-layer Perceptron (MLP) with a two-layer KAN in the Contrastive Unpaired Image-to-Image Translation (CUT) [19] model, resulting in the KAN-CUT model.

A. Kolmogorov-Arnold Networks (KAN)

Kolmogorov-Arnold Networks (KANs) have been discussed as a significant advancement in machine learning, often referred to as Machine Learning 2.0 among researchers. At a high level, our approach facilitates the generation of better-informed features in lower dimensions where contrastive learning can be performed. Due to the lack of previous research on KANs in representation learning, it is not straightforward to deduce their relevance for understanding or generating feature vectors in embedding space better than the well-established MLPs. To build our initial confidence, we conducted a simulation where data had three different labels (categories) with various features in a simulated embedding space. After training simple MLP and KAN models separately on the given simulated data, we visualized the results using t-SNE [29] and observed that KANs performed better in clustering similar data points in the feature representation, as depicted in Fig. 1.

Before exploring how KAN can be used in generative tasks, we first discuss the fundamentals of KAN. Unlike MLP, which is based on the Universal Approximation Theorem, KAN is based on the Kolmogorov-Arnold Representation Theorem [11]. This theorem states that if f is a multivariate continuous function on a bounded domain, then f can be simplified into a finite composition of continuous single-variable functions and the binary operation of addition:

$$f(x) = f(x_1, \cdots, x_n) = \sum_{q=1}^{2n+1} \Phi_q \left(\sum_{p=1}^{n} \varphi_{q,p}(x_p) \right) \quad (2.1)$$

where $\varphi_{q,p} : [0,1] \to \mathbb{R}$ and $\Phi_q : \mathbb{R} \to \mathbb{R}$.

Our main goal is to translate or generate images from one domain, say domain A, to another domain, say domain B. For this, we seek a function g such that $g(A)$ yields $B' \approx B$. According to [10], we can approximate g by learning a discrete number of one-dimensional functions, parameterized by B-spline curves with tunable coefficients of local B-spline functions.

KAN is comparable to MLP, but instead of learnable weights, it has learnable activation functions on edges, and summation on the resultant learned function's output is performed at the nodes. As explained in [10], a KAN layer with n_{in}-dimensional inputs and n_{out}-dimensional outputs can be represented as a matrix of 1D functions:

$$\Phi = \{\varphi_{q,p}\}, \quad p = 1, 2, \cdots, n_{\text{in}}, \quad q = 1, 2 \cdots, n_{\text{out}}, \quad (2.2)$$

where each function's $(\varphi_{q,p})$ parameters are trainable. In the Kolmogorov-Arnold theorem, the internal functions constitute a KAN layer with $n_{\text{in}} = n$ and $n_{\text{out}} = 2n + 1$, while the external functions form another KAN layer with $n_{\text{in}} = 2n + 1$ and $n_{\text{out}} = 1$. Thus, the Kolmogorov-Arnold representations in Eq. (2.1) are constructed by composing two KAN layers.

B. B-Splines

B-splines are piecewise polynomial curves constructed from a sequence of lower-order polynomial segments. They are defined by a set of control points P_i and a knot vector u that determines the influence of each control point on the B-spline curve.

The basis function for B-splines of order 0 (piecewise constant) is defined as:

$$M_{i,0}(u) = \begin{cases} 1 & \text{if } u_i \leq u < u_{i+1} \\ 0 & \text{otherwise} \end{cases}$$

For higher-order basis functions (piecewise linear, quadratic, etc.), the recursive definition is:

$$M_{i,k}(u) = \frac{u - u_i}{u_{i+k} - u_i} M_{i,k-1}(u) + \frac{u_{i+k+1} - u}{u_{i+k+1} - u_{i+1}} M_{i+1,k-1}(u)$$

where $k = 1, \ldots, d$, and d is the degree of the B-spline.

The B-spline curve is then defined by:

$$c(u) = \sum_{i=0}^{m} P_i M_{i,d}(u)$$

where P_i are the control points, $M_{i,d}(u)$ are the basis functions of degree d, and u is the parameter.

In this context, u_i represents the elements of the knot vector, and m corresponds to the total number of control points minus one.

We have demonstrated the importance of the two-layer MLP in the image-to-image translation domain, particularly in applications involving contrastive learning and generative adversarial networks, as mentioned in Section II. We have also explored the concept of the KAN network and its potential applicability. Below, we will introduce our novel changes and refinements of the KAN network, which we will later apply to generative mechanisms.

C. Efficient Two-Layer KAN

For a layer with in_features inputs and out_features outputs, the original implementation expands the input to a tensor of the shape $(\text{batch\_size}, \text{out\_features}, \text{in\_features})$ to apply the activation functions. As mentioned above in Subsection A, all the activation functions are linear combinations of a fixed set of basis functions, B-spline curves. Following the work of [30], for efficient processing, we reconstructed the computation process by refraining from the additional step of expanding the input. Instead, we apply learnable activation functions directly to the input tensor and then combine the results linearly. Mathematically, if $\mathbf{i}$ is the input, and $\mathbf{B}(\mathbf{i})$ represents the B-spline basis functions applied to the input, the reformulated procedure of applying the activation function σ on the input can be expressed as:

$$\sigma(\mathbf{i}) = \sigma_b(\mathbf{i}) + \mathbf{B}(\mathbf{i})$$

where $\sigma_b(\mathbf{i})$ is the base activation function applied to the input $\mathbf{i}$.

This modification simplifies the computation to a straightforward matrix multiplication, efficiently supporting both forward and backward passes. Similarly, regarding regularization, L1 regularization [31] is applied in MLP to achieve sparsity and improve model interpretability. In the original KAN [10], L1 regularization was adapted for learnable activation functions rather than linear weights since there are no learnable weights in terms of KAN. Specifically, the L1 norm was defined as the average magnitude of these activation functions over their inputs. This approach required the activation functions to be sparsified by their L1 norm. Additionally, an entropy regularization was necessary to further enforce sparsity. In our implementation of KAN, we align with the original approach by applying L1 regularization to the B-spline activation functions ('spline_activation') to enforce sparsity and control overfitting. The main difference is the removal of the tensor expansion step and additional entropy regularization, making the process more efficient.

To obtain an optimizable layer, the original KAN implementation [10] used a residual connection mechanism and expressed the activation function $\phi(x)$ as the combination of the basis function $b(x)$ and the spline function, given as:

$$\phi(x) = w_b b(x) + w_s \text{spline}(x)$$

where w_b and w_s are learnable weights, but they can be considered redundant since they can be merged into $b(x)$ and spline(x) [10].

Instead of the additive operation, we applied concatenation and processed it with Gated Linear Units (GLUs) [32], in a novel way, to enrich features while maintaining the same number of parameters to avoid an increase in computation, as detailed in the subsection below.

1) Integration of Basis and Spline Functions via Concatenation and Gated Linear Units (GLUs): In our Kolmogorov Arnold Network (KAN) layer implementation, we aim to enhance expressive power while maintaining the same feature dimensionality. To obtain this, we propose the following approach:

1) **Concatenation**: Compute the outputs of $b(x)$ and $\text{spline}(x)$, then concatenate them:

$$\text{concatenated\_output} = \text{concat}(w_b \cdot b(x), w_s \cdot \text{spline}(x))$$

2) **Gated Linear Units (GLUs)**: Apply GLU to the concatenated output for non-linear transformation while maintaining the original feature dimensionality:

$$\phi(x) = (W \cdot \text{concatenated\_output} + b) \odot \rho(V \cdot \text{concatenated\_output} + c)$$

where $\odot$ denotes element-wise multiplication and ρ represents the penalized tanh function with our customized implementation:

$$\rho_\alpha(x) = \tanh(x) + \alpha \cdot \max(-x, 0)$$

Here, α is a hyperparameter that controls the penalty applied to the negative part of the input x. It is worth mentioning that, while the sigmoid function is generally used in GLU, we integrated the penalized tanh as shown above, as it yielded better results during experiments.

With these steps, we make different use of the residual connection mechanism, unlike the original KAN implementation, and implement the activation function $\phi(x)$ as the concatenation of the basis function and the spline function. The construction ends with non-linear transformations to maintain the initial dimension of the concatenated inputs.

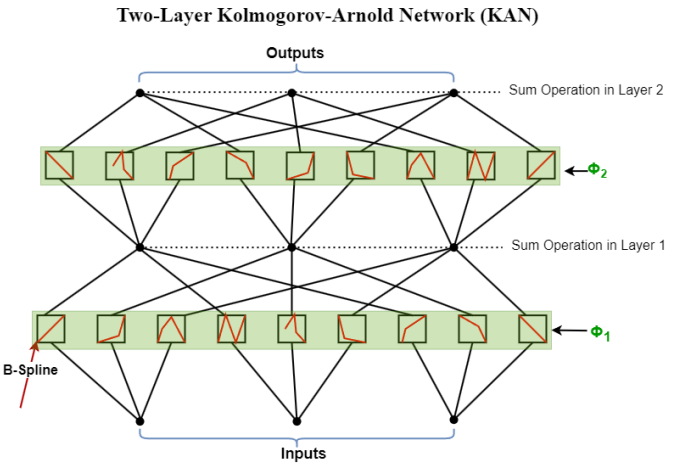

Fig. 2. **Illustration of Two-Layer KAN.** Φ_1 and Φ_2 represent the collection of all 1D functions parametrized by B-Spline in layer 1 and layer 2, respectively.

Through these novel customizations of KAN, we finally construct a two-layer efficient KAN. Fig. 2 illustrates the process. Now, we will show how we integrate this obtained innovative two-layer efficient KAN into contrastive learning and GANs in unpaired image-to-image translation and give rise to the KAN-CUT model.

D. Formulation of KAN-CUT Model

Conforming to the principle of Generative Adversarial Networks (GANs), we integrate two neural network models: a generator and a discriminator. The generator's role is translating (generating) from input domain A to output domain B, while the discriminator determines whether the provided image is genuine or artificial. Following recent literature in image-to-image translation [17], [19], [20], we have implemented ResNet-based encoder-decoder architecture of the generator.

During the translation of an image $a\ in\ A$ to an image $b\ in\ B$, not only should the entire image share information, but the individual patches within the image should also share information while transforming domain-dependent information in the generated image. For instance, let's say we are translating an image from a cat to a dog. The generated dog image should share local information in the patches, meaning the eyes and nose should appear in the same locations as they were in the cat image (see Fig. 3, where we depict the appearance of the eyes of a cat and a dog with blue-colored squares), while the obtained domain-dependent features such as fur patterns and facial structure should be unique to a dog (see Fig. 3, where we highlight fur patterns with red-colored squares). We can achieve this effect of the images sharing patch-level information by maximizing information between corresponding patches while minimizing information in the non-corresponding patches. Since our work is based on an unsupervised mechanism, it is not straight forward to determine which patches are similar or corresponding and

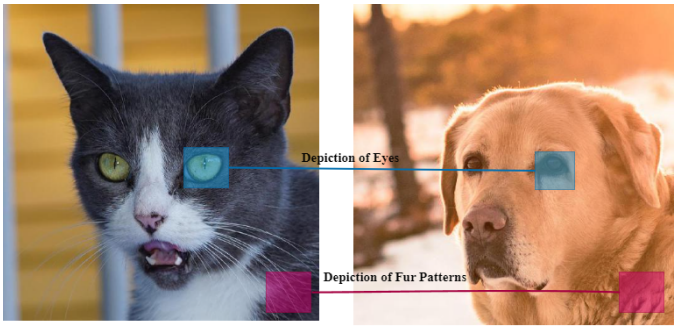

Fig. 3. Illustration showing that images should share information based on patches. Images are sourced from the AFHQ dataset [34] and are reproduced under the Creative Commons Attribution-NonCommercial 4.0 International License.

how to maximize the information. Thanks to the predictive mechanism of contrastive learning [22], [33], we can use contrastive loss to successfully select the correct positive patch from a given set of patches in which there exists one positive and several negatives for a given query. More specifically, following [19], working in a vector representation, we construct an (N+1)-way classification problem, where the probability of positive being selected over negatives is achieved with cross-entropy loss given as:

$$L(q, p, \{n_i\}) = -\log \frac{\exp(q, p)/\tau)}{\exp(q, p)/\tau) + \sum_{i=1}^{N} \exp(q, n_i)/\tau)} \tag{1}$$

where $q,\ p \in \mathbb{R}^K$ and $n_i \in \mathbb{R}^{N \times K}$ are K-dimensional vectors corresponding to query, positive, and N negatives, respectively. τ is a temperature parameter to scale the distances between vectors. Following this procedure, we can select the correct query and positive vector from the set of vectors and maximize the mutual information between them in the embedding space. It is clear that we can obtain the selection and maximization of mutual information between vectors. Now, the question arises: how can we obtain the vectors or features where we can apply contrastive estimation?

As mentioned in [19], since images are processed in the generator G, we can utilize features from intermediate layers of the generator's encoder, G_{enc}. Inspired by [19] and [22], we select X layers from the encoder and pass them to our innovative two-layer efficient KAN instead of a two-layer MLP (see Fig. 4 illustrating the difference in layer implementation between KAN and MLP), producing a more informative stack of features. For instance, let's say we feed image a to the generator G and generate image $G(a)$. To obtain stacks of features from which the positive feature or vector is chosen, we select the x-th layer from the encoder through which a is being processed and pass it to the two-layer KAN network K_l to produce a stack of features $\{z_x\}_X = \{K_x(G^l_{\text{enc}}(a))\}_X$ (as depicted with the red-colored arrow in Fig. 5).

In each layer, there are several spatial locations quantified by S_x, where $s \in \{1, \ldots, S_x\}$. We refer to the positive feature as $z_x^s \in \mathbb{R}^{C_x}$ and the negative features as $z_x^{S \setminus s} \in \mathbb{R}^{(S_x-1) \times C_x}$, where C_x denotes the channel count at each layer. To denote a

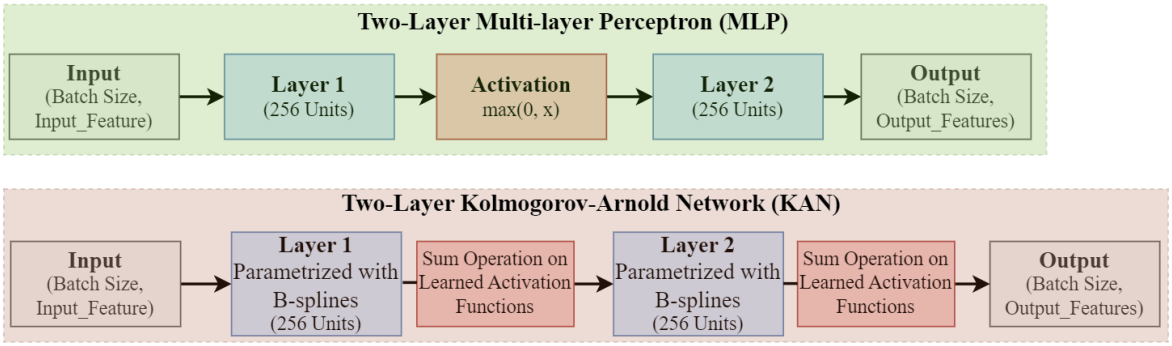

Fig. 4. Comparison between the Two-Layer MLP and Two-Layer KAN architectures.

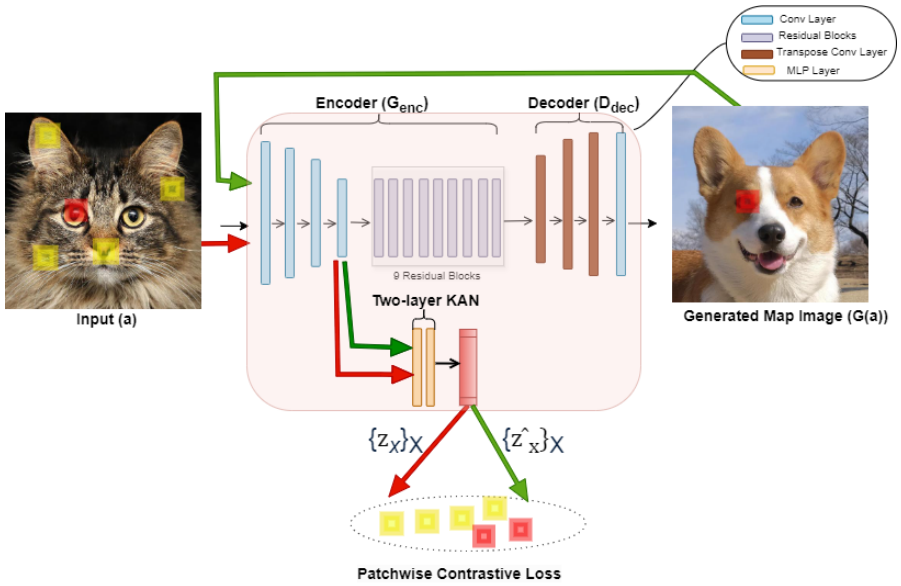

Fig. 5. Architecture of Generator used in KAN-CUT. The cat and dog images are sourced from the AFHQ dataset [34] and are reproduced under the Creative Commons Attribution-NonCommercial 4.0 International License.

query vector or feature in this instance, we feed the generated image $G(a)$ to the same encoder, select the x-th layer, and pass it to the two-layer KAN network K_l to produce another stack of features, $\{\hat{z}_x\}_X = \{K_x(G^l_{\text{enc}}(G(a)))\}_X$ (as depicted with green-colored arrow in Fig. 5). From this feature stack, we refer to a corresponding query point as $\hat{z}^s_x$. It is worth mentioning that negatives vectors are drawn from the same input image, following [19]. Once we have the query, positive, and negative vectors, we can perform constrastive learning using the loss function given above in equation (1). The above procedure is performed for an instance of a spatial location and from a layer; however, we need to do this for all spatial locations and layers. For that, we use PatchNCE loss as presented in [19], which can be expressed as:

$$L_{\text{PatchNCE}}(G, K, A) = \mathbb{E}_{a\sim A}\left[\sum_{x=1}^{X}\sum_{s=1}^{S_x} X\left(\hat{z}^s_x, z^s_x, z^{S\setminus s}_x\right)\right] \quad (2)$$

1) Adversarial Loss for Generation of Realistic-Looking Images: We can achieve mutual information maximization at the patch level with the above procedure using contrastive learning and the two-layer efficient KAN. For high-quality image generation, we use adversarial loss from generative adversarial networks (GANs). The initial work in GANs was introduced in [35], and this architecture is considered as Vanilla GAN. The authors introduced the use of a sigmoid cross-entropy loss for GANs. This loss can sometimes cause vanishing gradients when data samples fall within the correct classification boundary but remain distant from the actual data distribution. To mitigate this problem, [36] proposed Least Squares Generative Adversarial Networks (LSGANs), replacing the binary cross-entropy loss with a least squares loss. Following principles of [35] and [36], to guarantee the generation of authentic images perceived by humans in domain B', such that $B' \approx B$ for the given input images from domain A, we utilize an adversarial loss based on the least squares loss. The adversarial loss, also known as LSGAN loss, comprises two main components: the loss for the generator and the loss for the discriminator. These losses guide the back-

propagation process through which the neural networks are trained and updated. The general procedure involves training the generator to generate images in the target domain that are identical to real images, whereas the discriminator learns to differentiate real images from generated ones.

The adversarial loss can be formulated as:

$$L_{\text{LSGAN}}(D) = \frac{1}{2}\mathbb{E}_{b\sim B}[(D(b)-1)^2] + \frac{1}{2}\mathbb{E}_{a\sim A}[(D(G(a)))^2]$$
$$L_{\text{LSGAN}}(G) = \frac{1}{2}\mathbb{E}_{a\sim A}[(D(G(a))-1)^2] \tag{3}$$

In these equations, $D(b)$ represents the discriminator's decision for a ground truth image b from the target domain, where the label for real images is set to 1. $G(a)$ is the generated image from the input image a from the input domain, where the label for fake images is set to 0. The generator G aims to minimize $L_{\text{LSGAN}}(G)$, making the discriminator believe that the generated images are real by pushing $D(G(a))$ towards 1. The discriminator D aims to minimize $L_{\text{LSGAN}}(D)$, correctly distinguishing real images from generated ones by pushing $D(b)$ towards 1 and $D(G(a))$ towards 0. This results in a minimax game between the two, similar to the initial work in [35], promoting the generation of high-quality, realistic images.

2) Final Loss Functions: Following the same principle in [19], we utilize three different loss functions involving only one generator and discriminator. These loss functions include the adversarial loss for generating realistic-looking images, the PatchNCE loss, L_{PatchNCE}(G, K, A), for ensuring patch-level correspondence, and a similar PatchNCE loss, L_{PatchNCE}(G, K, B), to prevent inappropriate translation by the generator, similar to the identity loss presented in [17]. The combined final loss function is formulated as follows:

$$\begin{aligned} L_{\text{final}} =& L_{\text{LSGAN}}(G, D, A, B) + \\ & \lambda_{\text{PatchNCE-A}} L_{\text{PatchNCE}}(G, K, A) + \\ & \lambda_{\text{PatchNCE-B}} L_{\text{PatchNCE}}(G, K, B) \end{aligned} \tag{4}$$

IV. Experiments

For evaluation, we selected two well-known, widely utilized, and publicly accessible datasets for research in image-to-image translation: $Horse \rightarrow Zebra$ and $Cat \rightarrow Dog$.

A. Datasets

a) $Horse \rightarrow Zebra$: First introduced in CycleGAN [17], this dataset comprises 1067 horse images and 1344 zebra images, resulting in 2403 training images. Additionally, there are 120 horse images and 140 zebra images, totaling 260 test images.

b) $Cat \rightarrow Dog$: First introduced in StarGAN V2 [34], this dataset is a subset of AFHQ and includes 5000 training images and 500 test images for each category.

B. Evaluation Metrics

For the evaluation metric, we selected the Fréchet Inception Distance (FID) [37] score. It is one of the most utilized metrics for assessing the quality of images produced by GANs. The FID score evaluates the distance between feature vectors calculated for real and fake (generated) images, utilizing a pre-trained Inception v3 [38] network. A lower FID score signifies higher similarity to the real images, thus representing better quality of the generated images.

C. Experimental Environment and Baselines

a) Experiment Setting: Our experiments were performed in the Python 3.6.8 environment using the PyTorch framework for all facets of training and testing. The computational tasks were carried out on a system comprising NVIDIA A100-PCI GPUs, each featuring 80 GB of HBM2 memory. The computation tasks were facilitated using CUDA version 12.3 and NVIDIA driver version 545.23.08. All the models were trained for 400 epochs using the Adam optimization algorithm [39], set at a learning rate of 0.0001. During the training of our model, KAN-CUT, we chose $\lambda_{\text{PatchNCE-A}} = 1$ and $\lambda_{\text{PatchNCE-B}} = 1$.

b) Baselines: For our comparative analysis, we selected well-known GAN models as our baselines: CycleGAN [17], MUNIT [40], SelfDistance [41], GCGAN [18], CUT [19], and DCLGAN [20]. It is worth noting that all models were trained in our setup except for MUNIT [40], SelfDistance [41], and GCGAN [18], for which the FID scores were recorded from [20], where DCLGAN was proposed.

D. Results

The quantitative results from each baseline model, including the proposed KAN-CUT, are presented in Table I. It can be seen that KAN-CUT outperforms all the models on both selected datasets, with an FID score of 40.2 on the $Horse \rightarrow Zebra$ dataset, and 59.55 on the $Cat \rightarrow Dog$ dataset.

Additionally, our study presents a visualization of the performance of each model in generating zebra images from horse images and dog images from cat images, as depicted in Fig. 6. As mentioned above, we did not run the experiments for MUNIT [40], SelfDistance [41], and GCGAN [18] in our work and could not find the resulting images generated by these models online. Therefore, the visualization comparison was done against CycleGAN, DCLGAN, and CUT. The KAN-CUT model is able to generate higher quality zebra images based on stripes, structure, and color, compared to all these selected baseline models.

V. Conclusion, Limitation, and Future Work

GANs, being among the most prominent generative models, have played a major role in the success of Image-to-Image (I2I) generation (translation), a subdomain of Generative AI. Despite their success, there are several areas requiring further research, particularly in terms of accuracy, computational efficiency, and knowledge transferability. Focusing primarily on accuracy and demonstrating the efficacy of Kolmogorov-Arnold Networks (KAN) [10] in the image-to-image translation domain, we propose the KAN-CUT model. This novel model replaces the two-layer MLP in the CUT model [19] with a two-layer, efficient, and customized KAN.

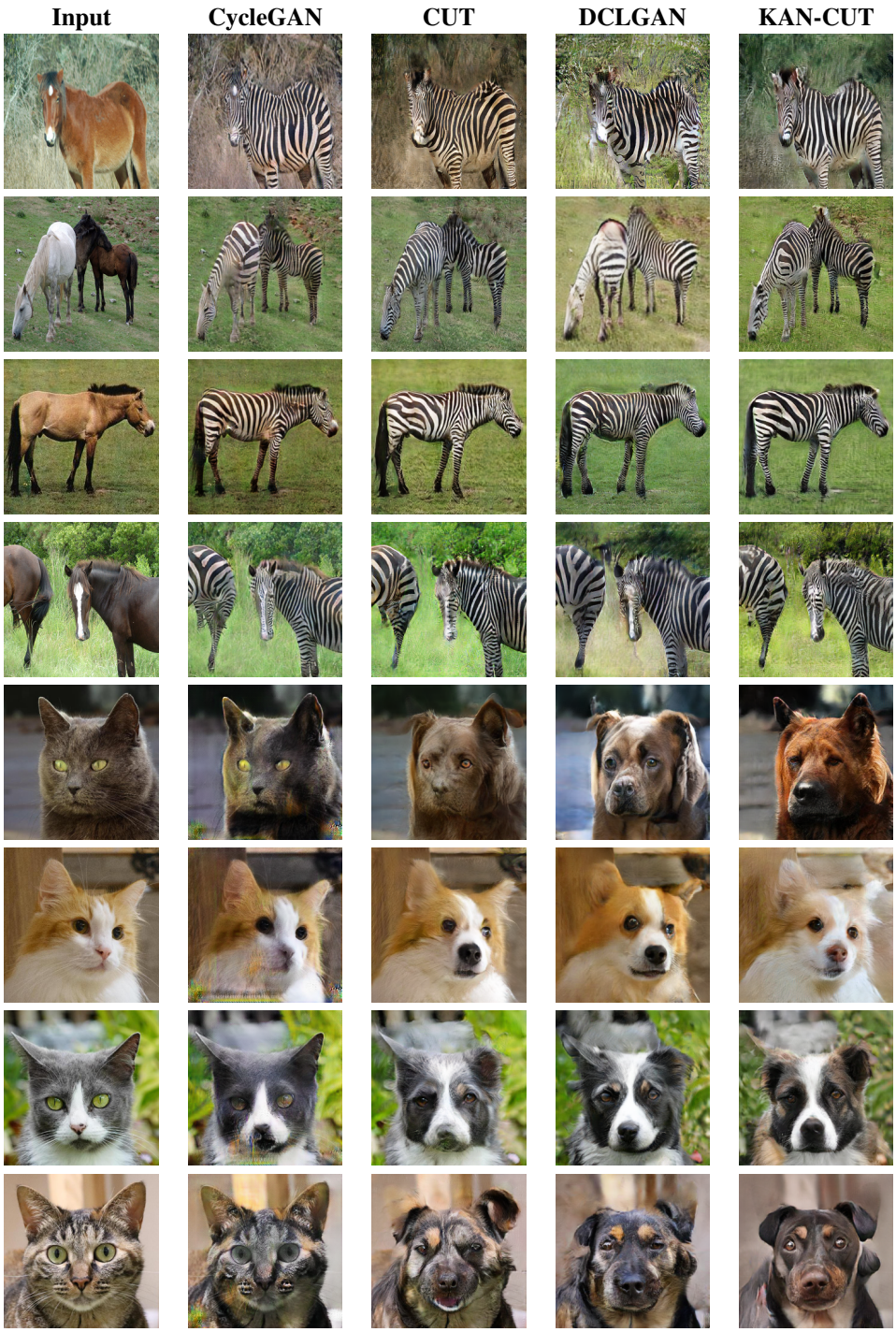

Fig. 6. **Comparative Results of Horse → Zebra and Cat → Dog**. The figure presents a side-by-side comparison of zebra images generated by various models, including KAN-CUT, in the first four rows, and the same comparison of dog images generated by those models in the fifth to eighth rows. The figure highlights the effectiveness of each approach in synthesizing accurate details such as structure and color. Images from the $Horse \rightarrow Zebra$ dataset are sourced from ImageNet [42] and are reproduced under the ImageNet terms of access, which allows use for non-commercial research and educational purposes. Images from the $Cat \rightarrow Dog$ dataset are sourced from the AFHQ dataset [34] and are reproduced under the Creative Commons Attribution-NonCommercial 4.0 International License.

As shown in the Results section, our proposed KAN-CUT model outperforms all selected baseline GAN models on two well-known research datasets, $Horse \rightarrow Zebra$ and $Cat \rightarrow Dog$, in both quantitative and qualitative assessments. Notably, KAN-CUT, being an upgraded version of CUT, achieved a superior FID score by 5.3 points on the $Horse \rightarrow Zebra$ dataset and by 16.65 points on the $Cat \rightarrow Dog$ dataset. To the best of our knowledge, our study is the first to

Method	Horse→Zebra FID ↓	Cat→Dog FID ↓
CycleGAN [17]	66.8	85.9
MUNIT [40]	133.8	104.4
SelfDistance [41]	80.8	144.4
GCGAN [18]	86.7	96.6
CUT [19]	45.5	76.2
DCLGAN [20]	43.2	60.7
KAN-CUT (Ours)	40.2	59.55

TABLE I
COMPARISON OF THE PERFORMANCE OF DIFFERENT GAN MODELS WITH THE PROPOSED MODEL, KAN-CUT, ON THE HORSE→ZEBRA AND CAT→DOG DATASETS. THE PERFORMANCE IS MEASURED USING THE FID SCORE (LOWER IS BETTER).

propose and demonstrate the applicability of KAN in image-to-image translation, potentially paving the way for numerous applications of KAN in Generative AI, which we find very promising.

While the present study has been successful, it has some limitations. Our primary goal was to demonstrate the applicability of KAN [10] and to showcase its improved performance compared to MLP. We do not claim improvement over the state-of-the-art performance in the image-to-image translation domain. Additionally, the KAN-CUT model's performance has been demonstrated on only two datasets, $Horse \rightarrow Zebra$ and $Cat \rightarrow Dog$, due to time and resource constraints. Experimental validation on other datasets remains an area for future work.

ACKNOWLEDGEMENT

This material is based in part upon work supported by the National Science Foundation under Grant Nos. CNS-2018611 and CNS-1920182.

REFERENCES

[1] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[2] A. Ramesh, M. Pavlov, G. Goh, S. Gray, C. Voss, A. Radford, M. Chen, and I. Sutskever, "Zero-shot text-to-image generation," in *International conference on machine learning*. Pmlr, 2021, pp. 8821–8831.

[3] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, and Y. Bengio, "Show, attend and tell: Neural image caption generation with visual attention," in *International conference on machine learning*. PMLR, 2015, pp. 2048–2057.

[4] J. Z. Wu, Y. Ge, X. Wang, S. W. Lei, Y. Gu, Y. Shi, W. Hsu, Y. Shan, X. Qie, and M. Z. Shou, "Tune-a-video: One-shot tuning of image diffusion models for text-to-video generation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 7623–7633.

[5] T.-C. Wang, M.-Y. Liu, J.-Y. Zhu, G. Liu, A. Tao, J. Kautz, and B. Catanzaro, "Video-to-video synthesis," *arXiv preprint arXiv:1808.06601*, 2018.

[6] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1125–1134.

[7] C. M. Bishop, "Neural networks and their applications," *Review of scientific instruments*, vol. 65, no. 6, pp. 1803–1832, 1994.

[8] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural networks*, vol. 2, no. 5, pp. 359–366, 1989.

[9] S. Haykin, *Neural networks: a comprehensive foundation*. Prentice Hall PTR, 1998.

[10] Z. Liu, Y. Wang, S. Vaidya, F. Ruehle, J. Halverson, M. Soljačić, T. Y. Hou, and M. Tegmark, "Kan: Kolmogorov-arnold networks," *arXiv preprint arXiv:2404.19756*, 2024.

[11] A. Kolmogorov, *On the representation of continuous functions of several variables by superpositions of continuous functions of a smaller number of variables*. American Mathematical Society, 1961.

[12] M. Hu and J. Guo, "Facial attribute-controlled sketch-to-image translation with generative adversarial networks," *EURASIP Journal on Image and Video Processing*, vol. 2020, no. 1, p. 2, 2020.

[13] Y. Song and N. Y. Chong, "S-cyclegan: Semantic segmentation enhanced ct-ultrasound image-to-image translation for robotic ultrasonography," *arXiv preprint arXiv:2406.01191*, 2024.

[14] A. Mahara and N. D. Rishe, "Generative adversarial model equipped with contrastive learning in map synthesis," in *Proceedings of the 2024 6th International Conference on Image Processing and Machine Vision*, 2024, pp. 107–114.

[15] Y. Zhao and C. Chen, "Unpaired image-to-image translation via latent energy transport," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 16 418–16 427.

[16] M. Zhao, F. Bao, C. Li, and J. Zhu, "Egsde: Unpaired image-to-image translation via energy-guided stochastic differential equations," *Advances in Neural Information Processing Systems*, vol. 35, pp. 3609–3623, 2022.

[17] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2223–2232.

[18] H. Fu, M. Gong, C. Wang, K. Batmanghelich, K. Zhang, and D. Tao, "Geometry-consistent generative adversarial networks for one-sided unsupervised domain mapping," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 2427–2436.

[19] T. Park, A. A. Efros, R. Zhang, and J.-Y. Zhu, "Contrastive learning for unpaired image-to-image translation," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part IX 16*. Springer, 2020, pp. 319–345.

[20] J. Han, M. Shoeiby, L. Petersson, and M. A. Armin, "Dual contrastive learning for unsupervised image-to-image translation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 746–755.

[21] Y. Choi, M. Choi, M. Kim, J.-W. Ha, S. Kim, and J. Choo, "Stargan: Unified generative adversarial networks for multi-domain image-to-image translation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 8789–8797.

[22] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *International conference on machine learning*. PMLR, 2020, pp. 1597–1607.

[23] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[24] Z. Yi, H. Zhang, P. Tan, and M. Gong, "Dualgan: Unsupervised dual learning for image-to-image translation," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2849–2857.

[25] M. Cheon, "Kolmogorov-arnold network for satellite image classification in remote sensing," *arXiv preprint arXiv:2406.00600*, 2024.

[26] C. Li, X. Liu, W. Li, C. Wang, H. Liu, and Y. Yuan, "U-kan makes strong backbone for medical image segmentation and generation," *arXiv preprint arXiv:2406.02918*, 2024.

[27] C. J. Vaca-Rubio, L. Blanco, R. Pereira, and M. Caus, "Kolmogorov-arnold networks (kans) for time series analysis," *arXiv preprint arXiv:2405.08790*, 2024.

[28] M. Kiamari, M. Kiamari, and B. Krishnamachari, "Gkan: Graph kolmogorov-arnold networks," *arXiv preprint arXiv:2406.06470*, 2024.

[29] L. Van der Maaten and G. Hinton, "Visualizing data using t-sne." *Journal of machine learning research*, vol. 9, no. 11, 2008.

[30] Blealtan, "Efficient kan," https://github.com/Blealtan/efficient-kan, 2024, accessed: April 7, 2024.

[31] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 58, no. 1, pp. 267–288, 1996.

[32] Y. N. Dauphin, A. Fan, M. Auli, and D. Grangier, "Language modeling with gated convolutional networks," in *International conference on machine learning*. PMLR, 2017, pp. 933–941.

[33] A. v. d. Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," *arXiv preprint arXiv:1807.03748*, 2018.

[34] Y. Choi, Y. Uh, J. Yoo, and J.-W. Ha, "Stargan v2: Diverse image synthesis for multiple domains," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 8188–8197.

[35] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," *Advances in neural information processing systems*, vol. 27, 2014.

[36] X. Mao, Q. Li, H. Xie, R. Y. Lau, Z. Wang, and S. Paul Smolley, "Least squares generative adversarial networks," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2794–2802.

[37] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, "Gans trained by a two time-scale update rule converge to a local nash equilibrium," *Advances in neural information processing systems*, vol. 30, 2017.

[38] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2818–2826.

[39] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[40] X. Huang, M.-Y. Liu, S. Belongie, and J. Kautz, "Multimodal unsupervised image-to-image translation," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 172–189.

[41] S. Benaim and L. Wolf, "One-sided unsupervised domain mapping," *Advances in neural information processing systems*, vol. 30, 2017.

[42] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 248–255.

[43] G. Kour and R. Saabne, "Real-time segmentation of on-line handwritten arabic script," in *Frontiers in Handwriting Recognition (ICFHR), 2014 14th International Conference on*. IEEE, 2014, pp. 417–422.

[44] ——, "Fast classification of handwritten on-line arabic characters," in *Soft Computing and Pattern Recognition (SoCPaR), 2014 6th International Conference of*. IEEE, 2014, pp. 312–318.

[45] G. Hadash, E. Kermany, B. Carmeli, O. Lavi, G. Kour, and A. Jacovi, "Estimate and replace: A novel approach to integrating deep neural networks with existing applications," *arXiv preprint arXiv:1804.09028*, 2018.

[46] T. An and C. Joo, "Cycleganas: Differentiable neural architecture search for cyclegan," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 1655–1664.