EAD-GAN: A Generative Adversarial Network for Disentangling Affine Transforms in Images

Letao Liu, Xudong Jiang, *Fellow, IEEE*, Martin Saerbeck, and Justin Dauwels

***Abstract*— This article proposes a generative adversarial network called explicit affine disentangled generative adversarial network (EAD-GAN), which explicitly disentangles affine transform in a self-supervised manner. We propose an affine transform regularizer to force the InfoGAN to have explicit properties of affine transform. To facilitate training an affine transform encoder, we decompose the affine matrix into two separate matrices and infer the explicit transform parameters by the least-squares method. Unlike the existing approaches, representations learned by the proposed EAD-GAN have clear physical meaning, where transforms, such as rotation, horizontal and vertical zooms, skews, and translations, are explicitly learned from training data. Thus, we set different values of each transform parameter individually to generate specifically affine transformed data by the learned network. We show that the proposed EAD-GAN successfully disentangles these attributes on the MNIST, CelebA, and dSprites datasets. EAD-GAN achieves higher disentanglement scores with a large margin compared to the state-of-the-art methods on the dSprites dataset. For example, on the dSprites dataset, EAD-GAN achieves the MIG and DCI score of 0.59 and 0.96 respectively, compared to 0.37 and 0.71, respectively, for the state-of-the-art methods.**

***Index Terms*— Affine transform, disentanglement, generative adversarial network (GAN).**

I. Introduction

GENERATIVE neural models [1], [2] have gained much attention in recent years due to their expressiveness and visualization effect. However, it is preferable for users (e.g., potential designers) to have control over the generated content. To solve this problem, researchers attempted to identify and isolate different attributes in the training data during the generation process.

Many studies have explored the effectiveness of disentangled representations [3]–[7]. The information in the data is encoded in an interpretable and compact manner, e.g., the texture style and the orientation of the objects [3], [5], [6]. The learned representation is generalizable and can be useful for downstream tasks, such as classification and visualization [3], [4], [8].

The concept of disentangled representation has been defined in several ways in the literature [9]–[11]. The necessity of explicit inductive biases both for learning approaches and the datasets is discussed in [9]. Inductive bias refers to a set of assumptions that a learner uses to predict outputs of given inputs that have not been encountered [12], [13]. For instance, in the dSprites dataset, objects are displayed at different angles and positions. Such prior knowledge helps to detect and classify objects. However, the inductive biases in existing disentangled representation approaches are mostly implicit. The explicit affine disentangled generative adversarial network (EAD-GAN) proposed in this article utilizes affine transform as an explicit inductive bias, leading to better disentangled representations with clear physical meaning in terms of affine transforms. Fig. 1 shows entangled representations with unclear physical meaning.

We define the physical meaning property as follows: the absence of physical meaning indicates that experts cannot interpret or map the latent dimensions of disentangled representations to physical or intuitive concepts (e.g., rotation angle), which is a common issue for the representations learned by existing methods [2], [7], [8], [10], [14]–[18]. A disentangled representation usually satisfies two conditions: modularity and compactness [10]. In addition, the representations learned by the EAD-GAN also achieve deterministic assignment property for affine transforms. Modularity measures whether a single latent dimension encodes no more than a single data generative factor. Since some of the latent dimensions of an entangled representation may not have a clear physical meaning, which could be a mixture of several data generative factors and lead to worse modularity. Compactness measures whether each data generative factor is encoded by a single latent dimension. An entangled representation may encode one data generative factor with multiple latent dimensions. On the contrary, each latent dimension learned by the proposed EAD-GAN can be one to one mapped to an affine transform, which leads to both better modularity and compactness for affine transforms.

In a deterministically assigned representation, each latent dimension learns a fixed attribute regardless of the training trials and random seeds. For modularity and compactness, the performance of existing approaches could be improved by utilizing techniques such as contrastive learning [16]. However, a deterministic assignment cannot be achieved by those techniques. For example, if we train an InfoGAN [8] algorithm two times on the MNIST dataset: trials A and B, then in trial A, the first latent dimension may learn the rotation of the digit and the second dimension may learn the thickness of the digit; while in trial B, the first latent dimension may learn the thickness of the digit and the second dimension may learn the rotation of the digit.

Manuscript received 24 July 2021; revised 13 February 2022 and 15 May 2022; accepted 24 July 2022. This work was supported in part by the Singapore Economic Development Board Industrial Postgraduate Program under Grant S17-1298-IPP-II. *(Corresponding author: Letao Liu.)*

Letao Liu and Xudong Jiang are with the Department of Electrical and Electronic Engineering, Nanyang Technological University, Singapore 639798 (e-mail: lliu022@e.ntu.edu.sg; exdjiang@ntu.edu.sg).

Martin Saerbeck is with TÜV SÜD PSB, Singapore 609937 (e-mail: martin.saerbeck@tuvsud.com).

Justin Dauwels is with the Department of Microelectronics, Delft University of Technology, 2628 CD Delft, The Netherlands (e-mail: j.h.g.dauwels@tudelft.nl).

This article has supplementary material provided by the authors and color versions of one or more figures available at https://doi.org/10.1109/TNNLS.2022.3195533.

Digital Object Identifier 10.1109/TNNLS.2022.3195533

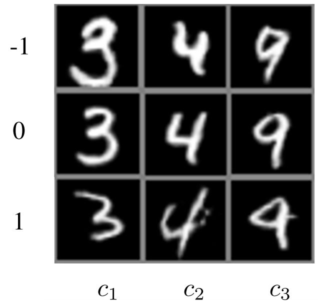

Fig. 1. Illustration of representations generated by InfoGAN [8] with unclear physical meaning. Given different values, −1, 0, and 1, of the latent vector $\mathbf{c} = (c_1, c_2, c_3)$, it is possible that the generated transforms are highly entangled, and thus, they have no clear physical meaning. For example, c_1 may represent both rotation and vertical zoom.

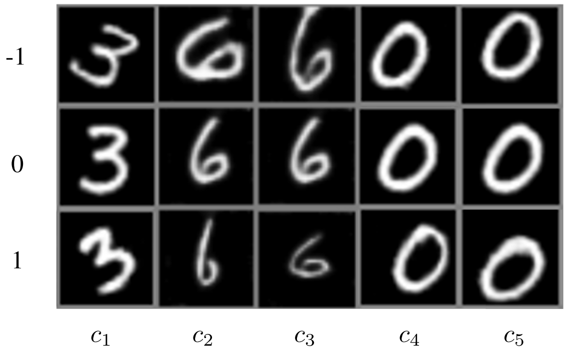

Fig. 2. Given different values, −1, 0, and 1, of the latent vector $\mathbf{c} = (c_1, c_2, c_3, c_4, c_5)$, different versions of images are generated by the proposed EAD-GAN trained on the MNIST dataset. c_1: rotation, c_2 and c_3: horizontal and vertical zooms, and c_4 and c_5: horizontal and vertical translations. Figs. 13–19 show an entire affine transform.

In that situation, to know the attribute assigned to a specific latent dimension for each trial, first, we need to generate a sequence of images (e.g., ten images) by changing the value of that latent dimension (e.g., also known as latent traversal). Then, expert knowledge is required to find the pattern (e.g., rotation of the digit) hidden among the sequence of images. This process could be cumbersome if: i) there are many latent dimensions to observe (e.g., 100 latent dimensions in [7], [14], and [15]) and ii) some sequences of images do not have clear physical meaning. For a disentangled representation with deterministic assignment, the attributes learned by the latent dimensions are fixed. For example, in EAD-GAN, we can predefine the sequence as rotation, horizontal and vertical zooms, and horizontal and vertical translations for latent dimensions 1–5.

A disentangled representation learned by the proposed EAD-GAN can explicitly make a tradeoff between compactness and expressiveness. For example, the zoom attribute can be decomposed into horizontal and vertical zooms. A compact representation encodes the zoom by one latent dimension, while an expressive representation decomposes it into horizontal and vertical, encoded by two latent dimensions. This tradeoff between compactness and expressiveness is beneficial [10], as different subsequent tasks may benefit from different feature decompositions.

We are motivated by the importance of a disentangled representation in particular for the affine transform (see Fig. 2), where disentangling object pose is an attractive property of an algorithm in the imaging domain [19]–[21]. Few algorithms have been able to successfully disentangle the affine transform. In [20], an algorithm is introduced that disentangles rotation and translation but not an entire affine transform. VITAE [22] proposes to separate the spatial transforms from the appearance of the input data, but the spatial transforms themselves are highly entangled in terms of rotation, translation, and zoom.

We propose EAD-GAN, which is a generative adversarial network (GAN) that utilizes the affine regularizer as an inductive bias to explicitly disentangle the affine transform. We assume that every image $\mathbf{X}_r$ is formed by the multiplication of an affine matrix $\mathbf{M}_r$ that describes its pose and a canonical image base $\mathbf{X}_b$. If we purposely transform the image $\mathbf{X}_r$ with a predefined affine matrix $\mathbf{M}$, we obtain another transformed image $\mathbf{X}_t$, where $\mathbf{X}_t$ can also be expressed as the multiplication of an affine matrix $\mathbf{M}_t$ and the same canonical image base $\mathbf{X}_b$. We derive the affine regularizer by decomposing an affine matrix $\mathbf{M}$ into two separate transforms $\mathbf{M}_r$ and $\mathbf{M}_t$ and inferring the transform parameters by the least-squares method. Unlike existing approaches, the representations learned by EAD-GAN are deterministically assigned and have clear physical meaning, where transform, including rotation, horizontal and vertical zooms, and translations, can be explicitly learned from data and hence can be individually selected to generate specific affine transformed data by the learned network (see Fig. 2).

In the remainder of this article, we first review the related work in Section II followed by reviewing InfoGAN and show its limitations in Section III. We introduce the EAD-GAN in Section IV, while in Section V, we show numerical results of the disentangled representation learned by EAD-GAN. We further discuss the advantages and weaknesses of EAD-GAN compared to other methods in Section VI.

Our contributions are given as follows.

1) The disentangled representations obtained by EAD-GAN have clear physical meaning in terms of affine transforms in images. To the best of our knowledge, EAD-GAN is the first algorithm that can disentangle an entire affine transform, including rotation, horizontal and vertical zooms, skews, and translations in an unsupervised manner.
2) The disentangled representations obtained by EAD-GAN have the deterministic assignment property. Each attribute is assigned to a unique component of the latent vector regardless of training trials and vice versa, which achieves better disentangled representations for affine transforms.

II. Related Literature

Recent approaches to learn disentangled representations are largely based on variational autoencoders (VAEs) [2] and InfoGAN [8]. To promote disentanglement, VAE encourages the factorization of the posterior $Q(\mathbf{z}|\mathbf{X})$. InfoGAN [8] proposes to maximize the mutual information between a subset $\mathbf{c}^I$ of latent representation $\mathbf{z}$ and the generated data. Much attention has been paid to regularizers that promote disentanglement. The β-VAE [7] encourages the disentanglement by increasing the weight of the KL regularizer, thus promoting the factorization of the posterior $Q(\mathbf{z}|\mathbf{X})$. Both FactorVAE [14] and β-TCVAE [15] penalize the total correlation, while the former relies on adversarial training and the latter directly calculates the total correlation through the decomposition of the β-VAE objective function. The HFVAE [18] proposes a two-level hierarchical objective to control the relative degree of statistical independence. In the ChyVAE [23], an inverse-Wishart (IW) prior on the covariance matrix of the latent code is augmented to promote statistical independence. The DIP-VAE [24] penalizes the difference between the aggregated posterior and a factorized prior. In the AnnealedVAE [25], the encoder concentrates on learning individual factors and variations by gradually increasing the bottleneck capacity. ControlVAE [26] adds a nonlinear PI controller to automatically tune the hyperparameter added in the VAE objective. Guided-VAE [27] guides the VAE learning by introducing a lightweight decoder that learns latent geometric transformation and principal components. OOGAN [28] improves disentanglement by introducing an orthogonal regularization term to the loss function. In [29], a regularizer is introduced to punish the disagreement between the extracted feature interactions. The IB-GAN [17] is an extension to InfoGAN rooted in the information bottleneck theory, which includes a mutual information upper bound and forms a mutual information bottleneck. The InfoGAN-CR [16] adds a contrastive regularizer on top of InfoGAN that compares the changes between the image and latent space.

Although the aforementioned methods have achieved better disentanglement performance compared to the baseline established by VAE and InfoGAN, none of them yield disentangled representations with deterministic assignments, nor have they successfully disentangled an entire affine transform, which is a desirable property in the imaging domain.

In [30]–[33], self-supervised regularization is applied, where the difference of images before and after the affine/projective transform is compared. The transform loss is defined as: $\mathcal{L} = ||\mathbf{M}(\theta') - \mathbf{M}(\theta)||_2^2$, where $\mathbf{M}$ is a parameterized matrix $\mathbf{M} \in \mathbb{R}^{3\times 3}$. However, those approaches do not achieve disentanglement since no relationship between the data generative factor and the transform is established. By contrast, the proposed EAD-GAN creates a link between the data generative factor and the transform by making a specific definition of each element of the transform matrix and further decomposing it to achieve explicit disentanglement (see Section IV).

As a byproduct of EAD-GAN, the encoder of EAD-GAN can learn the affine transform parameters of a given image and apply the inverse transform to the image to make it invariant to affine transforms. To achieve invariance of affine transforms for image data, spatial transformer network (STN) [19] can actively transform the input images by embedding the spatial transformer block into a target network or algorithm. Inverse compositional spatial transformer networks (IC-STNs) [34] use a recurrent transform manner to further improve the alignment ability of the STN. The intuition behind STN and IC-STN is to fulfill the target network's learning objectives, such as classification or object recognition. Different from STN and IC-STN, the encoder of EAD-GAN is trained in a self-supervised manner and does not need the aid of human-annotated learning objectives such as image classification or object recognition.

III. Background: InfoGAN and Its Limitations

A. GAN: Generative Adversarial Network

GAN [1] trains a deep generative model via a minimax game. The goal is to learn a generated data distribution $P_G(\mathbf{X})$ close to the training data distribution $P_{\text{data}}(\mathbf{X})$ by training a generator and discriminator. During training, first, a latent vector $\mathbf{z}$ is sampled from a prior distribution $P(\mathbf{z})$. Then, the "fake" data $\mathbf{X}_f \sim P_G(\mathbf{X})$ are generated from $\mathbf{z}$ through the generator G. To train the discriminator D, the fake data $\mathbf{X}_f$ are fed to the discriminator D with the label "fake," and the real data $\mathbf{X}_r$ sampled from training data are fed to the discriminator D with the label "real." By contrast, to train the generator G, the fake data $\mathbf{X}_f$ are labeled as "real." The generator G is trained by playing against an adversarial discriminator D that aims to distinguish between samples from the generated data $\mathbf{X}_f \sim P_G(\mathbf{X})$ and the observation $\mathbf{X}_r \sim P_{\text{data}}(\mathbf{X})$ [1]

$$\begin{aligned}\mathcal{L}_{\text{adv}} &= \min_G \max_D V(D, G)\\ &= \mathbb{E}_{\mathbf{X}} \sim P_{\text{data}}[\log D(\mathbf{X})]\\ &\quad + \mathbb{E}_{\mathbf{z}} \sim P(\mathbf{z})[\log(1 - D(G(\mathbf{z})))]. \qquad (1)\end{aligned}$$

B. InfoGAN: Information Maximization GAN

The GAN uses a simple latent vector $\mathbf{z}$ without imposing any constraints on how the generator uses this latent vector, which may lead to a highly entangled mapping between the latent vector $\mathbf{z}$ and the generated data $\mathbf{X}_f$. This is undesirable since there is no intuitive control, i.e., a designer that uses this model would like to generate images with explicit transforms. To overcome this limitation and achieve disentanglement, InfoGAN [8] decomposes the latent vector $\mathbf{z}$ into two parts: $\mathbf{z}'$ representing uncompressible noise and $\mathbf{c}^I$ representing a semantic generative factor (e.g., the number of the generated digit and the rotation of the generated digit in MNIST). In InfoGAN, the mutual information $\mathcal{I}(\mathbf{c}^I; \mathbf{X}_f)$ between the semantic data generative factor $\mathbf{c}^I$ and the generated data $\mathbf{X}_f$ is maximized to promote mapping between $\mathbf{c}^I$ and $\mathbf{X}_f$. Thus, the variation of the generated data $\mathbf{X}_f$ can be reflected by that of the data generative factor $\mathbf{c}^I$. Specifically, InfoGAN maximizes the objective function [8]

$$\mathcal{L}_{\text{Info}} = \mathcal{L}_{\text{adv}} + \lambda \mathcal{I}(\mathbf{c}^I; x_f). \qquad (2)$$

However, InfoGAN achieves the disentanglement in an implicit way since it only uses the mutual information as an inductive bias. This representation has several limitations: 1) the latent vector (data generative factor) $\mathbf{c}^I$ does not necessarily have a clear physical meaning [8], which makes the learned representations difficult to interpret and applicable in downstream tasks, and 2) the modularity and compactness are not optimized and deterministic assignment is not achieved.

IV. Proposed EAD-GAN

To mitigate the aforementioned limitations, i.e., lack of clear physical meaning and deterministic assignment, we aim to equip the disentangled representation with clear physical meaning by adding physical priors as inductive biases. Since disentangling the object pose is an attractive property in the imaging domain, we propose to explore the affine transform as an explicit inductive bias to guide the disentanglement process. We propose a network called EAD-GAN that imposes an affine regularizer in conjunction with InfoGAN. For example, a designer that requires an audience foreground could create one by generating many individuals translated and skewed with EAD-GAN. To derive the affine regularizer, we first introduce the matrix construction process in Section IV-A, where an affine matrix $\mathbf{M}$ is constructed from a latent vector. In Section IV-B, we describe how to decompose a known affine matrix $\mathbf{M}$ into two unknown affine matrices $\mathbf{M}_r$ and $\mathbf{M}_t$.

Next, we estimate the matrices $\hat{\mathbf{M}}_r$ and $\hat{\mathbf{M}}_t$ with a neural network and further compute the matrix $\hat{\mathbf{M}}$. Thus, we can calculate the affine regularizer with $\mathbf{M}$ and $\hat{\mathbf{M}}$. Next, to align each affine transform to an individual latent dimension, we need to estimate each affine parameter from an affine matrix. As explained in Section IV-C, since the estimation process is nonlinear and overdetermined, we apply the LSE to approximate the optimized solution. Finally, we show the consolidated network structure, algorithm flow, and overall loss function in Section IV-D.

A. *Affine Matrix Construction*

To build a connection between the latent vector $\mathbf{c}^I$ (semantic generative factor) and the affine transform, we propose to construct an affine transform matrix $\mathbf{M}$ by a given latent vector. Considering all possible combinations of affine transform (rotation, horizontal and vertical zooms, skews, and translations), there are many ways to construct the affine matrix from a semantic latent vector. As an illustration, here, we select rotation θ, horizontal and vertical zooms (p, q), and translations (x, y) as the components of the affine matrix (see Appendix A in the Supplementary Material for a construction of the entire affine matrix).

Given a latent vector $\mathbf{c}^I = (\mathbf{c}, \mathbf{c}')$, we separate $\mathbf{c}^I$ into $\mathbf{c}$ and another latent vector $\mathbf{c}'$. The latent vector $\mathbf{c}^I$ in InfoGAN encodes various attributes; the component $\mathbf{c}$ of $\mathbf{c}^I$ encodes the affine transform. Given a latent vector $\mathbf{c} = (c_1, c_2, c_3, c_4, c_5)$ randomly sampled from the uniform distribution Unif$[-1, 1]$, we first normalize it to the given range of affine parameters. As an illustration, we set the affine transform range as rotation $\theta \in [-\varepsilon_\theta, \varepsilon_\theta]$, horizontal and vertical zooms $p, q \in [1 - \varepsilon_{pq}, 1 + \varepsilon_{pq}]$, and horizontal and vertical translations $x, y \in [-\varepsilon_{xy}, \varepsilon_{xy}]$. The parameter ε is the multiplier that adjusts the latent vector to a proper affine parameter range. For example, if we want the rotation range to be $[-\pi/10, \pi/10]$, we should set $\varepsilon_\theta = \pi/10$. The affine parameters are computed from the latent vector $\mathbf{c}$ as follows:

$$\begin{aligned} &\theta = c_1\varepsilon_\theta, \quad p = c_2\varepsilon_{pq} + 1, \; q = c_3\varepsilon_{pq} + 1 \\ &x = c_4\varepsilon_{xy}, \quad y = c_5\varepsilon_{xy}. \end{aligned} \tag{3}$$

From those parameters, the affine matrix $\mathbf{M}$ is constructed as in (4) (A_{ij} are the elements of an affine matrix $\mathbf{M}$). For 2-D affine transfom, a 2 × 2 matrix controls the rotation, zooms, and skews of an image. A 2 × 3 matrix adds control over horizontal and vertical translations. We add [0, 0, 1] as the third row for the matrix for the convenience of inverse matrix calculation

$$\begin{aligned} \mathbf{M} &= \begin{bmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ 0 & 0 & 1 \end{bmatrix} \\ &= \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} p & 0 & 0 \\ 0 & q & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & x \\ 0 & 1 & y \\ 0 & 0 & 1 \end{bmatrix}. \end{aligned} \tag{4}$$

B. *Decomposition of Affine Transform*

An affine transform links two images before and after the transform, but an encoder infers the affine transform parameter from a single input image. To let a network learn an affine transform encoder, we propose to decompose an affine transform $\mathbf{M}$ into two parts $\mathbf{M}_r$ and $\mathbf{M}_t$.

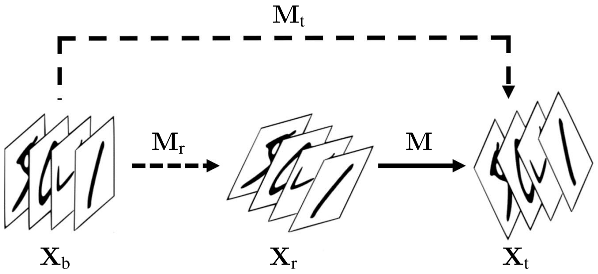

Fig. 3. Decomposition of the affine transform. The solid line refers to the affine transform from a real image $\mathbf{X}_r$ to a transformed image $\mathbf{X}_t$. The dashed lines refer to the affine transform from the canonical base image $\mathbf{X}_b$ to the real image $\mathbf{X}_r$ and to the transformed image $\mathbf{X}_t$.

We represent the spatial coordinates of an image $\mathbf{X}$ by the variables (x, y) and define a column vector $\mathbf{x} = (x, y, 1)^T$. Then, an affine transform of an image $\mathbf{X}$ by a transform matrix $\mathbf{M}$ can be expressed by the matrix multiplication $\mathbf{Mx}$. We express an image $\mathbf{X}_r$ as the combination of an affine matrix $\mathbf{M}_r$ that describes its pose and a canonical image base $\mathbf{X}_b$, $\mathbf{x}_r = \mathbf{M}_r\mathbf{x}_b$. If we purposely transform the image $\mathbf{X}_r$ with a predefined affine matrix $\mathbf{M}$, we obtain another transformed image $\mathbf{X}_t$ from $\mathbf{x}_t = \mathbf{M}\mathbf{x}_r$. Both $\mathbf{X}_r$ and $\mathbf{X}_t$ can be expressed as different transformed versions of the same image $\mathbf{X}_b$, where $\mathbf{x}_r = \mathbf{M}_r\mathbf{x}_b$ and $\mathbf{x}_t = \mathbf{M}_t\mathbf{x}_b = \mathbf{M}\mathbf{M}_r\mathbf{x}_b$ (see Fig. 3). The purpose of introducing canonical image base $\mathbf{X}_b$ is to construct the equation $\mathbf{M}_t\mathbf{x}_b = \mathbf{M}\mathbf{M}_r\mathbf{x}_b$. Once the equation is established, $\mathbf{X}_b$ can be removed from both sides of the equation, and we obtain the relative affine transform equation $\mathbf{M}_t = \mathbf{M}\mathbf{M}_r$. The relative affine transform equation is further used to calculate the affine regularizer.

Thus, from one image, we generate a pair of images $\mathbf{X}_r$ and $\mathbf{X}_t$ for training the transform encoder E (which is equivalent to the auxiliary network Q in InfoGAN). To map the transform from image space to latent space, we encode both $\mathbf{X}_r$ and $\mathbf{X}_t$ to latent vectors $\hat{\mathbf{c}}_r$ and $\hat{\mathbf{c}}_t$ using a learned encoder E. The estimated affine matrices $\hat{\mathbf{M}}_r$ and $\hat{\mathbf{M}}_t$ are then constructed from $\hat{\mathbf{c}}_r$ and $\hat{\mathbf{c}}_t$. The estimated affine matrix $\hat{\mathbf{M}}$ is eventually obtained by $\hat{\mathbf{M}} = \hat{\mathbf{M}}_t\hat{\mathbf{M}}_r^{-1}$ (see Fig. 4).

The base image $\mathbf{X}_b$ does not refer to any particular image, rather a canonical basis of the images from the training dataset (see Fig. 5). It could be the average manifold of all images within the same category. For instance, the digits “0,” “1,”..., “9” in MNIST are different categories. If there are n images of digit “1” with α_i degrees of rotation in the dataset, $\mathbf{X}_b$ could be an image of digit “1” with $\sum_{i=1}^{n}(\alpha_i/n)$ degrees of rotation.

C. *LSE of the Affine Parameter*

Although we can minimize the difference between the ground-truth affine transform matrix $\mathbf{M}$ and its prediction $\hat{\mathbf{M}}$, during training, this does not promote one-to-one mapping between individual affine transform parameters and latent representations $\mathbf{c}$. Thus, we further decompose the predicted affine matrix $\hat{\mathbf{M}}$ into affine parameters $\hat{\theta}$, $\hat{p}$, $\hat{q}$, $\hat{x}$, and $\hat{y}$.

Equation (4) leads to a simultaneous equation group that has six nonlinear equations with five unknowns. Hence, there is no closed-form solution since the equation is nonlinear and overdetermined.

To resolve this problem, we propose to infer the affine parameters from the affine matrix $\hat{\mathbf{M}}$ by the least-squares estimation (LSE). To obtain estimations of the affine parameters,

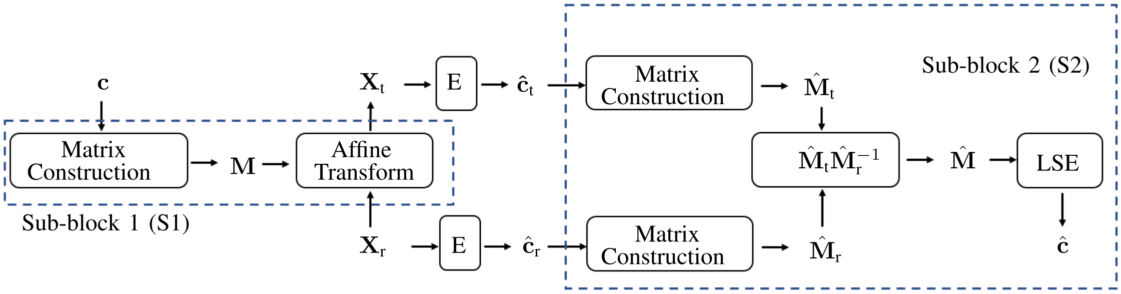

Fig. 4. Pipeline of the affine block. Inputs: latent vector $\mathbf{c}$ randomly sampled from Unif$(-1, 1)$ and image $\mathbf{X}_r$ sampled from training data. Output: transformed image $\mathbf{X}_t$ and predicted latent vector $\hat{\mathbf{c}}$. The affine regularizer loss is: $\mathcal{L}_{\text{affine}} = \min ||\mathbf{c} - \hat{\mathbf{c}}||_2^2$. E stands for encoder. LSE stands for least-squares estimation. Affine transform refers to the operation: $\mathbf{x}_t = \mathbf{M}\mathbf{x}_r$.

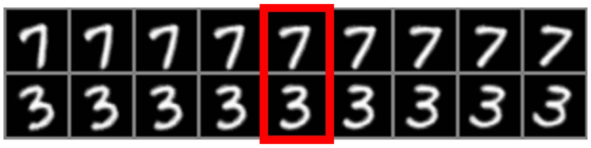

Fig. 5. Illustration of canonical image base $\mathbf{X}_b$. The image in the red box is the $\mathbf{X}_b$ in its respective category.

we minimize the sum

$$\begin{aligned}\mathcal{L}_s(\theta, p, q, x, y) &= \sum_{i=1}^{2}\sum_{j=1}^{3}(A_{ij} - f_{ij}(\theta, p, q, x, y))^2\\ &= (A_{11} - p\cos\theta)^2 + (A_{12} + q\sin\theta)^2\\ &\quad + (A_{21} - p\sin\theta)^2 + (A_{22} - q\cos\theta)^2\\ &\quad + (A_{13} - px\cos\theta + qy\sin\theta)^2\\ &\quad + (A_{23} - px\sin\theta - qy\cos\theta)^2. \qquad (5)\end{aligned}$$

The resulting LSEs are as follows (see more detail in Appendix B in the Supplementary Material):

$$\begin{cases}\hat{\theta} = \dfrac{1}{2}\arctan\dfrac{2(A_{11}A_{21} - A_{12}A_{22})}{A_{11}^2 + A_{22}^2 - A_{12}^2 - A_{21}^2}\\ \hat{p} = A_{11}\cos\hat{\theta} + A_{21}\sin\hat{\theta}\\ \hat{q} = -A_{12}\sin\hat{\theta} + A_{21}\sin\hat{\theta}\\ \hat{x} = \dfrac{A_{13}\cos\hat{\theta} + A_{23}\sin\hat{\theta}}{\hat{p}}\\ \hat{y} = \dfrac{-A_{13}\sin\hat{\theta} + A_{23}\cos\hat{\theta}}{\hat{q}}.\end{cases} \qquad (6)$$

To compare with the ground-truth latent vector $\mathbf{c}$, the estimated affine parameters $\hat{\theta}, \hat{p}, \hat{q}, \hat{x}$, and $\hat{y}$ are converted to a latent vector $\hat{\mathbf{c}}$

$$\begin{aligned}&\hat{c}_1 = \hat{\theta}(1/\varepsilon_\theta), \quad \hat{c}_2 = (\hat{p} - 1)(1/\varepsilon_{pq})\\ &\hat{c}_3 = (\hat{q} - 1)(1/\varepsilon_{pq}), \quad \hat{c}_4 = \hat{x}(1/\varepsilon_{xy}), \quad \hat{c}_5 = \hat{y}(1/\varepsilon_{xy}). \qquad (7)\end{aligned}$$

D. Framework of the Proposed EAD-GAN

The main framework of EAD-GAN is shown in Fig. 6, where the affine block is shown in Fig. 4. Algorithm 1 describes the procedures to compute the affine regularization loss $\mathcal{L}_{\text{affine}} = \min ||\mathbf{c} - \hat{\mathbf{c}}||_2^2$, where $\mathbf{c}$ is the sampled latent vector and $\hat{\mathbf{c}}$ is the estimated latent vector. The loss function of the proposed EAD-GAN is

$$\mathcal{L}_{\text{EAD}} = \mathcal{L}_{\text{adv}} + \lambda\mathcal{I}(\mathbf{c}^I; \mathbf{X}_f) + \beta\mathcal{L}_{\text{affine}}. \qquad (8)$$

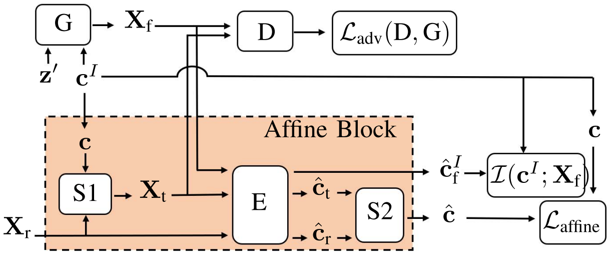
G
$\mathbf{X}_\mathrm{f}$
D
$\mathcal{L}_{\text{adv}}(\mathrm{D}, \mathrm{G})$
$\mathbf{z}'$
$\mathbf{c}^I$
c
Affine Block
c
S1
$\mathbf{X}_\mathrm{t}$
$\mathbf{X}_\mathrm{r}$
E
$\hat{\mathbf{c}}_\mathrm{t}$
$\hat{\mathbf{c}}_\mathrm{r}$
S2
$\hat{\mathbf{c}}_\mathrm{f}^I$
$\mathcal{I}(\mathbf{c}^I; \mathbf{X}_\mathrm{f})$
$\hat{\mathbf{c}}$
$\mathcal{L}_{\text{affine}}$

Fig. 6. Main framework of EAD-GAN. G stands for generator, D stands for discriminator, and E stands for encoder. $\mathbf{X}_f$ is the generated image, $\mathbf{X}_r$ is the image sampled from the training dataset, and $\mathbf{X}_t$ is the affine transformed image from $\mathbf{X}_r$. $\mathbf{z}'$ is the latent noise sampled from the normal distribution. $\mathbf{c}^I = (\mathbf{c}, \mathbf{c}')$ is the sampled semantic latent vector. $\mathbf{c}$ is a subset of $\mathbf{c}^I$ representing affine transform. $\hat{\mathbf{c}}_t$ and $\hat{\mathbf{c}}_r$ are the affine parameter predictions of $\mathbf{X}_t$ and $\mathbf{X}_r$ from the encoder. $\hat{\mathbf{c}}$ is the prediction of $\mathbf{c}$ from the network. $\hat{\mathbf{c}}_f^I$ is the prediction of $\mathbf{X}_f$ from the encoder. $\mathcal{I}(\mathbf{c}^I; \mathbf{X}_f)$ is the mutual information loss. $\mathcal{L}_{\text{adv}}(D, G)$ is the GAN loss. $\mathcal{L}_{\text{affine}}$ is the affine regularization loss. Fig. 4 shows more details about the affine block. S1 stands for Subblock 1. S2 stands for Subblock 2.

The loss function of EAD-GAN only adds one more loss term $\mathcal{L}_{\text{affine}}$ to the loss of InfoGAN, which is easy to implement and computationally efficient. To compute the affine regularizer, the encoder of InfoGAN is reutilized. Hence, the EAD-GAN has the same trainable parameters as InfoGAN. Unlike EAD-GAN, InfoGAN-CR uses an additional encoder to compute the contrastive loss. IB-GAN uses an additional encoder to compute the mutual information upper bound. The proposed affine regularization achieves the following targets: 1) clear physical meaning is assigned to each component of the latent vector $\mathbf{c}$ and 2) each latent dimension in $\mathbf{c}$ is deterministically assigned by constructing the affine matrix from $\mathbf{c}$ and decomposing the affine matrix with the LSE to obtain the estimated $\hat{\mathbf{c}}$. Thus, each latent dimension is motivated to be assigned to a specific affine transform. Besides, by constructing the affine matrix with different combinations of the latent vector, we can flexibly select the desired affine transform. For example, we can construct the horizontal and vertical zooms (p, q) from a single c_1 for a more compact representation or construct p from c_1 and q from c_2 for a more expressive representation.

Compared to InfoGAN, three new components are integrated to the network.

1) The random semantic latent vector $\mathbf{c}$ of EAD-GAN is used to construct affine transform $\mathbf{M}$.
2) Affine transform augmented image $\mathbf{X}_t$ is introduced. While in InfoGAN $\mathbf{X}_r$ is the positive sample fed to the discriminator, in EAD-GAN, we use $\mathbf{X}_t$ as the positive

Algorithm 1 Affine Regularizer

```
Input: training images: $\mathbf{X}_r$, latent vector: $\mathbf{c}$
Output: $\mathcal{L}_{\text{affine}}$
$\mathbf{M}$ = Matrix Construction($\mathbf{c}$) with Eq. 3 and 4
$\mathbf{x}_t = \mathbf{M}\mathbf{x}_r$
$\hat{\mathbf{c}}_t$ = Encoder($\mathbf{X}_t$)
$\hat{\mathbf{c}}_r$ = Encoder($\mathbf{X}_r$)
$\hat{\mathbf{M}}_t$ = Matrix Construction($\hat{\mathbf{c}}_t$) with Eq. 3 and 4
$\hat{\mathbf{M}}_r$ = Matrix Construction($\hat{\mathbf{c}}_r$) with Eq. 3 and 4
$\hat{\mathbf{M}} = \hat{\mathbf{M}}_t\hat{\mathbf{M}}_r^{-1}$
$\hat{\mathbf{c}} = \text{LSE}(\hat{\mathbf{M}})$
$\mathcal{L}_{\text{affine}} = \min ||\mathbf{c} - \hat{\mathbf{c}}||_2^2$
```

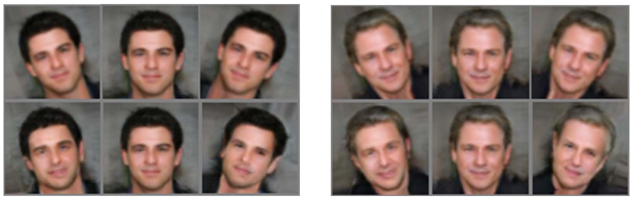

Fig. 7. Ground-truth rotated images and latent traversal with latent vector c_1. Row 1: ground-truth transformed images. The image in the middle is generated by given value 0 to $\mathbf{c}$. The images on the sides are obtained by manually transforming (e.g., rotating) the middle image with the boundary value of the predefined affine transform range (e.g., $[-\pi/9, \pi/9]$). Row 2: latent traversal images: given different values, −1, 0, and 1, of a component c_i (e.g., c_1: rotation) while fixing all other values of the latent vector $\mathbf{c} = (c_1, c_2, c_3, c_4, c_5)$, different versions of images are explicitly generated by the proposed EAD-GAN trained on the CelebA dataset.

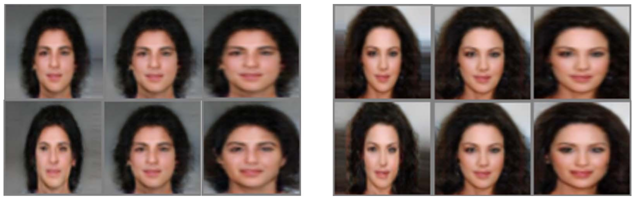

Fig. 8. Ground-truth (row 1) horizontal zoom images and latent traversal (row 2) with latent vector c_2 on the CelebA dataset.

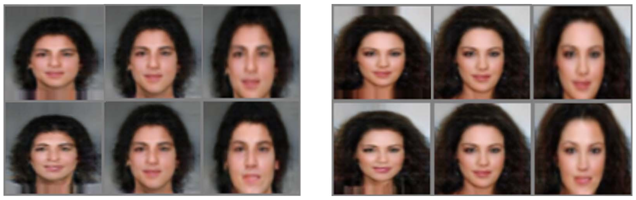

Fig. 9. Ground-truth (row 1) vertical zoom images and latent traversal (row 2) with latent vector c_3 on the CelebA dataset.

sample fed to the discriminator, which guarantees that the affine transform is observed by the network.

3) Affine regularization loss $\mathcal{L}_{\text{affine}}$ is added by comparing the ground-truth latent vector $\mathbf{c}$ and its prediction $\hat{\mathbf{c}}$ from the network. $\mathcal{L}_{\text{affine}}$ builds up the correspondence between the representation learned by InfoGAN and affine transform parameters.

V. Numerical Results

The goal of the experiments in this section is to investigate, both qualitatively and quantitatively, the disentangled representations obtained by EAD-GAN. The datasets evaluated in this section are MNIST, CelebA [35], dSprites [36], and colored dSprites [9]. MNIST contains 60 000 training and 10 000 testing grayscale hand-written digits. CelebA is a more challenging dataset that involves 200 000 RGB celebrity images with large pose variations and background clutter. dSprites is a well-known dataset designed for evaluating the performance of disentangled representations, which contains 737 280 grayscale images with different shapes, scales, orientations, and positions. Colored dSprites adds random RGB color to the object in the dSprites images, where random scaling for each channel uniformly between 0.5 and 1 is multiplied to the object. A major difference between dSprites/colored dSprites and the other aforementioned datasets is that dSprites/colored dSprites contain the ground-truth value for all the variations, making it possible to calculate the disentanglement score. Some sample images generated by the proposed EAD-GAN trained on the CelebA [35], MNIST, and dSprites [36] datasets are shown in Figs. 7–12, in Figs. 13–20, and in Figs. 21–22, respectively. For quantitative results, the disentanglement score for EAD-GAN is presented and compared to benchmarks on the dSprites and colored dSprites datasets (see Tables I and II), while the disentanglement scores for MNIST and CelebA datasets are not presentable due to the lack of ground truth of transform in the dataset. As an alternative, we compare the correspondence between the predefined transform value and the latent vector value predicted by EAD-GAN in Appendix F in the Supplementary Material. We also provide manually transformed images as the ground truth to compare with the latent traversal results (see Figs. 7–12 and 13–20). The parameters of the affine transform are selected as follows: rotation range: $[-\pi/9, \pi/9]$, zoom range: [0.8, 1.2], and translation range: [−0.1, 0.1]. The affine transform range should be small to keep the data distribution of the transformed image x_t close to the training image x_r. For all the experiments, we use the Adam optimizer [37] with the learning rate of 0.0002 for the discriminator and 0.0001 for the generator and encoder. The batch size is 128 for MNIST, 128 for dSprites, and 16 for CelebA. The regularization weights α and β in (8) are set to 1 by default. Our code is available at https://github.com/letao1991/EAD-GAN.

A. Qualitative Results

As mentioned before, deterministic assignment refers to the property that each attribute corresponds to a specific latent dimension. In the CelebA dataset, typical attributes are azimuth, sunglasses, emotion, and so on. Existing methods [8], [14], [15] have successfully disentangled those attributes (see Appendix G in the Supplementary Material). However, other attributes, such as the roll, width, and length of the face and relative position of the face in the frame, are rarely tackled. Due to the deterministic assignment property, EAD-GAN can explicitly learn those attributes (see Figs. 7–12).

We notice that there are some negligible differences between the ground-truth images and the images generated by the EAD-GAN in Figs. 9, 10, and 12. In Figs. 9 and 12, the ground-truth images have the artifacts due to the interpolation effect, while the images generated by EAD-GAN do not have such imperfections. In Fig. 10, for the images generated by the EAD-GAN, the human faces at the sides tend to gaze at the center of the image frame, while the ground-truth images always gaze at the front. This is because GAN tends to generate “realistic” images that are close to the training data distribution. For the human face dataset, most of the human faces at the sides in the training data gaze at the center of

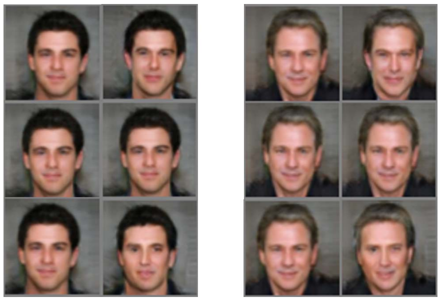

Fig. 10. Ground-truth (left) horizontal translation images and latent traversal (right) with latent vector c_4 on the CelebA dataset.

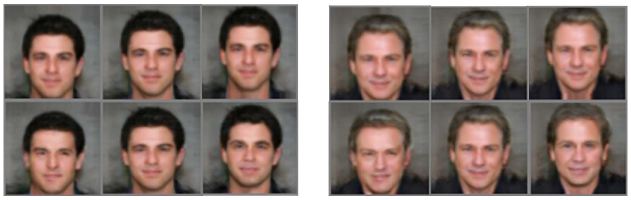

Fig. 11. Ground-truth (row 1) vertical translation images and latent traversal (row 2) with latent vector c_5 on the CelebA dataset.

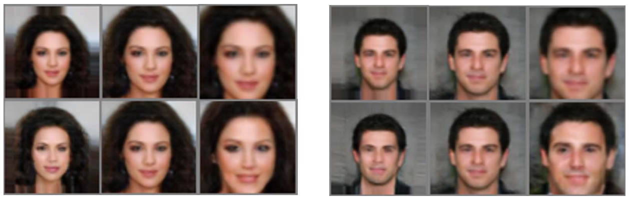

Fig. 12. Ground-truth (row 1) zoom images and latent traversal (row 2) with latent vector c_2 and c_3 on the CelebA dataset.

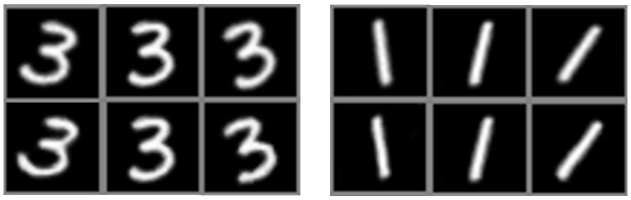

Fig. 13. Ground-truth (row 1) rotation images and latent traversal (row 2) with latent vector c_1 on the MNIST dataset.

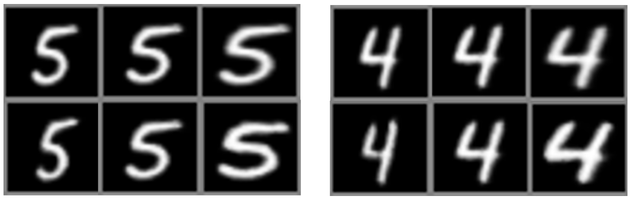

Fig. 14. Ground-truth (row 1) horizontal zoom images and latent traversal (row 2) with latent vector c_2 on the MNIST dataset.

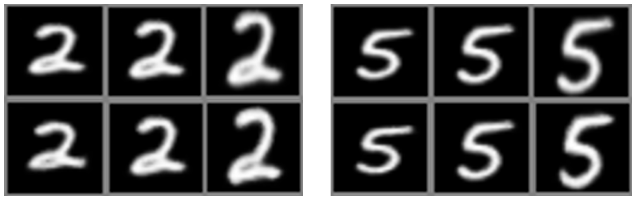

Fig. 15. Ground-truth (row 1) vertical zoom images and latent traversal (row 2) with latent vector c_3 on the MNIST dataset.

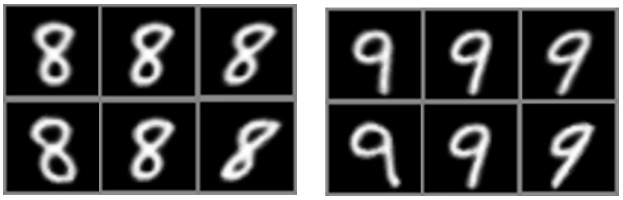

Fig. 16. Ground-truth (row 1) horizontal skew images and latent traversal (row 2) with latent vector c_4 on the MNIST dataset.

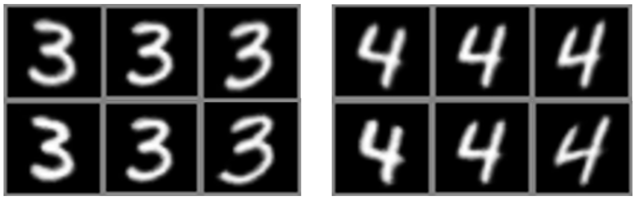

Fig. 17. Ground-truth (row 1) vertical skew images and latent traversal (row 2) with latent vector c_5 on the MNIST dataset.

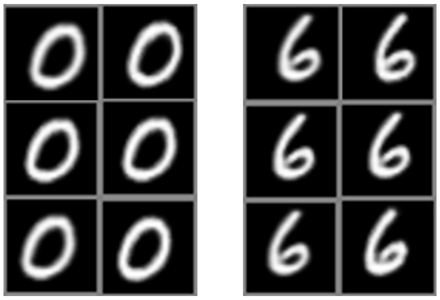

Fig. 18. Ground-truth (row 1) horizontal translation images and latent traversal (row 2) with latent vector c_6 on the MNIST dataset.

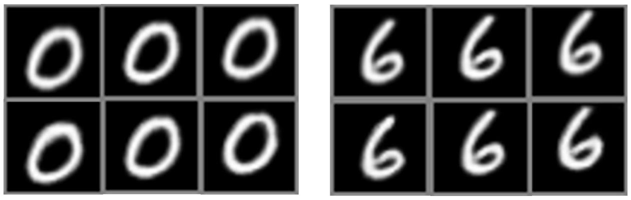

Fig. 19. Ground-truth (row 1) rotation images and latent traversal (row 2) with latent vector c_7 on the MNIST dataset.

the frame (this is also observed in StyleGAN [38]). Overall, the EAD-GAN generates more "natural" images compared to manually transformed images.

Figs. 13–19 show the disentangled representation generated by EAD-GAN with an entire affine transform, which includes rotation, horizontal and vertical zooms, skews, and translations. To the best of our knowledge, EAD-GAN is the first algorithm that can disentangle an entire affine transform in an unsupervised manner (see Appendix A in the Supplementary Material for the construction of the entire affine matrix). In Figs. 14 and 16, we notice that the images generated by the EAD-GAN have larger transform compared to the ground-truth images, and this is because the transform range of the EAD-GAN is the sum of the predefined transform range and the variation of the data distribution. Since the horizontal zoom and skew are the dominant attributes in the MNIST dataset (also observed in InfoGAN), the overall transform range is larger than the predefined transform range.

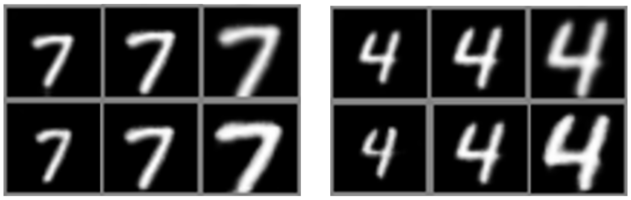

Fig. 20. Ground-truth (row 1) zoom images and latent traversal (row 2) with latent vector c_2 and c_3 on the MNIST dataset.

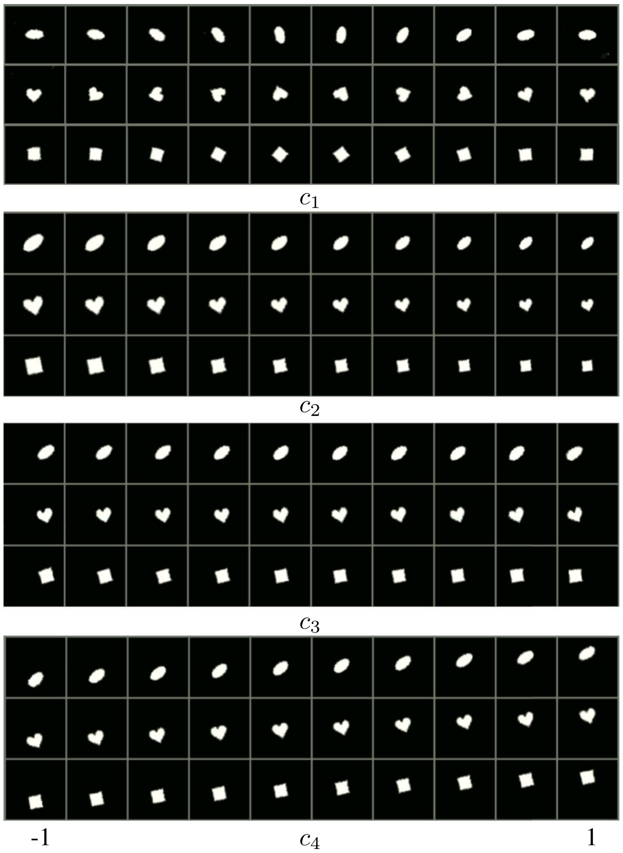

Fig. 21. Given different values in [−1, 1] of a component c_i while fixing all other values of the latent vector $\mathbf{c} = (c_1, c_2, c_3, c_4)$, different versions of images are explicitly generated by the proposed EAD-GAN trained on the dSprites dataset. c_1: rotation, c_2: horizontal and vertical zoom, c_3: horizontal translation, and c_4: vertical translation. The variation by rows is the changing of shape controlled by giving different values, 0, 1, and 2, of the categorical latent vector c_{cat}. Rows 1–3: ellipse, heart, and square, respectively.

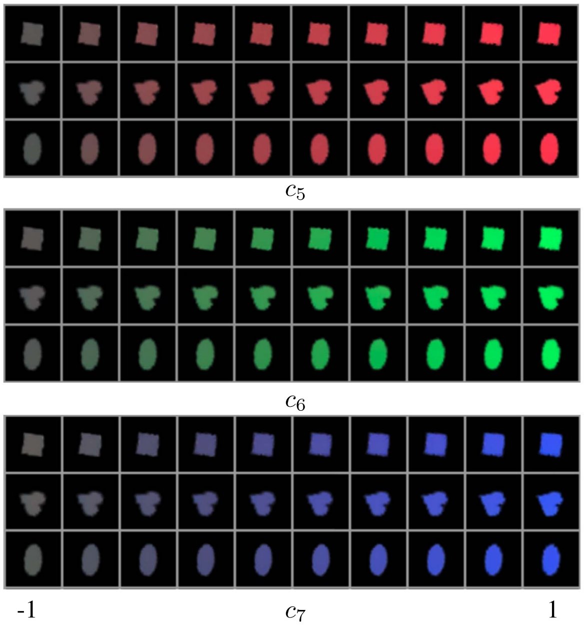

Fig. 22. Given different values in [−1, 1] of a component c_i while fixing all other values of the latent vector $\mathbf{c} = (\ldots, c_5, c_6, c_7)$, different versions of images are explicitly generated by the proposed EAD-GAN trained on the colored dSprites dataset. c_5: red, c_6: green, and c_7: blue. The variation by rows is the changing of shape controlled by giving different values, 0, 1, and 2, of the categorical latent vector c_{cat}. Rows 1–3: square, heart, and ellipse, respectively.

To disentangle object style on dSprites, we choose to model the latent space by a 4-D continuous latent vector sampled from uniform distribution $[-1, 1]$—c_1: rotation, c_2: zoom, and c_3 and c_4: horizontal and vertical translations. We also use a 3-D categorical latent vector c_{cat} (three classes) sampled from a uniform categorical distribution [8] to model the shape attribute. Since the rotation and zoomed-in view dSprites is object centered rather than image frame centered, where the objects are located at random positions, we break the training into two steps. We first train an EAD-GAN network that only learns horizontal and vertical translations and then train another EAD-GAN network that learns all the transforms. A detailed process is described in Appendix C in the Supplementary Material. To the best of our knowledge, EAD-GAN is the first algorithm that can disentangle the shape attribute by means of a categorical latent variable in the dSprites dataset (see Fig. 21), while existing methods [2], [8], [10], [14]–[18], [26]–[28] tend to mix shape and other attributes (e.g., rotation) together (see Appendix D in the Supplementary Material). Note that the affine transform range (e.g., rotation range: $[-\pi/9, \pi/9]$) we set is only for training the encoder. The affine transform range of the image generated by the learned network is determined by the training data distribution (e.g., see Fig. 21 c_1, rotation range for heart: $[0, 2\pi]$, ellipse: $[0, \pi]$, and square: $[0, \pi/2]$).

Besides the affine transform, we show in Appendix E in the Supplementary Material that the RGB color transform can also be explicitly modeled with a similar methodology to our proposed one for the affine transform. To disentangle the object style on colored dSprites, we use a 3-D categorical latent vector c_{cat} (three classes) and a 7-D categorical continuous latent vector: c_1: rotation, c_2: zoom, c_3 and c_4: horizontal and vertical translations, and c_5–c_7: red, green, and blue color transforms. Similar to dSprites, we also break the training into two steps (see Appendix C in the Supplementary Material), where we first train an EAD-GAN network that only learns horizontal and vertical translation, and the RGB color transforms, and then train another EAD-GAN network that learns all the transforms. The disentanglement of color transform for colored dSprites dataset is shown in Fig. 22.

B. Quantitative Results

Tables I and II show that the proposed EAD-GAN outperforms the state-of-the-art methods for all disentanglement

TABLE I

Disentanglement Scores on the dSprites Dataset. For VAE Approaches, the Reference Values From β-VAE to Annealed-VAE Approaches Are the Best Scores of the Violin Plots From [9, Table 13], the Reference Value for Control-VAE Is From [26, Table 2], and the Reference Value for Guided-VAE and Guided-β-TCVAE Are From [27, Table 2]. For GAN Approaches, the Reference Values for GAN, InfoGAN, and IB-GAN Are From [17, Table 1], the Reference Value for GAN-Variation Is From [29, Table 2], the Reference Value for OOGAN Is From [28, Table 1], and the Reference Values for InfoGAN-CR Are From [16, Table 1]. A Perfect Disentanglement Corresponds to a Score of 1.0. The Proposed EAD-GAN Outperforms State-of-the-Art Methods for All Disentanglement Metrics on the dSprites Dataset. The Results of the Proposed EAD-GAN Are the Average of Ten Runs With Random Initialization

	Model	**BetaVAE**	**FactorVAE**	**MIG**	**DCI**	**Modularity**	**SAP**
VAE	β-VAE	0.90	0.82	0.39	0.52	0.91	0.09
	FactorVAE	0.94	0.88	0.34	0.51	0.93	0.08
	β-TCVAE	0.91	0.90	0.38	0.53	0.95	0.09
	DIP-VAE-I	0.87	0.70	0.20	0.23	0.96	0.08
	DIP-VAE-II	0.92	0.78	0.16	0.24	0.93	0.08
	Annealed-VAE	0.88	0.66	0.38	0.41	0.94	0.08
	Control-VAE			0.56 ± .02			
	Guided-VAE		0.67				0.43
	Guided-β-TCVAE		0.73				0.45
GAN	GAN		0.40 ± .05				
	InfoGAN		0.61 ± .03				
	IB-GAN		0.80 ± .07				
	OOGAN		0.81 ± .08				
	GAN-variation		0.88				
	InfoGAN-CR	0.95 ± .01	0.88 ± .01	0.37 ± .01	0.71 ± .01	0.96 ± .00	0.58 ± .01
	EAD-GAN (ours)	**1.0 ± .00**	**0.97 ± .01**	**0.59 ± .01**	**0.96 ± .01**	**1.0 ± .00**	**0.73 ± .01**

TABLE II

Disentanglement Scores on the Colored dSprites Dataset. For VAE Approaches, the Reference Values Are the Best Scores of the Violin Plots From [9, Table 13]. For GAN Approaches, the Reference Values for GAN, InfoGAN, and IB-GAN Are From [17, Table 1]. A Perfect Disentanglement Corresponds to a Score of 1.0. The Proposed EAD-GAN Outperforms State-of-the-Art Methods for All Disentanglement Metrics on the Colored dSprites Dataset. The Results of the Proposed EAD-GAN Are the Average of Ten Runs With Random Initialization

	Model	**BetaVAE**	**FactorVAE**	**MIG**	**DCI**	**Modularity**	**SAP**
VAE	β-VAE	0.90	0.88	0.33	0.49	0.94	0.09
	FactorVAE	0.90	0.9	0.35	0.47	0.92	0.08
	β-TCVAE	0.91	0.91	0.30	0.54	0.96	0.08
	DIP-VAE-I	0.86	0.70	0.18	0.25	0.95	0.08
	DIP-VAE-II	0.92	0.75	0.14	0.21	0.93	0.07
	Annealed-VAE	0.87	0.68	0.38	0.47	0.93	0.08
GAN	GAN		0.35 ± .04				
	InfoGAN		0.55 ± .08				
	IB-GAN		0.79 ± .05				
	EAD-GAN (ours)	**1.0 ± .00**	**0.94 ± .01**	**0.59 ± .01**	**0.90 ± .01**	**1.0 ± .00**	**0.73 ± .01**

metrics on the dSprites and colored dSprites datasets. Both BetaVAE [7] and FactorVAE [14] measure the correlation between the change of the ground-truth attribute and the change of the latent vector predicted from the encoder (we omit “predicted from the encoder” for brevity in the following description). DCI [11] measures the deviation between the latent vector and the ground-truth attribute. SAP [24] measures the average difference of the prediction error of the two most predictive latent dimensions for each attribute. Modularity [39] measures whether each latent dimension conveys information about at most one attribute. MIG [15] measures the mutual information between the latent vector and the ground-truth attribute. The clear physical meaning assigned to the latent vector links the learned representations and the ground-truth attributes. The one-to-one mapping between individual transform parameters and latent dimensions promotes the independence between each latent dimension and avoids the permutation between latent dimensions. The results in Table I suggest that the disentangled representations learned by the proposed EAD-GAN are better aligned with the definition of disentanglement on the dSprites dataset. Compared to InfoGAN-CR [16], which achieves the state-of-the-art disentanglement score, both EAD-GAN and InfoGAN-CR utilize the contrastive learning loss. However, InfoGAN-CR does not explicitly model the affine transforms as the data generative factors.

VI. Discussion

In the literature, several methods have been proposed to learn semantic attributes from data. However, oftentimes, the learned representations do not have a clear physical meaning [2], [6]–[9], [14]–[17]. Moreover, the learned representations of existing methods are sometimes not one-to-one mapped to the interpretable attributes, which makes the learned representations less explainable and inefficient for downstream tasks. Besides, in existing methods, each latent dimension may not learn a fixed attribute in different trials or with different random seeds. To mitigate these problems, the proposed EAD-GAN introduces the affine transform to facilitate the training process, where the physical meaning of the affine transform is transferred and integrated into the network. The qualitative results in Figs. 7–21 show that the proposed EAD-GAN consistently learns the affine transform across different datasets. By comparing the qualitative results on the dSprites dataset between EAD-GAN in Fig. 21 and other methods in Appendix D in the Supplementary Material, we see that EAD-GAN achieves much better disentanglement for affine transform. EAD-GAN achieves the highest disentanglement scores for all the metrics compared to the benchmarks in Tables I and II, which suggest that EAD-GAN is better aligned with the definition of disentanglement. This is consistent with the purpose of modularity and compactness, where each attribute should be assigned to a unique component of the latent vector and vice versa. Several methods have been proposed to promote the independence between each latent component c_i for better disentanglement [7], [14]–[16]. From the perspective of independence, affine transform parameters are intrinsically independent of each other, which may also explain why EAD-GAN achieves the highest disentanglement scores. The resulting affine parameter estimates are accurate, showing that EAD-GAN does not simply memorize the affine transform, as it can extrapolate beyond the the parameter ranges explored during training. However, there are still limitations for EAD-GAN. The component $\mathbf{c}'$ in $\mathbf{c}^I = (\mathbf{c}, \mathbf{c}')$, which is not covered by the affine transform, encoding other information, may lack physical meaning and may not be deterministically assigned.

VII. Conclusion

This article proposes an EAD-GAN that explicitly learns disentangled representations by incorporating an affine transform encoder in the generative model. The encoder learns to represent the affine transform of images by an unsupervised learning procedure. In contrast to the earlier approaches to disentanglement where inductive biases are not explicit, the disentangled representations obtained by EAD-GAN are explicit; as a result, they are deterministically assigned and have clear physical meaning. As the proposed affine regularizer is model-based, it can be extended to include other forms of expert knowledge as inductive bias. Besides affine transform, we show how to explicitly disentangle color transform on the colored dSprites dataset as an illustration. As a possible extension of the 2-D affine transform, the 3-D transform can be learned by constructing and decomposing the 3-D affine transform matrix. The proposed explicit regularizer provides a task-specific pathway to disentanglement compared to the existing general implicit regularizers.

Acknowledgment

The authors would like to thank Dr. Duan Juanyong for the insightful discussion.

References

[1] I. Goodfellow *et al.*, "Generative adversarial nets," in *Proc. NeurIPS*, 2014, pp. 1–9.

[2] D. P. Kingma and M. Welling, "Auto-encoding variational Bayes," 2013, *arXiv:1312.6114*.

[3] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 8, pp. 1798–1828, Aug. 2013.

[4] S. N. Paige *et al.*, "Learning disentangled representations with semi-supervised deep generative models," in *Proc. NeurIPS*, 2017, pp. 1–11.

[5] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436–444, May 2015.

[6] M. Tschannen, O. F. Bachem, and M. Lucic, "Recent advances in autoencoder-based representation learning," in *Proc. NeurIPS*, 2018, pp. 1–25.

[7] I. Higgins *et al.*, "Beta-VAE: Learning basic visual concepts with a constrained variational framework," in *Proc. ICLR*, 2017. [Online]. Available: https://openreview.net/forum?id=Sy2fzU9gl

[8] X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, and P. Abbeel, "InfoGAN: Interpretable representation learning by information maximizing generative adversarial nets," in *Proc. NeurIPS*, 2016, pp. 1–9.

[9] F. Locatello *et al.*, "Challenging common assumptions in the unsupervised learning of disentangled representations," in *Proc. ICML*, 2019, pp. 1–11.

[10] I. Higgins *et al.*, "Towards a definition of disentangled representations," 2018, *arXiv:1812.02230*.

[11] C. Eastwood and C. K. I. Williams, "A framework for the quantitative evaluation of disentangled representations," in *Proc. ICLR*, 2018, pp. 1–15.

[12] T. M. Mitchell, "The need for biases in learning generalizations," Rutgers Univ., New Brunswick, NJ, USA, Tech. Rep. CBM-TR-117, May 1980. [Online]. Available: http://dml.cs.byu.edu/~cgc/docs/mldm_tools/Reading/Need%20for%20Bias.pdf

[13] J. A. Baxter, "A model of inductive bias learning," *J. Artif. Intell. Res.*, vol. 12, no. 1, pp. 149–198, 2000.

[14] H. Kim and A. Mnih, "Disentangling by factorising," in *Proc. ICML*, 2018, pp. 1–10.

[15] R. T. Q. Chen, X. Li, R. B. Grosse, and D. K. Duvenaud, "Isolating sources of disentanglement in variational autoencoders," in *Proc. NeurIPS*, 2018, pp. 1–11.

[16] Z. Lin, K. Thekumparampil, G. Fanti, and S. Oh, "InfoGAN-CR and modelcentrality: Self-supervised model training and selection for disentangling GANs," in *Proc. ICML*, 2020, pp. 1–13.

[17] I. Jeon, W. Lee, M. Pyeon, and G. Kim, "IB-GAN: Disentangled representation learning with information bottleneck generative adversarial networks," in *Proc. AAAI Conf. Artif. Intell.*, May 2021, vol. 35, no. 9, pp. 7926–7934. [Online]. Available: https://ojs.aaai.org/index.php/AAAI/article/view/16967

[18] B. Esmaeili *et al.*, "Structured disentangled representations," in *Proc. 22nd Int. Conf. Artif. Intell. Statist.* (Proceedings of Machine Learning Research). PMLR, 2019, pp. 2525–2534. [Online]. Available: http://proceedings.mlr.press/v89/esmaeili19a.html

[19] M. Jaderberg, K. Simonyan, A. Zisserman, and K. Kavukcuoglu, "Spatial transformer networks," in *Proc. NeurIPS*, 2015, pp. 1–9.

[20] T. Bepler, E. Zhong, K. Kelley, E. Brignole, and B. Berger, "Explicitly disentangling image content from translation and rotation with spatial-VAE," in *Proc. NeurIPS*, 2019, pp. 1–11.

[21] L. Engstrom, B. Tran, D. Tsipras, L. Schmidt, and A. Madry, "Exploring the landscape of spatial robustness," in *Proc. ICML*, 2019, pp. 1802–1811.

[22] N. Skafte and S. R. Hauberg, "Explicit disentanglement of appearance and perspective in generative models," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 32, H. Wallach, H. Larochelle, A. Beygelzimer, F. d' Alché-Buc, E. Fox, and R. Garnett, Eds. Red Hook, NY, USA: Curran Associates, 2019, pp. 1–11. [Online]. Available: https://proceedings.neurips.cc/paper/2019/file/3493894fa4ea036cfc6433c3e2ee63b0-Paper.pdf

[23] A. F. Ansari and H. Soh, "Hyperprior induced unsupervised disentanglement of latent representations," in *Proc. AAAI*, 2019, pp. 1–8.

[24] A. Kumar, P. Sattigeri, and A. Balakrishnan, "Variational inference of disentangled latent concepts from unlabeled observations," in *Proc. ICLR*, 2018, pp. 1–16.

[25] C.-W. Huang, S. Tan, A. Lacoste, and A. C. Courville, "Improving explorability in variational inference with annealed variational objectives," in *Proc. NeurIPS*, 2018, pp. 1–11.

[26] H. Shao *et al.*, "ControlVAE: Controllable variational autoencoder," in *Proc. ICML*, 2020, pp. 8655–8664.

[27] Z. Ding *et al.*, "Guided variational autoencoder for disentanglement learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 7920–7929.

[28] B. Liu, Y. Zhu, Z. Fu, G. de Melo, and A. Elgammal, "OOGAN: Disentangling GAN with one-hot sampling and orthogonal regularization," in *Proc. AAAI Conf. Artif. Intell.*, Apr. 2020, vol. 34, no. 4, pp. 4836–4843. [Online]. Available: https://ojs.aaai.org/index.php/AAAI/article/view/5919

[29] E. Beyazýt, D. Tuncel, X. Yuan, N.-F. Tzeng, and X. Wu, "Learning interpretable representations with informative entanglements," in *Proc. 29th Int. Joint Conf. Artif. Intell.*, Jul. 2020, pp. 1970–1976, doi: 10.24963/ijcai.2020/273.

[30] S. Gidaris, P. Singh, and N. Komodakis, "Unsupervised representation learning by predicting image rotations," in *Proc. ICLR*, 2018, pp. 1–16.

[31] T. Chen, X. Zhai, M. Ritter, M. Lucic, and N. Houlsby, "Self-supervised GANs via auxiliary rotation loss," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 12154–12163.

[32] J. Wang, W. Zhou, G.-J. Qi, Z. Fu, Q. Tian, and H. Li, "Transformation GAN for unsupervised image synthesis and representation learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 472–481.

[33] L. Zhang, G.-J. Qi, L. Wang, and J. Luo, "AET vs. AED: Unsupervised representation learning by auto-encoding transformations rather than data," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 2547–2555.

[34] C.-H. Lin and S. Lucey, "Inverse compositional spatial transformer networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2568–2576.

[35] Z. Liu, P. Luo, X. Wang, and X. Tang, "Deep learning face attributes in the wild," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 3730–3738.

[36] L. Matthey, I. Higgins, D. Hassabis, and A. Lerchner, "dSprites: Disentanglement testing sprites dataset," DeepMind, London, U.K., Tech. Rep., 2017. [Online]. Available: https://github.com/deepmind/dsprites-dataset

[37] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. 3rd Int. Conf. Learn. Represent. (ICLR)*, Y. Bengio and Y. LeCun, Eds. San Diego, CA, USA: ICLR, May 2015, pp. 1–10.

[38] T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, and T. Aila, "Analyzing and improving the image quality of styleGAN," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 8110–8119.

[39] K. Ridgeway and M. C. Mozer, "Learning deep disentangled embeddings with the F-statistic loss," in *Proc. NeurIPS*, 2018, pp. 1–10.

Letao Liu received the B.S. degree in electrical and electronic engineering from Nanyang Technological University, Singapore, in 2014, where he is currently pursuing the Ph.D. degree in electrical and electronic engineering.

From 2017 to 2021, he was an Artificial Intelligence Engineer with TÜV SÜD PSB, Singapore. His research interests include representational learning with self-supervised transformation, interpretability of deep learning, disentangled representation with generative adversarial networks, and computer vision-based interactive robot system design.

Mr. Liu's awards and honors include the Senior Middle School 2 Scholarship (SM2) and the Industrial Postgraduate Program Scholarship (IPP).

Xudong Jiang (Fellow, IEEE) received the B.Eng. and M.Eng. degrees from the University of Electronic Science and Technology of China (UESTC), Chengdu, China, in 1983 and 1986, respectively, and the Ph.D. degree from Helmut Schmidt University, Hamburg, Germany, in 1997.

He was a Lecturer with UESTC, where he received two Science and Technology Awards from the Ministry for Electronic Industry. He was with the I2R, A*STAR, Singapore, as a Lead Scientist and the Head of the Biometrics Laboratory, where he achieved the most efficiency and the second most accuracy at the International Fingerprint Verification Competition in 2000. He joined Nanyang Technological University (NTU), Singapore, as a Faculty Member in 2004, where he was the Director of the Centre for Information Security. He is currently a Professor with NTU. He holds seven patents and has authored over 200 papers with over 40 articles in IEEE journals: IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, IEEE TRANSACTIONS ON IMAGE PROCESSING, IEEE TRANSACTIONS ON SIGNAL PROCESSING, IEEE TRANSACTIONS ON MULTIMEDIA, IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY, Signal Processing Magazine (SPM), and Signal Processing Letter (SPL). His research interests include image processing, pattern recognition, computer vision, machine learning, and biometrics.

Dr. Jiang was an IFS TC Member of IEEE Signal Processing Society and an Associate Editor of IEEE SIGNAL PROCESSING LETTERS, IEEE TRANSACTIONS ON IMAGE PROCESSING, and *IET Biometrics*. He is a Senior Area Editor of IEEE TRANSACTIONS ON IMAGE PROCESSING and the Editor-in-Chief of *IET Biometrics*.

Martin Saerbeck received the master's degree in computer science from Bielefeld University, Bielefeld, Germany, in 2005, and the Ph.D. degree from the Eindhoven University of Technology, Eindhoven, The Netherlands, in 2010.

From 2010 to 2013, he was the Project Leader of the Social Situation Awareness Project at the Institute for High-Performance Computing, A*STAR, Singapore. From 2013 to 2017, he was the Capability Group Manager and a Senior Scientist at the Institute for High-Performance Computing, A*STAR. He joined TÜV SÜD PSB, Singapore, in 2017, where he is currently the Chief Technology Officer. His research interest includes human–machine interaction, system verification, software architectures, robotics, user perception and usability, interactive system design, and embedded systems.

Justin Dauwels received the master's degree in engineering physics from the University of Ghent, Ghent, Belgium, in 2000, and the Ph.D. degree in electrical engineering from the Swiss Federal Institutes of Technology (ETH), Zürich, Switzerland, in December 2005.

From 2006 to 2007, he held a post-doctoral position at the RIKEN Brain Science Institute, Saitama, Japan (Prof. Shun-ichi Amari and Prof. Andrzej Cichocki). From 2008 to 2010, he was a Research Scientist with the Stochastic Systems Group (SSG), Massachusetts Institute of Technology (MIT), Cambridge, MA, USA, led by Prof. Alan Willsky. He was an Associate Professor with the School of Electrical and Electronic Engineering, Nanyang Technological University (NTU), Singapore, in 2010. He starts in January 2021 as an Associate Professor at TU Delft, Delft, The Netherlands. His research interests are in Bayesian statistics, iterative signal processing, and computational neuroscience. He enjoys working on real-world problems, often in collaboration with medical practitioners. He also tries to bring real-world problems into the classroom. Research from his team is featured at the Singapore Science Center, Singapore, and in the Straits Times.