ChildPredictor: A Child Face Prediction Framework with Disentangled Learning

Yuzhi Zhao, *Student Member, IEEE,* Lai-Man Po, *Senior Member, IEEE,* Xuehui Wang, Qiong Yan, Wei Shen, Yujia Zhang, Wei Liu, Chun-Kit Wong, Chiu-Sing Pang, Weifeng Ou, Wing-Yin Yu, Buhua Liu

***Abstract*—The appearances of children are inherited from their parents, which makes it feasible to predict them. Predicting realistic children's faces may help settle many social problems, such as age-invariant face recognition, kinship verification, and missing child identification. It can be regarded as an image-to-image translation task. Existing approaches usually assume domain information in the image-to-image translation can be interpreted by "style", i.e., the separation of image content and style. However, such separation is improper for the child face prediction, because the facial contours between children and parents are not the same. To address this issue, we propose a new disentangled learning strategy for children's face prediction. We assume that children's faces are determined by genetic factors (compact family features, e.g., face contour), external factors (facial attributes irrelevant to prediction, such as moustaches and glasses), and variety factors (individual properties for each child). On this basis, we formulate predictions as a mapping from parents' genetic factors to children's genetic factors, and disentangle them from external and variety factors. In order to obtain accurate genetic factors and perform the mapping, we propose a ChildPredictor framework. It transfers human faces to genetic factors by encoders and back by generators. Then, it learns the relationship between the genetic factors of parents and children through a mapping function. To ensure the generated faces are realistic, we collect a large Family Face Database to train ChildPredictor and evaluate it on the FF-Database validation set. Experimental results demonstrate that ChildPredictor is superior to other well-known image-to-image translation methods in predicting realistic and diverse child faces. Implementation codes can be found at https://github.com/zhaoyuzhi/ChildPredictor.**

***Index Terms*—Child Face Prediction, Disentangled Learning, Generative Adversarial Network, Image-to-image Translation.**

Manuscript received December 6, 2021; Revised February 24, 2022 and March 29, 2022; accepted March 31, 2022; date of current version March 31, 2022. This article was recommended by Associate Editor Chang Xu. *(Corresponding author: Yuzhi Zhao.)*

Y. Zhao, L.-M. Po, Y. Zhang, C.-K. Wong, C.-S. Pang, W. Ou, W.-Y. Yu are with the Department of Electrical Engineering, City University of Hong Kong, Hong Kong, China (e-mail: yzzhao2-c@my.cityu.edu.hk; eelmpo@cityu.edu.hk; yzhang2383-c@my.cityu.edu.hk; ckwong535-c@my.cityu.edu.hk; chiuspang2-c@my.cityu.edu.hk; weifengou2-c@my.cityu.edu.hk; wingyinyu8-c@my.cityu.edu.hk).

X. Wang, W. Shen are with the Artificial Intelligence Institute, Shanghai Jiao Tong University, Shanghai, China (e-mail: wangxuehui@sjtu.edu.cn; wei.shen@sjtu.edu.cn).

Q. Yan is with the SenseTime Research and Tetras.AI, Hong Kong, China. (e-mail: sophie.yanqiong@gmail.com).

W. Liu is with the ByteDance Ltd., Beijing, China (e-mail: liu-jikun63@gmail.com)

B. Liu is with the Department of Computer Science, Hong Kong Baptist University, Hong Kong, China (e-mail: csbhliu@comp.hkbu.edu.hk)

This article has supplementary material provided by the authors and color versions of one or more figures available at https://doi.org/10.1109/TMM.2022.3333333.

Digital Object Identifier 10.1109/TMM.2022.3333333

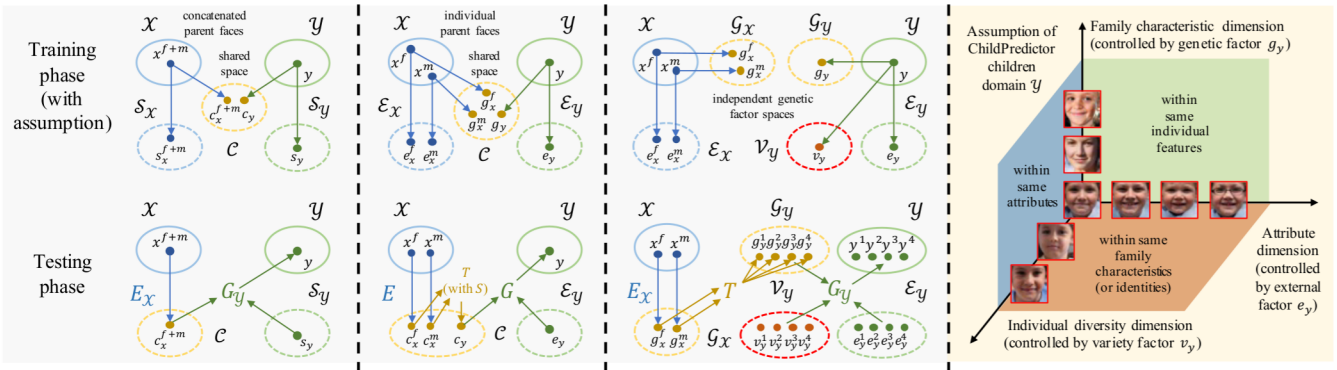
I. Introduction

THE appearances of children are inherited from parents. Their internal relations (e.g., kinship verification and identification) have been well studied [1]–[5]. It provides the prerequisite for predicting child faces from their parents. Predicting realistic child faces are beneficial to many social issues such as law enforcement, criminal investigations, age-invariant face recognition [6]–[9], kinship verification [1]–[5], kinship identification [10], and missing children identification [11], especially under the circumstances that *only parent faces are known*. Recently, the generative adversarial network (GAN) [12] has shown its advance in the face generation area. If we treat child face prediction as an image- to-image translation issue, there is a potential for GAN to predict high-quality child faces. In this paper, we propose a GAN-based ChildPredictor framework for realistic child face prediction.

The fundamental difficulty of child face prediction lies in the requirement on both *diversity* and *similarity* at the same time, in addition to *image quality*. Conditioned on parent faces, the predicted child faces need to be similar to real faces while retaining diversity. The existing strategies to address the issues fall into the two categories:

1) Image-to-image translation (I2I) (Figure 1 (a)): Assuming parents' and children's faces form two individual domains, and predicting child faces by transferring "style" from parent domain to children domain;
2) DNA-Net [13] (Figure 1 (b)): Learning the direct mapping between parents' and children's features, which are generated by the same pre-trained encoder. Diverse prediction is implemented by random selection S.

Unfortunately, these methods have difficulties in predicting pleasant faces. Firstly, state-of-the-art I2I methods [14]–[16] proposed a shared content space and individual style spaces to improve image translation quality. If simply applying it to child face prediction, though the paired parent-child data is used for training I2I methods, they easily fall into "appearance collapse" (e.g., the generated children have the same facial structures and contours with parents). We assume that the disentanglement of content (i.e., structure or shape) and style (i.e., texture) is not reasonable for this task since different children could have similar structures with parent faces but not the same. Secondly, DNA-Net fused features of parents and then changed them using an age regression model [17]. Since parent and child faces share the same latent space, it restricts the feature representation and reality. Though it applied a random selection S to combine mothers' and fathers' features,



(a) I2I: MUNIT, DRIT, DRIT++ (b) DNA-Net (c) ChildPredictor (d) Assumption of ChildPredictor

Fig. 1. Assumption and data flow: (a) I2I methods [14]–[16]; (b) DNA-Net [13]; (c), (d) ChildPredictor. The $\mathcal{X}$ and $\mathcal{Y}$ represent parent and children domain, respectively. The $\mathcal{C}$, $\mathcal{S}$, $\mathcal{E}$, $\mathcal{V}$, $\mathcal{G}$ are content, style, external, variety (or attribute for (b)) and genetic latent domains, respectively. The network E and G are pair of encoder and generator that connect image space and latent space. The network T is a mapping function used in the latent space.

the framework is trained in a unimodal fashion. Therefore, the results are not diverse enough.

This paper presents a novel framework for synthesizing realistic and diverse child faces. Built upon previous experiences, we first assume a human face is determined by a *genetic factor, external factor, and variety factor*, among which a child's genetic factor is predicted from parents' genetic factors. These three factors are trained to be *orthogonal*, as shown in Figure 1 (d), and detailed definitions are concluded as:

1) Genetic factor g: Family characteristics such as contour, eye shape, pupil color, skin color, etc., which model the inter-identity variation between different identities;
2) External factor e: Categorical and gene-irrelevant attributes, e.g., moustaches, ages, expressions, glasses;
3) Variety factor v: Individual diversities for an identity. It is not correlated to genetic and external factors, which models acquired factors (intra-identity variations).

Based on the assumption, we propose a GAN-based framework called ChildPredictor to predict child faces from their parents. The data flow of ChildPredictor is shown in Figure 1 (c). There are two sequential steps in the training phase compared with previous one-step pipelines:

1) Domain-specific disentangled learning: Learning disentangled representations of different factors for both domains, i.e., the disentanglement of $\mathcal{G}_\mathcal{X}/\mathcal{E}_\mathcal{X}$ and $\mathcal{G}_\mathcal{Y}/\mathcal{E}_\mathcal{Y}/\mathcal{V}_\mathcal{Y}$ (please see the caption of Figure 1 and the training phase of Figure 1 (c) for their definitions), respectively;
2) Inter-domain multimodal mapping: Predicting multiple children's genetic factors between the disentangled genetic factors in both spaces $\mathcal{G}_\mathcal{X}$ and $\mathcal{G}_\mathcal{Y}$.

There are four advances of our setting: 1) Compared with the general content-style separation [15], the proposed genetic-external-variety disentangled learning, which is a three-way decomposition specially designed for the task, considers inter-identity variation, gene-irrelevant attributes, and intra-identity variation; 2) Since the data distribution of parents' and children's faces are quite different, training in each domain can lead to better domain-specific features than joint training [13]; 3) Since domain-specific training is not based on limited parent-child pairs, additional data can be used, e.g., FFHQ child faces [18] are used to improve the generation quality of $\mathcal{G}_\mathcal{Y}$; 4) Our mapping function learns a multimodal prediction instead of unimodal [13] thus it is able to simulate different child identities from the same parent. In summary, our ChildPredictor framework largely improves child face prediction quality in similarity, diversity, and realness.

In addition, to train ChildPredictor, we newly collect a large-scale FF-Database from the Internet. It includes 7488 parent faces and 8558 child faces with 128×128 resolution and even gender distribution. Each face is aligned to be almost frontal and labeled with 6 attributes which are gender, age, expression, glasses, moustache, and skin color. For evaluation, we notice that common metrics (e.g., PSNR) fail to measure face similarity since the generated and ground truth faces are not pixel-wisely aligned. We use a cosine similarity metric to compute feature distances from a pre-trained face recognition model and conduct a human perceptual study for subjective evaluation. The experiment results on the FF-Database validation set demonstrate that ChildPredictor performs better than previous pipelines [13]–[16], [19]–[21] on cosine similarity, FID [22], LPIPS [23] scores and human preference rates.

The main contributions of this paper are as follows:

1) We propose a GAN-based ChildPredictor framework specializing in the task of child face prediction;
2) We propose a latent representation disentangled learning method by sampling from three latent factors;
3) We newly collect a large-scale Family Face Database (FF-Database). To our best knowledge, it is the first dataset for child face prediction with labeled attributes;
4) We propose a cosine similarity metric and a human perceptual study for evaluating paired predicted and real faces. The ChildPredictor achieves the best performance.

II. Related Work

Generative Adversarial Network (GAN) and Inversion. The GAN [12] has accomplished great improvements in image generation. It consists of a generator and a discriminator, where the generator produces realistic samples and the discriminator distinguishes input samples are from ground truth or generated. However, GAN is hard to converge and unstable. To address the issue, some methods minimized mode collapse

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2022.3164785, IEEE Transactions on Multimedia

[24] and loss fluctuation [25], or proposed new architectures [18], [26]–[28]. To discover the relation between generated images and latent space, GAN inversion techniques have been widely studied. For instance, [29] and [30] learned to search interpretable directions in GAN's latent space. To edit images accurately, inverse encoders [31]–[37] were proposed to simulate reverse process of GANs. They learned the inversion in original latent space or extend W+ space.

Face Attribute Transfer. The previous face attribute transfer methods are normally based on paired data [38]. However, it is hard to collect many different attributes for the same person. To avoid that, researchers used several representations such as pre-defined attributes [39]–[43], facial landmarks [44], face mask [45], and action unit [46]. These methods were further enhanced by processing multiple attributes simultaneously [47]–[49]. More recently, the GAN inversion techniques [50], [51] and latent disentangled learning [52]–[54] have been used in the face attribute transfer area. For instance, Nitzan et al. [50] transferred attributes based on an exemplar face by manipulating pre-trained GAN's latent space.

Image-to-Image Translation (I2I) and Disentangled Learning. The I2I denotes the mappings of images from one domain to other domains. For instance, Pix2Pix [55] used a conditional GAN to perform domain transfer on pixel-aligned data, which is necessary, otherwise, the results are blurry. To extend I2I to unaligned data, some approaches have been developed, e.g., enlarging the distance between generated samples and source [56], [57] and cycle consistency [19], [20], [58]. However, the models often fail (i.e., mode collapse) when there exist extreme transformations or training data is limited.

To improve the image translation quality, the disentangled learning [14]–[16], [21], [59]–[67] assumed that images from individual domains share the same latent content space but separate style space. By changing "style code", the network transfers input images to other domains while maintaining the content information. To further address the mode collapse issue, [24] proposed a *mode-seeking loss* to enlarge distances of different generated samples for regularization. However, such disentangled learning methods are not appropriate for child face prediction since the goal is not only to transfer the style but also to consider facial attributes and intra-identity variations. To address the issue, we propose a new disentangled learning method based on genetic, external, and variety factors (please see Figure 1 (c) and (d) for their definitions).

III. Data Collection of FF-Database

We collect a large-scale Family Face Database (FF-Database), consisting of 16046 images with 128×128 resolution. Built upon it, we learn the child prediction in a data-driven manner. There are 4 steps to collect a group of images in FF-Database, as shown in Figure 2: 1) Downloading family images by country or district names in 6 continents from the Internet, and filtering out unrelated face images; 2) Extracting faces by dlib[1] and aligning them to be almost frontal; 3) Enhancing faces by denoising [68], inpainting [69],

[1]http://dlib.net/

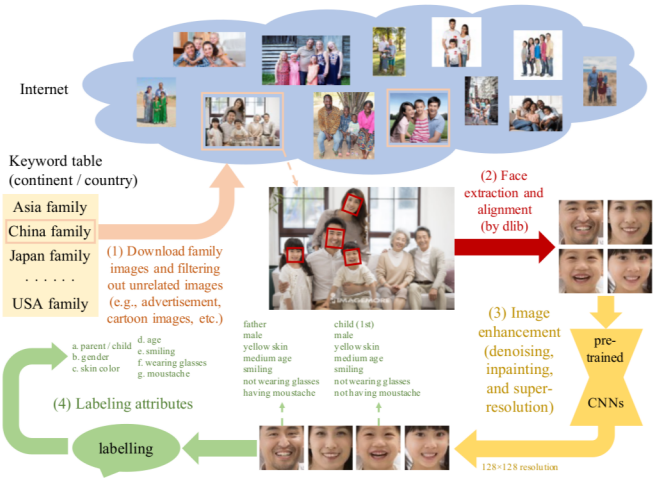


Fig. 2. Illustration of image collection workflow for the FF-Database.

TABLE I
Conclusion of the labeled facial attributes of the FF-Database training and validation sets.

Tag	Attribute	Parent faces			Child faces		
		True	False	Total	True	False	Total
Train	Male	3574	3574	7148	4104	4086	8190
Train	Young	804	6344	7148	1043	7147	8190
Train	Smile	6384	764	7148	5860	2330	8190
Train	Glass	572	6576	7148	279	7911	8190
Train	Moustache	1281	5867	7148	54	8136	8190
Val	Male	170	170	340	192	176	368
Val	Young	8	332	340	35	333	368
Val	Smile	334	6	340	352	16	368
Val	Glass	12	328	340	6	362	368
Val	Moustache	53	287	340	0	368	368

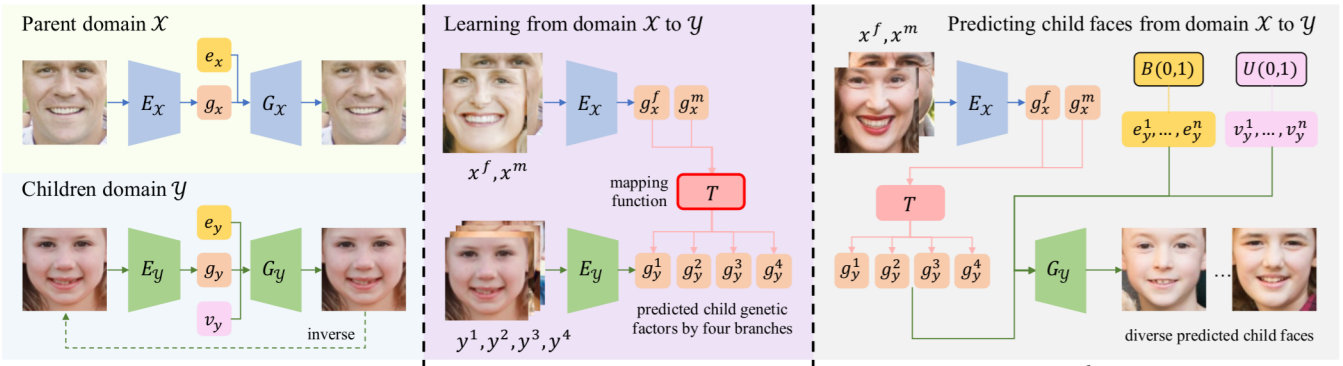
super-resolution [70] algorithms, and resizing to 128×128 resolution. Note that the very low-quality images are discarded by human effort; 4) Labeling them with 6 attributes including gender, age, expression, glasses, moustache, and skin color.

We divide the whole dataset into two parts, where the training set includes 15538 faces and the validation set includes 708 faces. Specifically, there are 7148 parents and 8190 children in the training set; and there are 340 and 368 faces in the validation set, respectively. The attributes and the division of training and validation sets are concluded in Table I, respectively.

IV. Methodology

A. Problem Formulation

Given paired parent faces $x^f, x^m \in \mathcal{X}$ and a child face $y \in \mathcal{Y}$, the target of child face prediction is to learn $p(y|x^f, x^m)$. Note that their corresponding genetic factors g_y, g_x^f, g_x^m are the compact and simplified representations of face images by our definition. Therefore, it is more tractable to solve the problem in the genetic domain, i.e., to learn $p(g_y|g_x^f, g_x^m)$, only if the genetic factors and faces are transferable from each other. To achieve this, we design the two-stage framework, **domain-specific disentangled learning** which learns to extract the genetic factors from faces and restore them to faces for parent and children domain separately, and **inter-domain multimodel mapping** which maps parent genetic factors to children domain.



(a) Domain-specific disentangled training (b) Inter-domain multimodal mapping (c) Inference pipeline (from x^f, x^m to diverse child faces)

Fig. 3. Illustration of (a) self-supervised disentangled learning (1st training step); (b) inter-domain multimodal mapping (2nd training step); (c) inference pipeline. The latent codes g_*, e_*, $v_\mathcal{Y}$ represent genetic, external and variety factors.

TABLE II
LIST OF ALL THE NOTATIONS USED IN THE PAPER.

Notation	Definition
$D_\mathcal{X}$	Discriminator for parent domain $\mathcal{X}$
$D_{\mathcal{G}_\mathcal{X}}$	Discriminator for parent genetic factor distribution $\mathcal{G}_\mathcal{X}$
$D_\mathcal{Y}$	Discriminator for children domain $\mathcal{Y}$
e_x^f	External factor of father face image x^f
e_x^m	External factor of mother face image x^m
$\mathcal{E}_\mathcal{X}$	Distribution of parent external factor e_x^f and e_x^m
e_y	External factor of child face image y
$\mathcal{E}_\mathcal{Y}$	Distribution of child external factor e_y
$E_\mathcal{X}$	Encoder from $\mathcal{X} \to \mathcal{G}_\mathcal{X}$ of ChildPredictor
$E_\mathcal{Y}$	Encoder from $\mathcal{Y} \to \mathcal{G}_\mathcal{Y}$ of ChildPredictor
g_x^f	Genetic factor of father face image x^f
g_x^m	Genetic factor of mother face image x^m
$\mathcal{G}_\mathcal{X}$	Distribution of parent genetic factor g_x^f and g_x^m
g_y	Genetic factor of child face image y
$\mathcal{G}_\mathcal{Y}$	Distribution of child genetic factor g_y
$G_\mathcal{X}$	Generator from $\mathcal{G}_\mathcal{X} \to \mathcal{X}$ of ChildPredictor
$G_\mathcal{Y}$	Generator from $\mathcal{G}_\mathcal{Y} \to \mathcal{Y}$ of ChildPredictor
L_*	Loss functions for training the ChildPredictor
λ_*	Trade-off parameters for different loss functions
T	Mapping function from $\mathcal{G}_\mathcal{X} \to \mathcal{G}_\mathcal{Y}$ of ChildPredictor
v_y	Variety factor of child face image y
$\mathcal{V}_\mathcal{Y}$	Distribution of child variety factor g_y
x^f	Father face image
x^m	Mother face image
$\widehat{x_e}$	Generated parent face image according to e_x
$\mathcal{X}$	Parent domain (the set of parent face images)
y	Child face image
$\widehat{y_{e,v}}$	Generated child face image according to e_y and v_y
$\mathcal{Y}$	Children domain (the set of child face images)

Our framework is different from existing methods, as shown in Figure 1. Firstly, the I2I methods assume a shared content space for parent and children domains and easily falls into appearance collapse. However, the child face prediction is not simply transferring styles of the faces. Secondly, DNA-Net assumes a shared latent space for parent and child faces and then performs age-regression. However, this design cannot ensure the generator recovers realistic child faces. As for the ChildPredictor, we assume parent and child genetic factors are in the individual latent spaces and propose a mapping function T to learn the prediction from parents' genetic factors to child genetic factors.

During inference, our ChildPredictor firstly extracts genetic factors from the parent faces, then maps them to children genetic domain, finally predicts diverse outputs by sampling different genetic, external, and variety factors. The workflow is shown in Figure 3 (c) and all the notations are concluded in Table II. Details on the training process are presented below.

B. Domain-specific Disentangled Learning

1) Assumption: We perform the domain-specific disentangled learning in the parent and children domain separately. To ensure the transferability between face images and genetic factors for both domains, the two encoder-generator pairs ($E_\mathcal{X}$, $G_\mathcal{X}$), ($E_\mathcal{Y}$, $G_\mathcal{Y}$) must satisfying:

$$x^f = G_\mathcal{X}(E_\mathcal{X}(x^f), e_x^f), E_\mathcal{X}(x^f) = g_x^f, \tag{1}$$

$$x^m = G_\mathcal{X}(E_\mathcal{X}(x^m), e_x^m), E_\mathcal{X}(x^m) = g_x^m, \tag{2}$$

$$y = G_\mathcal{Y}(E_\mathcal{Y}(y), e_y, v_y), E_\mathcal{Y}(y) = g_y, \tag{3}$$

where e_* and v_* denote external and variety factors, respectively. The goal is to ensure that encoders should only extract genetic factors. To further differentiate their roles, we assume genetic factors follow normal distribution and variety factors follow uniform distribution. External factors are categorical attributes thus binary distributed.

2) Parent Domain $\mathcal{X}$: We adopt [71] as $E_\mathcal{X}$ and $G_\mathcal{X}$, as shown in Figure 3 (a). The encoder $E_\mathcal{X}$ firstly encodes a parent image x to a genetic factor $\widehat{g_x}$. Then, the generator $G_\mathcal{X}$ receives the produced genetic factor with a given external factor e_x to recover a face image $\widehat{x_e}$. It is an identity transformation only when the given e_x is from x; otherwise, $G_\mathcal{X}$ performs attribute transfer while maintaining identity unchanged. It is because genetic factors are disentangled from external factors. The *whole loss* $L_\mathcal{X}$ for training $E_\mathcal{X}$ and $G_\mathcal{X}$ is given as:

$$L_\mathcal{X} = \lambda_\mathcal{X}^1 L_\mathcal{X}^1 + \lambda_\mathcal{X}^2 L_\mathcal{X}^C + \lambda_\mathcal{X}^3 L_\mathcal{X}^G + \lambda_{\mathcal{G}_\mathcal{X}}^4 L_{\mathcal{G}_\mathcal{X}}^G, \tag{4}$$

where $\lambda_\mathcal{X}^1$, $\lambda_\mathcal{X}^2$, $\lambda_\mathcal{X}^3$ and $\lambda_{\mathcal{G}_\mathcal{X}}^4$ are trade-off parameters. The *L1 reconstruction loss* $L_\mathcal{X}^1$ maintains attribute-invariant features determined by genetic factors. The *classification loss* $L_\mathcal{X}^C$ enlarges the differences between faces with and without attributes determined by different dimensions of external factors, which share similar definitions with [48]. The *image-level adversarial loss* $L_\mathcal{X}^G/L_\mathcal{X}^D$ [25] enhances perceptual similarity. To further disentangle genetic and external factors in the latent space, we use a *genetic-factor-level adversarial loss* $L_{\mathcal{G}_\mathcal{X}}^G/L_{\mathcal{G}_\mathcal{X}}^D$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2022.3164785, IEEE Transactions on Multimedia

to make $\mathcal{G}_\mathcal{X}$ approach to standard normal distribution. The detailed formulations are:

$$L^1_\mathcal{X} = \mathbb{E}[||x - \widehat{x_e}||_1], \tag{5}$$

$$\begin{aligned} L^C_\mathcal{X} = &-e_x log(D^C_\mathcal{X}(x)) + (1-e_x)log(1-D^C_\mathcal{X}(x)) \\ &- e_x log(D^C_\mathcal{X}(\widehat{x_e})) + (1-e_x)log(1-D^C_\mathcal{X}(\widehat{x_e})), \end{aligned} \tag{6}$$

$$\begin{aligned} L^G_\mathcal{X} &= -\mathbb{E}[D^G_\mathcal{X}(\widehat{x_e})], \\ L^D_\mathcal{X} &= \mathbb{E}[D^G_\mathcal{X}(\widehat{x_e})] - \mathbb{E}[D^G_\mathcal{X}(x)], \end{aligned} \tag{7}$$

$$\begin{aligned} L^G_{\mathcal{G}_\mathcal{X}} &= -\mathbb{E}[D_{\mathcal{G}_\mathcal{X}}(\widehat{g_x})], \\ L^D_{\mathcal{G}_\mathcal{X}} &= \mathbb{E}[D_{\mathcal{G}_\mathcal{X}}(\widehat{g_x})] - \mathbb{E}[D_{\mathcal{G}_\mathcal{X}}(u)], \end{aligned} \tag{8}$$

where two discriminators are used to compute these losses. Image-level discriminator $D_\mathcal{X}$ has two branches, one of which outputs a binary vector for computing $L^C_\mathcal{X}$ and the other of which outputs a scalar for computing $L^G_\mathcal{X}$. Factor-level discriminator $D_{\mathcal{G}_\mathcal{X}}$ receives genetic factors. In Equation 8, u is a random sample from a standard Gaussian distribution $N(0,1)$ with the same dimension of genetic factors $\widehat{g_x}$. Note that the variety factor is not considered in the parent domain as there is only one parent pair for any child, while there is more than one child for each parent pair.

3) Children Domain $\mathcal{Y}$: We adopt PGGAN as $G_\mathcal{Y}$ to achieve a higher-quality generation. It generates images based on the given genetic factors, external factors, and variety factors, as shown in Figure 3 (a). To perform disentangled learning, we use a GAN-inverse encoder $E_\mathcal{Y}$ to recover only genetic factors from images generated by a pre-trained $G_\mathcal{Y}$.

To pre-train $G_\mathcal{Y}$, we define the *loss* $L_{G_\mathcal{Y}}$ as:

$$L_{G_\mathcal{Y}} = \lambda^1_\mathcal{Y} L^G_\mathcal{Y} + \lambda^2_\mathcal{Y} L^C_\mathcal{Y} + \lambda^3_\mathcal{Y} L^M_\mathcal{Y}, \tag{9}$$

where $L^G_\mathcal{Y}$ is an *adversarial loss* [12], which promotes $G_\mathcal{Y}$ generating realistic faces. $L^C_\mathcal{Y}$ is the *auxiliary classification loss* [72], which ensures external factors can control the facial attributes. $L^M_\mathcal{Y}$ is the *mode-seeking loss* [24] which ensures variety factors are related to individual variations. Their detailed definitions are as follows:

$$\begin{aligned} L^G_\mathcal{Y} &= -\mathbb{E}[D^G_\mathcal{Y}(\widehat{y_{e,v}})], \\ L^D_\mathcal{Y} &= \mathbb{E}[D^G_\mathcal{Y}(\widehat{y_{e,v}})] - \mathbb{E}[D^G_\mathcal{Y}(y)], \end{aligned} \tag{10}$$

$$\begin{aligned} L^C_\mathcal{X} = &-e_y log(D^C_\mathcal{Y}(y)) + (1-e_y)log(1-D^C_\mathcal{Y}(y)) \\ &- e_y log(D^C_\mathcal{Y}(\widehat{y_{e,v}})) + (1-e_y)log(1-D^C_\mathcal{Y}(\widehat{y_{e,v}})), \end{aligned} \tag{11}$$

$$L^M_\mathcal{Y} = \max_{G_\mathcal{Y}}\left(\frac{d_\mathcal{Y}(\widehat{y_{e,v^2}} - \widehat{y_{e,v^1}})}{d_{\mathcal{V}_\mathcal{Y}}(v^2_y - v^1_y)}\right), \tag{12}$$

where $\widehat{y_{e,v}}$ is the randomly generated result based on an external factor e_y and a variety factor v_y. y is a real sample selected from all training child images. $D^G_\mathcal{Y}$ and $D^C_\mathcal{Y}$ are two branches of the discriminator $D_\mathcal{Y}$. When computing $L^M_\mathcal{Y}$, there are two individual outputs $\widehat{y_{e,v^1}}$ and $\widehat{y_{e,v^2}}$ generated from the same external factor e_y but individual variety factors v^1_y and v^2_y. We adopt the l_1-norm as distance metric $d_*(\cdot)$, which includes 3 sequential operations: subtraction, taking absolute value, and computing average.

Then, we train $E_\mathcal{Y}$ to disentangle the genetic factor from the other two factors in the latent space of $G_\mathcal{Y}$. Suppose that a fixed genetic factor is used to generate faces with different external and variety factors, an ideal encoder can restore the same genetic factor (i.e., same identity) from those faces. Based on the assumption, we train the $E_\mathcal{Y}$ by *loss* $L_{E_\mathcal{Y}}$:

$$L_{E_\mathcal{Y}} = \mathbb{E}[||g_y - E_\mathcal{Y}(G_\mathcal{Y}(g_y, e_y, v_y))||_1], \tag{13}$$

where the $G_\mathcal{Y}$ is fixed. The e_y and v_y are randomly sampled for a same g_y. After its convergence, we can obtain the disentangled genetic factors for arbitrary child faces by $E_\mathcal{Y}$.

C. Inter-domain Multimodal Mapping

Different from previous I2I methods, we perform the "multimodal mapping" from parent domain to children domain only on genetic factors in the latent space. The reason is after the disentangled learning for domain $\mathcal{X}$ and $\mathcal{Y}$, the genetic factors g_* are disentangled from other factors e_* and v_*, and by our definition only the genetic factors among the three are related between the two domains.

To learn $\mathcal{G}_\mathcal{X} \rightarrow \mathcal{G}_\mathcal{Y}$, we firstly obtain parent-child genetic factor pairs (g^f_x/g^m_x, g_y) for each family by encoding faces through $E_\mathcal{X}$ and $E_\mathcal{Y}$. Then, we use a neural network as mapping function T, as shown in Figure 3 (b). To simulate multiple children, we simply let T predict k different genetic factors by k branches, i.e., $T : \widehat{g^1_y}, \widehat{g^2_y}, ..., \widehat{g^k_y} = T(g^f_x, g^m_x)$. We set $k = 4$ in our experiments. The training for T is supervised by parent-child genetic factor pairs and the following loss L_T:

$$\begin{aligned} L_T = &\mathbb{E}[||\widehat{g^1_y} - g^1_y||_1] + \lambda^1_T \mathbb{E}[||\widehat{g^2_y} - g^2_y||_1] \\ &+ \lambda^2_T \mathbb{E}[||\widehat{g^3_y} - g^3_y||_1] + \lambda^3_T \mathbb{E}[||\widehat{g^4_y} - g^4_y||_1], \end{aligned} \tag{14}$$

where $\widehat{g^j_y}$ and g^j_y are the j-th predicted and real genetic factor, respectively. Note that different loss coefficients (1, λ^1_T, λ^2_T, and λ^3_T) are used for the 4 predictions in order to fulfill the multimodal prediction. For families with 4 children or more, we select the first 4 children as ground truth g^1_y, g^2_y, g^3_y, and g^4_y. For families with less than 4 children, we replace the blank position g^j_y with g^1_y. For instance, for families with only 3 children, we use the 1-st child as ground truth for the 4-th branch. Note that, different loss coefficients are applied to 1-st and 4-th branches though their ground truth images are the same; while the ground truth images of 2-nd and 3-rd branches are different from 1-st and 4-th branches. Therefore, different optimizations are achieved for all 4 branches.

D. Network Architecture

1) Parent Domain $\mathcal{X}$: The parent domain networks consist of encoder-decoder pair $E_\mathcal{X}$, $G_\mathcal{X}$ and two discriminators $D_\mathcal{X}$/$D_{\mathcal{G}_\mathcal{X}}$. $E_\mathcal{X}$ has 5 generator convolutional blocks (including a convolutional layer, a BatchNorm [73] layer, and a LeakyReLU [74] activation function) to produce genetic factors g_x. $G_\mathcal{X}$ adopts 5 generator convolutional blocks to progressively upsample the combinations of genetic and external factors. The feature maps of the encoder are injected into the generator as U-Net [71]. $D_\mathcal{X}$ has 5 discriminator convolutional blocks (including a transposed convolutional layer, a InstanceNorm [75] layer, and a ReLU [76] activation function), followed by 2 parallel MLP layers. The outputs are used for

TABLE III
THE DETAILS FOR THE TRAINING STEPS. THE "NETWORK", "LR", "EPOCH" AND "LOSS" REPRESENT NETWORKS TRAINED IN THIS STEP, CURRENT LEARNING RATE, TOTAL EPOCHS AND LOSSES USED IN THE STEP, RESPECTIVELY. THE "ALL" DENOTES ALL NETWORKS OR ALL LOSSES ARE USED.

Step	Network	LR	Epoch	Loss	Dataset
1	$E_{\mathcal{X}}$, $G_{\mathcal{X}}$	2×10^{-4}	200	$L^1_{\mathcal{X}}$, $L^C_{\mathcal{X}}$, $L^G_{\mathcal{X}}$	FF-Database parent images
1	$G_{\mathcal{Y}}$	1.5×10^{-3}	200	$L_{\mathcal{Y}}$ ($L^G_{\mathcal{Y}}$, $L^C_{\mathcal{Y}}$, $L^M_{\mathcal{Y}}$)	FF-Database and FFHQ child images
2	$E_{\mathcal{X}}$, $G_{\mathcal{X}}$	2×10^{-4}	200	$L_{\mathcal{X}}$ ($L^1_{\mathcal{X}}$, $L^C_{\mathcal{X}}$, $L^G_{\mathcal{X}}$, $L^G_{\mathcal{Z}_{\mathcal{X}}}$)	FF-Database parent images
2	$E_{\mathcal{Y}}$	1×10^{-4}	200	$L_{E_{\mathcal{Y}}}$	no data needed ($E_{\mathcal{Y}}$ is trained by sampling from latent codes)
3	T	1×10^{-3}	200	L_T	FF-Database parents-children image pairs
4	All	1×10^{-5}	10	All	FF-Database parents-children image pairs

computing *WGAN adversarial loss* $L^G_{\mathcal{X}}$ and *classification loss* $L^G_{\mathcal{X}}$, respectively. $D_{\mathcal{G}_{\mathcal{X}}}$ contains 2 discriminator convolutional blocks followed by a MLP layer. The output is adopted to compute the *genetic factor adversarial loss* $L_{\mathcal{G}_{\mathcal{X}}}$.

2) Children Domain $\mathcal{Y}$: We modify the official PGGAN [27] as $G_{\mathcal{Y}}$, which receives genetic, external, and variety factors as inputs. The factors are concatenated along the channel dimension. The output resolution is 128×128. We use PixelNorm [27] and PReLU [77] as the normalization layer and activation function, respectively. $E_{\mathcal{Y}}$ inverses an image generated by $G_{\mathcal{Y}}$ back to the genetic factor. It is composed of a VGG-network [78] and a MLP layer, where the MLP layer projects the features from VGG-network into 480-dimensional output, which is consistent to the dimension of genetic factor.

3) Mapping Function T: The mapping network T receives the genetic factors of mother and father as the inputs. It consists of a head module, a body module, and a tail module. The head module is a simple combination of two convolutional layers and a LeakyReLU activation function. The body module contains 5 residual layers [79] with LeakyReLU activation function. The tail module includes 4 MLP layers which predicts 4 genetic factors ($\widehat{g^1_y}$, $\widehat{g^2_y}$, $\widehat{g^3_y}$, and $\widehat{g^4_y}$). In addition, we apply a normalization operation (minus mean and divide variance for each output genetic factor) to the output genetic factors to push them to standard Guassian distribution.

V. EXPERIMENT

A. Training Details

General Training Details. The training process of ChildPredictor can be concluded in 4 steps, as shown in Table III. Specifically, for step 1, the training of $E_{\mathcal{X}}$, $G_{\mathcal{X}}$, and $G_{\mathcal{Y}}$ are parallel. However, the training of step 2 is based on the results of step 1, and so as step 3 and 4. The batch size is set to 16 for each step. We initialize the network parameters using the Xavier initialization [80]. We use Adam optimizer [81] with β_1=0.5 and β_2=0.999. The discriminators share the same learning rates as corresponding generators. There is no weight decay used in the training procedure, but the learning rates for individual steps are different, as listed in Table III. There is no regularization terms used for the training.

Dataset. We adopt the training set of FF-Database (7148 parent and 8190 child faces) and 5000 high-quality child faces from FFHQ dataset [18]. The FFHQ child faces are post-processed based on the same pipeline as FF-Database, e.g., 1) We use dlib to extract and align faces; 2) We label them with the same attributes as in FF-Database. All parent and child faces (including FF-Database and FFHQ) are used in the domain-specific training of domain $\mathcal{X}$ and $\mathcal{Y}$, respectively.

TABLE IV
THE TRAINING DETAILS OF THE BASELINE CYCLEGAN. "$G_{p\to c}$" AND "$G_{c\to p}$" REPRESENT THE GENERATORS FROM THE PARENT DOMAIN TO THE CHILDREN DOMAIN AND ITS REVERSE, RESPECTIVELY. "D_p" AND "D_c" ARE THE DISCRIMINATORS FOR PARENT AND CHILDREN DOMAINS, RESPECTIVELY. "INPUT NC" AND "OUTPUT NC" ARE INPUT AND OUTPUT IMAGES WITH N CHANNELS, RESPECTIVELY.

Item	Original CycleGAN	Baseline CycleGAN
$G_{p\to c}$	input 3c, output 3c	input 6c, output 3c
$G_{c\to p}$	input 3c, output 3c	input 3c, output 6c
D_p	input 3c	input 6c
D_c	input 3c	input 3c
Training data	random parent-child data	paired parents-child data
Parameters	the same as [19]	the same as [19]

The motivation to use FFHQ child faces is to enhance the generation quality of $G_{\mathcal{Y}}$.

The attributes used in the training for parent domain $\mathcal{X}$ are gender, moustache, glasses and expression, while they are age, gender, glasses and expression for children domain $\mathcal{Y}$. To simplify the computation of the classification losses, we define age and expression as binary attributes (i.e., young or senior and laugh or not laugh, respectively).

Loss Function. The loss functions used in each step are concluded in Table III. The coefficients of different loss terms are shared for different steps, i.e., $\lambda^1_{\mathcal{X}}$, $\lambda^2_{\mathcal{X}}$, $\lambda^3_{\mathcal{X}}$, $\lambda^4_{\mathcal{Z}_{\mathcal{X}}}$, $\lambda^1_{\mathcal{Y}}$, $\lambda^2_{\mathcal{Y}}$, $\lambda^3_{\mathcal{Y}}$, $\lambda^1_{\mathcal{T}}$, $\lambda^2_{\mathcal{T}}$, and $\lambda^3_{\mathcal{T}}$ are empirically set to 100, 10, 1, 0.1, 1, 1, 5, 0.8, 0.6, and 0.4, respectively.

Time. The ChildPredictor is trained on 8 NVIDIA Titan Xp GPUs (12 Gb memories for each). It is implemented by PyTorch 1.1.0 framework and Python 3.6. The training time of PGGAN $G_{\mathcal{Y}}$ is approximately 5 days. Considering the parallel training procedure (see Table III), the remaining training time takes approximately 7 days.

B. Experiment Settings

1) Baselines: To give a comprehensive and fair comparison, we do experiment on both domain transferring methods and state-of-the-art child face prediction DNA-Net. They are,

1) I2I: DualGAN [20], CycleGAN [19], UNIT [21], DRIT [14], MUNIT [15], and DRIT++ [16], where DRIT, MUNIT, and DRIT++ predict multimodal by changing latent style codes;
2) DNA-Net [13]: It predicts multimodal by changing the linear factor in the random selection S.

Since input and output are paired, we adjust the training scheme of all baselines by feeding parent-child pairs instead of randomly choosing samples from the whole training dataset for fairness. An example of I2I method is given in Table IV. Since DNA-Net is not open-source, we train it with the same

attributes as ChildPredictor, where fathers and mothers are separately encoded, as shown in Figure 1 (b). In addition, to avoid ChildPredictor (using glasses, emotion, age and gender) and DNA-Net (using age and gender) using different face attributes in comparison, we fix the child external factors equal to ground truth for fairness.

2) Evaluation: We perform experiments on 170 validation parent-child pairs of the FF-Database. They are of the same image resolution as training data. Normally, the generated and ground truth faces are not strictly unaligned. To evaluate the generation quality, it is more reasonable to compare feature-level face similarity. To evaluate the generation diversity, we evaluate how large the differences are among different generated child faces from the same parent. For every family and every multimodal method, we predict 40 different child faces, which forms 40 groups of results. The experiments of subsections *C-H* are performed on FF-Database validation set.

Cosine distance (Cos. Dis.). It measures feature-level cosine similarity between two faces. We use features from the second-last fully-connected layer of a Sphere20 network [82], [83] pre-trained on CelebA dataset [84]. We compute pairwise cosine similarity for every family in every group and then compute the average of 40 groups. In order to demonstrate the effectiveness of cosine similarity, we randomly shuffle the 170 real parent-child pairs to obtain 6800 random parent-child pairs. Then, we compute the average of all the cosine similarity values. The average is 0.3204.

Fréchet Inception Distance (FID) [22]. It measures the distance between two sets of images, i.e., the generated child faces and training data. The training data denotes child faces in the FF-Database training set. We use the features from the default "pool3" layer of the Inception-V3 [85] pre-trained on ImageNet [86]. We compute the FID for every group and then compute the average of 40 groups.

Learned Perceptual Image Patch Similarity (LPIPS) [23]. It measures the diversity of the generated child faces. It is represented by the L1 distance between the features extracted from the AlexNet [87] pre-trained on ImageNet [86]. We compute the average of the pairwise distances among 40 groups of outputs for every family (i.e., 780 pairs if 40 outputs, $C_{40}^2 = 780$). Then, we compute the average across all 170 families. Note that, the color change is unwanted in child face prediction, although it facilitates the output diversity. For accurate evaluation, we conduct histogram equalization on all the generated images of all algorithms in comparisons. The histogram equalization operation removes the effect of color shift obviously. Thus, the evaluation is more fair in terms of the baselines (i.e., the style transfer methods often change the skin color, which is not real for this task).

Human Perceptual Study. We perform a human perceptual study to subjectively evaluate different methods. If a method obtains higher preference rates, it shows the method can produce higher-quality and more diverse faces. There are overall 14 human observers. The generated faces, input parents, and real child faces are presented to observers. The observers need to select one method that predicts faces closest to ground truth for each family. Finally, we count the preference rates.

TABLE V
QUANTITATIVE ANALYSIS OF BASELINES AND OUR CHILDPREDICTOR ON COSINE DISTANCE, FID, AND LPIPS METRICS. THE BEST PERFORMANCES ARE HIGHLIGHTED WITH THE RED COLOR.

Method	Cos. Dis. ↑	FID ↓	LPIPS ↑
DualGAN	0.3733	82.01	/
CycleGAN	0.3805	71.23	/
UNIT	0.3727	71.78	/
DRIT	0.3843	62.93	0.0041
MUNIT	0.3702	63.61	0.1669
DRIT++	0.2013	79.92	0.0056
DNA-Net	0.3137	88.23	0.0087
ChildPredictor (normal)	0.4245	60.73	0.0063
ChildPredictor (full)	**0.4303**	**38.15**	**0.2757**

TABLE VI
SUBJECTIVE HUMAN PERCEPTUAL STUDY RESULTS OF BASELINES AND CHILDPREDICTOR ON PREFERENCE RATES (PRS).

Method	Similarity PR	Diversity PR
DualGAN	0.84%	/
CycleGAN	1.01%	/
UNIT	4.08%	/
DRIT	7.61%	5.04%
MUNIT	6.09%	9.83%
DRIT++	1.21%	1.14%
DNA-Net	5.13%	1.05%
ChildPredictor (full)	**74.03%**	**82.94%**

C. Experiment on Child Face Prediction Reality

1) Qualitative Analysis: We illustrate some generated samples by different methods in Figure 4. There are 4 samples for each multimodal method in the figure, where we sample different style codes for baselines to generate multiple faces, while we change genetic factors or variety factors for ChildPredictor.

Firstly, results from I2I methods (yellow background) are very similar to mothers. We claim it is an "appearance collapse" issue caused by the shared content space assumption. It promotes the networks simply copying the face structures from mothers; thereby the results are not similar enough compared with real children. Though we feed parent-child pairs to train baselines (see Figure IV), the disentanglement of content and style is not appropriate for this task. Secondly, DNA-Net assumes parents and children share the same content space like I2I methods but additionally uses a mapper to learn the relation between parent content codes and child content codes. After that, it performs an age-regression to the mapped child content codes to obtain a face. It is not like a natural biological process thus leading to artifacts in the generated faces.

ChildPredictor predicts very similar results with real children (e.g., face structure, color, and facial features). Also, the generated faces have less artifacts than baselines. We claim it is because the proposed disentangled learning is more accurate than the separation of content and style. Since external and variety factors are disentangled from genetic factors, the learning between parent and children domains of ChildPredictor is only on genetic factors. Compared with style codes, genetic factors are a special design for this task.

2) Quantitative Analysis: The quantitative results are concluded in Table V. Since the I2I baselines do not adopt face attributes and FFHQ child data, we exclude them at the training stage for fairness, i.e., "ChildPredictor (normal)". Firstly, "ChildPredictor (normal)" obtains better cosine similarity and FID than the baselines. It demonstrates the archi-

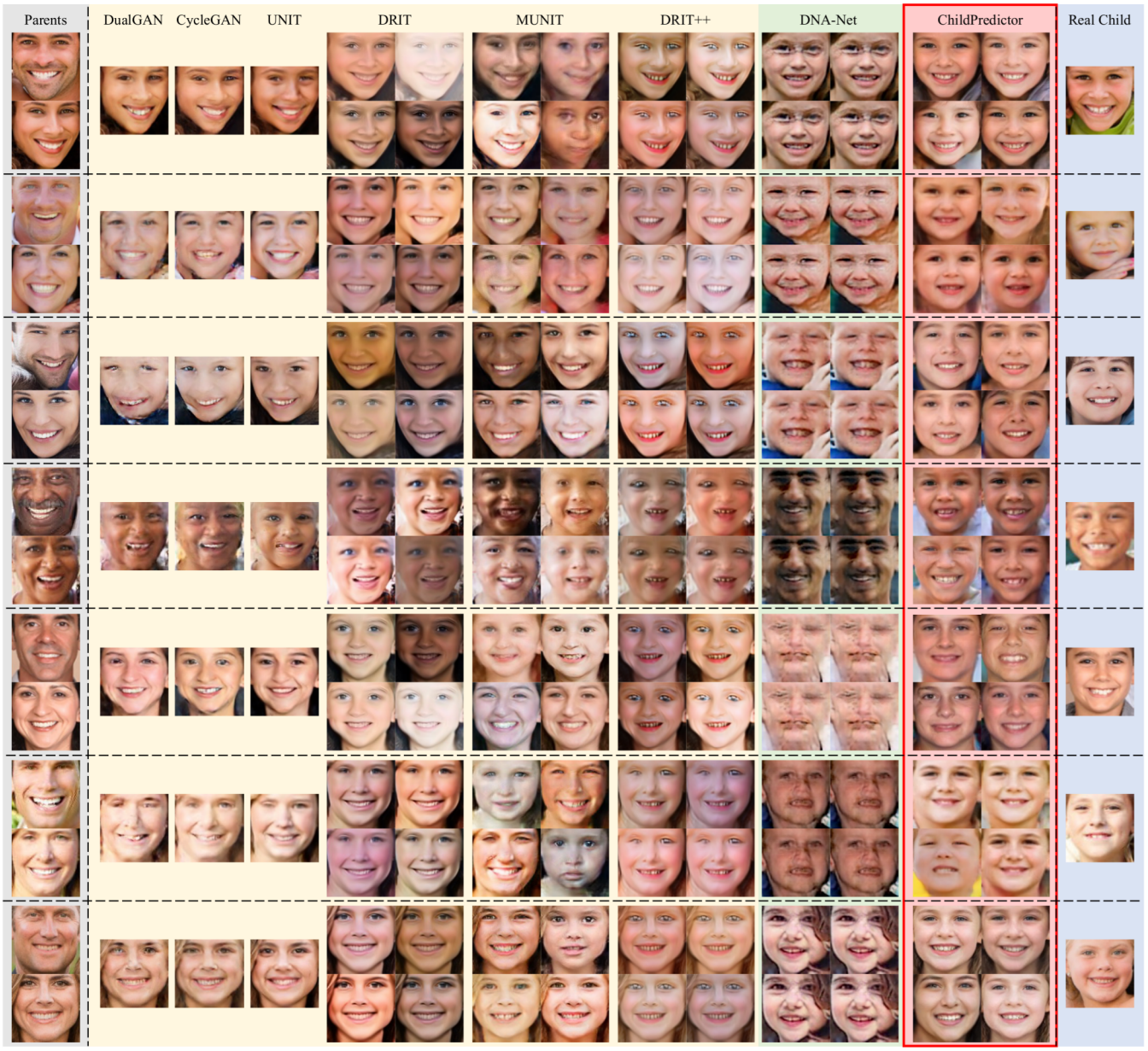


Fig. 4. Illustration of some generated samples by ChildPredictor and baselines. The 1st column is input parents and the last column is real children. The 2nd-4th columns include parent-child pairs and single predicted children by different methods, respectively. The 5th-9th columns include diverse children generated by representative I2I methods (yellow background), DNA-Net (green background), and ChildPredictor (red background), respectively.

tecture predicts the closest samples with ground truth with the best perceptual quality. Secondly, the full ChildPredictor enhances the results of "ChildPredictor (normal)". It is because of the use of attributes for the disentangled learning, which denotes the prediction-irrelevant information is disentangled from genetic factors. In addition, ChildPredictor obtains the highest LPIPS. It demonstrates that the design of predicting 4 genetic factors by T and variety factors are helpful for producing diverse faces.

3) Human Perceptual Study: The preference rates (PRs) for each method in terms of both similarity and diversity are concluded in Table VI. The ChildPredictor obtains clearly higher PRs than baselines, which demonstrates that it predicts perceptually more realistic and diverse faces, respectively.

D. Ablation Study

We conduct 9 experiments to evaluate several key components of ChildPredictor. The comparison results are included in the Table VII and Figure 5. The analysis is as follows:

1) Disentangled Learning Ability: We exclude the classification losses $L_{\mathcal{X}}^{C}$ or $L_{\mathcal{Y}}^{C}$ or both to train ChildPredictor without external factors. Also, we optimize $E_{\mathcal{Y}}$ in image space rather than on genetic factors (i.e., $L_{E_{\mathcal{Y}}}$ performs on images), leading to false disentangled learning in children domain. All the settings lead to obvious decreases of metrics (e.g., more than 0.06 decrease of Cos. Dis for "w/o $L_{\mathcal{Y}}^{C}$"). The outputs are not realistic, even contain visual artifacts, e.g., the faces, eyes and mouths marked by green rectangles of Figure 5 1).

2) Generation Diversity: We exclude the mode-seeking loss $L_{\mathcal{Y}}^{M}$ or let mapping function T produce one genetic factor

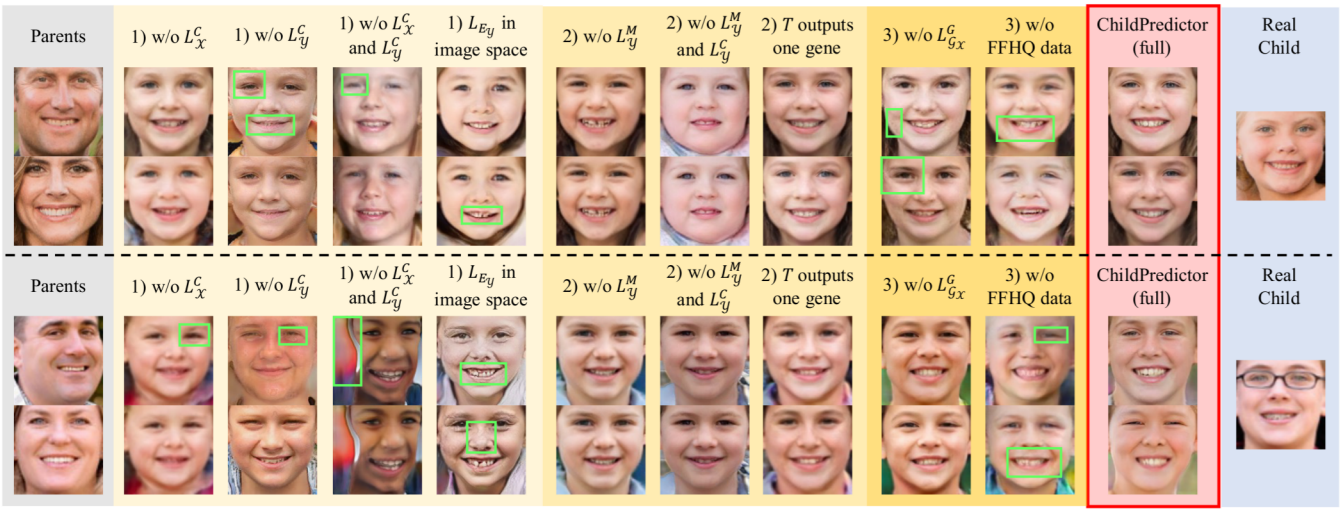


Fig. 5. Illustration of the generated images under 9 different ablation study settings. There are 2 output images for each setting by only changing the variety factor. The left column is input parents (gray background), the middle part includes results of different settings (yellow background), the right parts are full ChildPredictor's results and real child faces (red and blue background, respectively).

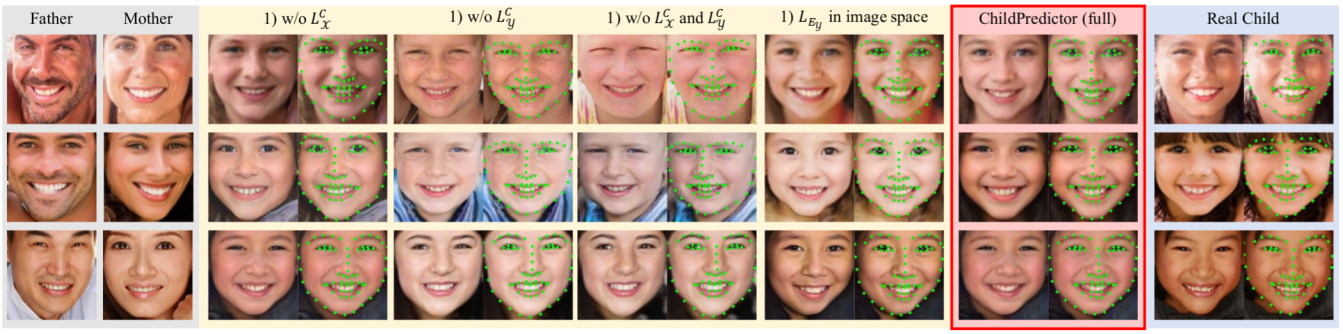


Fig. 6. Illustration of more results on ChildPredictor with and without disentangled learning (i.e., ablation study setting 1)). The left two columns are inputs for the ChildPredictor. The 68 face landmarks (extracted by dlib) are illustrated alongside the generated child faces and real child faces.

TABLE VII
COMPARISON OF DIFFERENT ABLATION STUDY SETTINGS OF THE PROPOSED CHILDPREDICTOR.

Ablation study setting	Cos Dis. ↑	FID ↓	LPIPS ↑
1) w/o L_X^C	0.4167	43.76	0.2875
1) w/o L_Y^C	0.3665	39.71	0.2161
1) w/o L_X^C and L_Y^C	0.3572	39.89	0.2306
1) L_{E_Y} in image space	0.3592	39.39	**0.3511**
2) w/o L_Y^M	0.4247	54.51	0.2213
2) w/o L_Y^M and L_Y^C	0.4168	64.14	0.2213
2) T outputs one gene	0.4161	52.57	0.0653
3) w/o $L_{G_X}^G$	0.4303	45.94	0.2236
3) w/o FFHQ data	0.4296	69.11	0.1482
ChildPredictor (full)	**0.4303**	**38.15**	0.2757

instead of 4 factors. The LPIPS metric of these settings decreases obviously due to no effect of variety factors or no diverse output genetic factors. Therefore, those different generated faces are almost the same, as shown in Figure 5 2).

3) Other Terms: We exclude the auxiliary loss $L_{G_X}^G$ or additional FFHQ data when training G_Y. All the metrics decrease since $L_{G_X}^G$ and FFHQ data contribute to high-quality image generation. The predicted faces are blurry or ghosted if excluding them, as the specific regions marked by green rectangles of Figure 5 3).

In conclusion, all the network components, loss functions, and disentangled learning method are significant for ChildPredictor to generate realistic and diverse child faces.

E. Experiment on Disentangled Learning

To further demonstrate the effectiveness of disentangled learning, we show more predicted faces with and without disentangled learning (w/o L_X^C, w/o L_Y^C, w/o L_X^C and L_Y^C, and L_{E_Y} in image space; please see ablation study setting 1)). The results are illustrated in Figure 6, where 68 landmarks are illustrated for every face. The results from full ChildPredictor are more similar to real children.

F. Disentangled Learning and Robustness Analysis

The disentangled learning is the basic of ChildPredictor since it assists to extract accurate genetic factors. For parent domain, we walk the latent code of g_x or e_x and fix the other, as shown in Figure 7. Obviously, changing genetic factors g_x will not influence attributes, and modifying attributes e_x will not change identity. Therefore, the parent genetic factors can well represent prediction-relevant information. For children domain, we illustrate generated faces from one input parent, e.g., from different genetic factors $\widehat{g_y^j}$ predicted by mapping function or different external factors e_y^j and variety factors v_y^j, as shown in Figure 8 (a). The e_y^j only changes specific attributes, while the v_y^j only influences individual properties when other factors are fixed. We also show generated faces by changing one of input parents, as shown in Figure 8 (b). The model is robust to different inputs, which proves that it does not fall into the appearance collapse.

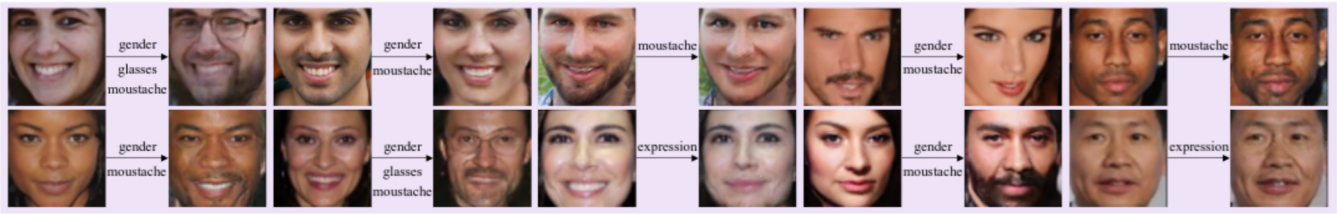
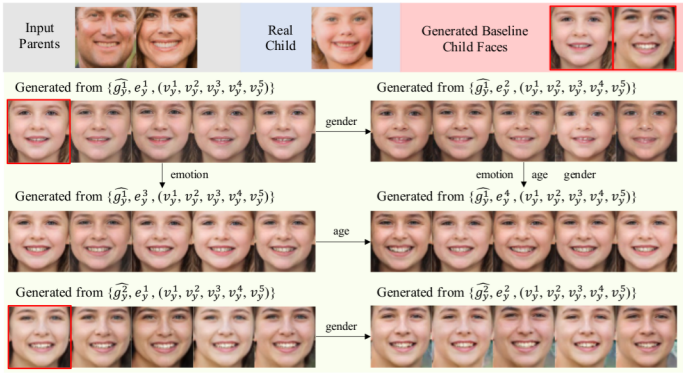
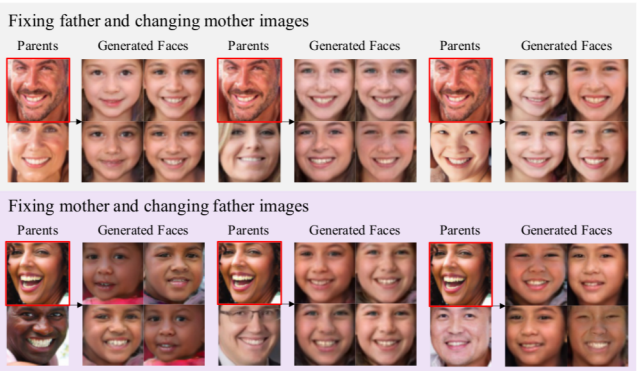


Fig. 7. Disentangled learning in parent domain $\mathcal{X}$. The left is input and the right is the generated face. The texts around arrows denote changed attributes.

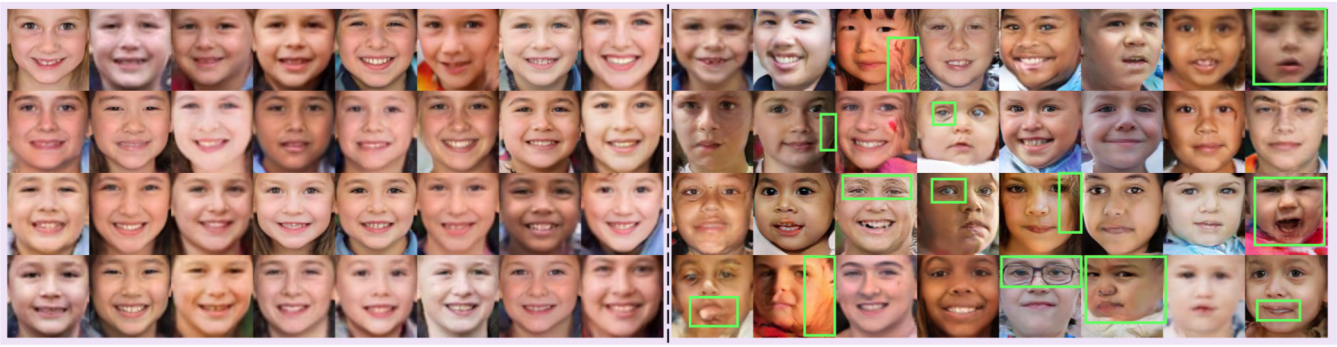


(a) Disentangled learning in domain $\mathcal{Y}$. The top row includes the real family faces and output. There are 6 groups of faces shown, where the genetic and external factors remain unchanged in each group, while variety factors change. The arrow denotes the change of external factor between each 2 groups.



(b) Robustness analysis in domain $\mathcal{Y}$. The upper 3 groups of images show the ChildPredictor can generate diverse and high-quality child faces when fixing father image and changing mother images. The lower 3 groups of images show the results when fixing mother image and changing father images.

Fig. 8. Illustration of (a) Disentangled learning in children domain $\mathcal{Y}$: generated faces of the same parents from different genetic, external, and variety factors; (b) Robustness analysis in children domain $\mathcal{Y}$: fixing one of input parents and changing the other.



(a) Randomly selected face images on FF-Dataset validation set generated by the full ChildPredictor (supervised, conditioned on parent images). There are almost no artifacts.

(b) Randomly selected face images on by the pre-trained children domain generator G_y' (unsupervised, generated from random latent codes). The unreal regions or artifacts are highlighted by green rectangles.

Fig. 9. Illustration of child face prediction results by supervised ChildPredictor and the unsupervised pre-trained children domain generator $G_{\mathcal{Y}}^{'}$. There are 32 images shown for each setting, which are randomly sampled from 6800 predicted face images.

TABLE VIII
COMPARISON OF THE FID FOR THE FULL CHILDPREDICTOR AND THE PRE-TRAINED CHILDREN DOMAIN GENERATOR $G_{\mathcal{Y}}^{'}$ ON VALIDATION SET.

Method	FID ↓
Full ChildPredictor (supervised learning)	**50.77**
Pre-trained $G_{\mathcal{Y}}^{'}$ (unsupervised learning)	53.89

G. Supervised Learning Analysis

To demonstrate the information from parents improves the child prediction quality, we compare results from the full ChildPredictor framework with randomly generated samples from the pre-trained PGGAN ($G_{\mathcal{Y}}^{'}$). The parent-child pairs are utilized for supervised learning for our framework. Note that $G_{\mathcal{Y}}^{'}$ is pre-trained in an unsupervised manner (following PGGAN training as Equation 9); therefore, the weights are not the same as $G_{\mathcal{Y}}$ in the ChildPredictor framework. There are overall 6800 generated face images on validation set by ChildPredictor and we also randomly generate 6800 samples by pre-trained $G_{\mathcal{Y}}$. To compare the image generation quality of supervised learning and unsupervised learning, we compute the FID for them on validation set and the results are concluded in Table VIII. It shows that results from the full ChildPredictor are more similar to real child faces in the validation set than the pre-trained PGGAN $G_{\mathcal{Y}}^{'}$.

In addition, we illustrate the generated results of the full ChildPredictor and $G_{\mathcal{Y}}^{'}$ in Figure 9. It is obvious that the children domain generator can predict faces with better quality conditioned on parent images. The supervised learning provides a more fixed latent space for children domain generator $G_{\mathcal{Y}}$; therefore, it can produce more reasonable faces.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2022.3164785, IEEE Transactions on Multimedia

(a) Child face prediction results on grandparents. Faces are extracted by dlib. "Input 1" and "Input 2" denote the "father input" and "mother input" for the ChildPredictor, respectively.

(b) Child face prediction results on black background. Faces are extracted by dlib. "Input 1" and "Input 2" denote the "father input" and "mother input" for the ChildPredictor , respectively.

Fig. 10. Illustration of child face prediction results on grandparents and black background, respectively. The original images are also shown for reference.

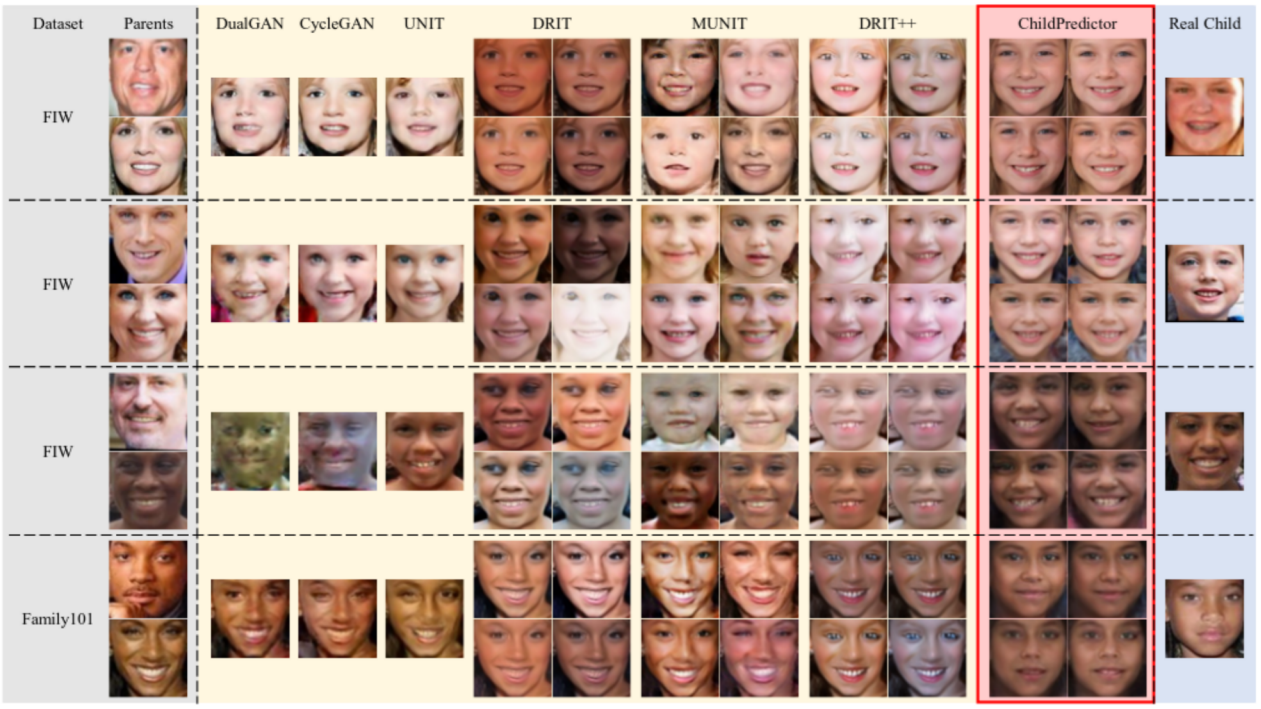


Fig. 11. Illustration of some generated samples by ChildPredictor (red background) and I2I baselines (yellow background). Input parents and real children are shown in the left and right, respectively. The samples are selected from the processed FIW dataset and Family101 dataset, respectively.

H. Experiment on Other Circumstances

We consider two other circumstances: 1) The "parents" are replaced by "grandparents" and 2) The background is darkened. The predicted child faces are illustrated in Figure 10, where results from the normal setting are also shown for comparisons. In case 1), the ChildPredictor can still output child face given grandparent faces. It is because the mapping function can still map the grandparents' genetic factors to child genetic factors in the learned space, while the children domain generator transforms the predicted child genetic factors to child faces. In case 2), the ChildPredictor can still predict a face similar with ground truth when changing the background color (e.g., the background color is darkened in Figure 10 (b)). However, the two predicted child faces under normal lighting condition and dark background are not very similar. It is because different background colors are not modeled during the training. In future work, we will consider it and make the ChildPredictor more robust.

I. Experiment on Other Datasets

To further evaluate the proposed ChildPredictor, we include two more datasets: Families in the Wild (FIW)[2] [88] and Family101[3] [89]. However, the original images in the datasets are not processed with the same pipeline as FF-Database, or have different image resolutions and formats (e.g., grayscale format). To minimize the gap, we apply the same pre-processing procedures to the images in the two datasets. In addition, we manually exclude some grandparent-parent pairs and some profile face images. The pre-processing procedures result in 50 validation pairs from the FIW dataset (extracted from "F0001" to "F0200" of FIW training images) and 17 validation pairs from the Family101 dataset, respectively. They are of 128×128 resolution and not overlapped with FF-Database images. The processed images will be publicly available.

[2]https://web.northeastern.edu/smilelab/fiw/
[3]http://chenlab.ece.cornell.edu/projects/KinshipClassification/index.html

TABLE IX
QUANTITATIVE ANALYSIS OF BASELINES AND OUR CHILDPREDICTOR ON COSINE DISTANCE AND LPIPS METRICS. THE RESULTS ON FIW AND FAMILY101 ARE SEPARATELY REPRESENTED.

Method	FIW		Family101	
	Cos. Dis. ↑	LPIPS ↑	Cos. Dis. ↑	LPIPS ↑
DualGAN	0.3533	/	0.3612	/
CycleGAN	0.3649	/	0.3771	/
UNIT	0.3793	/	0.3878	/
DRIT	0.3666	0.0047	0.3531	0.0049
MUNIT	0.3658	0.1727	0.3665	0.1715
DRIT++	0.1907	0.0065	0.1648	0.0069
ChildPredictor	**0.4259**	**0.2247**	**0.3914**	**0.2323**

Since the validation sets are relatively small, we only adopt the cosine distance and LPIPS as quantitative metrics because they evaluate the pairwise image quality. The baselines and ChildPredictor trained on the FF-Database are used in the experiment. Note that the DNA-Net cannot well generalize to FIW and Family101 images (e.g., predicted faces are extremely ambiguous) so we do not include it in the experiment. The quantitative results are concluded in Table IX. The proposed ChildPredictor achieves better performances than other methods according to face prediction similarity and diversity on both datasets. The results are consistent with the conclusion on the FF-Database validation set.

Some samples are illustrated in Figure 11. Compared with other methods, ChildPredictor predicts more similar faces to real children, more diverse faces, and higher-quality faces. For instance, I2I baselines (please see the yellow background in Figure 11) still produce faces with very similar shapes to input mothers, i.e., appearance collapse. However, ChildPredictor does not have this issue since the mapping is performed in the genetic domain, while the face-prediction-irrelevant factors are disentangled. The visual results also demonstrate that the proposed ChildPredictor has better generalization ability since it is not trained on these two datasets.

J. Experiment on Famous Families

We download 4 famous family images from the Internet for testing the ChildPredictor. The same pre-processing pipeline is applied to these images. The predicted face images are illustrated in Figure 12. The ChildPredictor framework can predict reasonable child faces for these real-world cases.

K. Limitation of ChildPredictor Framework

For many situations, ChildPredictor can generate high-quality child faces. However, there are some common failure cases shown in Figure 13 including: side face input (left two samples) and low-quality input (right two samples). It may be because there are little profile or low-quality faces in FF-Database. In the future, we will enhance the ChildPredictor to be more robust to such input images.

VI. BROADER IMPACT

The success of ChildPredictor depends on the proposed large-scale FF-database. Although the faces in the dataset are anonymous, they may be sensitive to privacy issues. *We are strongly aware that privacy protection is a significant issue in the community.* To largely protect privacy: 1) We plan not to disclose original face images of the FF-Database; however, *we will alternatively release the features extracted by state-of-the-art face recognition networks for future study*. 2) In terms of application, we encode faces by an irreversible process and delete the background data; 3) In terms of algorithm, we can adopt privacy-preserving generative models [90]–[93]. These methods promote discoveries that may be hindered by data-protection barriers and maintain the reproducibility of the algorithm; 4) In terms of generated data, we can learn the characteristics of generated samples to identify them [94]. They are orthogonal work and we will not include them in this paper. In the future, we will use them to enhance the privacy protection of ChildPredictor.

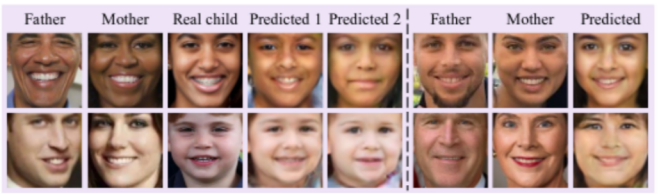


Fig. 12. Illustration of child face prediction results on 4 famous family images downloaded from the Internet.

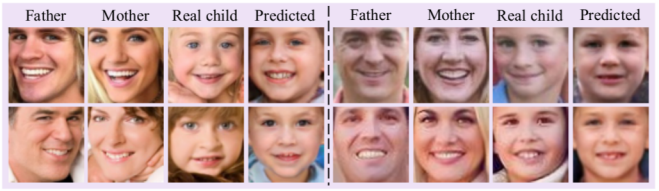


Fig. 13. Illustration of common failure predicted faces.

The ChildPredictor can synthesize realistic child faces from parents. It helps to solve many social issues, such as missing child identification and criminal investigations. However, to avoid improper use, we are very cautious about the pros and cons of ChildPredictor. We will strictly control the use of ChildPredictor in the aforementioned social applications.

VII. CONCLUSION

In this paper, we presented a ChildPredictor framework to automatically predict diverse child faces from parents. In order to simulate a biological process, we formulate it as a genetic factor mapping problem. We learn this mapping from parents to children in the latent space. We adopt the encoder-generator architecture to connect the image spaces and latent spaces of the parent and children domains. To extract precise genetic factors, we exclude external factors (facial attributes) and variety factors (individual properties) based on disentangled learning. For the parent domain, it is achieved by enforcing classification loss on generated faces. For the children domain, it is implemented by regularizing the latent space by a GAN encoder. We collected a large-scale Family Face Database (FF-Database) to train the ChildPredictor. It includes 16046 faces (7148 parent faces and 8190 child faces) with labeled facial attributes. Finally, we validated the ChildPredictor with

several state-of-the-art methods by both quantitative analysis and human perceptual study on the FF-Database. Experiment results demonstrate that our ChildPredictor can predict higher-quality, more diverse, and more realistic child faces than state-of-the-art methods.

ACKNOWLEDGMENT

The authors would like to thank Kangcheng Liu, Qinbin Li, Zhanghan Ke, and Yurou Zhou for their reviews, and Su Wang for labeling a part of facial attributes in the proposed FF-Database. The authors would also like to thank the anonymous reviewers and editors for their helpful comments.

REFERENCES

[1] S. Xia, M. Shao, and Y. Fu, "Kinship verification through transfer learning," in *Proc. IJCAI*, 2011, pp. 2539–2544.
[2] J. Lu, X. Zhou, Y.-P. Tan, Y. Shang, and J. Zhou, "Neighborhood repulsed metric learning for kinship verification," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 2, pp. 331–345, 2013.
[3] H. Dibeklioglu, A. Ali Salah, and T. Gevers, "Like father, like son: Facial expression dynamics for kinship verification," in *Proc. ICCV*, 2013, pp. 1497–1504.
[4] Y. Sun, J. Li, Y. Wei, and H. Yan, "Video-based parent-child relationship prediction," in *Proc. VCIP*, 2018, pp. 1–4.
[5] W. Li, S. Wang, J. Lu, J. Feng, and J. Zhou, "Meta-mining discriminative samples for kinship verification," in *Proc. CVPR*, 2021, pp. 16 135–16 144.
[6] U. Park, Y. Tong, and A. K. Jain, "Age-invariant face recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 5, pp. 947–954, 2010.
[7] Z. Huang, J. Zhang, and H. Shan, "When age-invariant face recognition meets face age synthesis: A multi-task learning framework," in *Proc. CVPR*, 2021, pp. 7282–7291.
[8] C. Yan, L. Meng, L. Li, J. Zhang, Z. Wang, J. Yin, J. Zhang, Y. Sun, and B. Zheng, "Age-invariant face recognition by multi-feature fusionand decomposition with self-attention," *ACM Trans. on Multimedia Computing, Communications, and Applications*, vol. 18, no. 1s, pp. 1–18, 2022.
[9] J. Zhao, S. Yan, and J. Feng, "Towards age-invariant face recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 1, pp. 474–487, 2022.
[10] W. Wang, S. You, and T. Gevers, "Kinship identification through joint learning using kinship verification ensembles," in *Proc. ECCV*, 2020, pp. 613–628.
[11] P. S. Chandran, N. Byju, R. Deepak, K. Nishakumari, P. Devanand, and P. Sasi, "Missing child identification system using deep learning and multiclass svm," in *Proc. IEEE RAICS*, 2018, pp. 113–116.
[12] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Proc. NeurIPS*, 2014, pp. 2672–2680.
[13] P. Gao, J. Robinson, J. Zhu, C. Xia, M. Shao, and S. Xia, "Dna-net: Age and gender aware kin face synthesizer," in *Proc. ICME*, 2021, pp. 1–6.
[14] H.-Y. Lee, H.-Y. Tseng, J.-B. Huang, M. Singh, and M.-H. Yang, "Diverse image-to-image translation via disentangled representations," in *Proc. ECCV*, 2018, pp. 35–51.
[15] X. Huang, M.-Y. Liu, S. Belongie, and J. Kautz, "Multimodal unsupervised image-to-image translation," in *Proc. ECCV*, 2018, pp. 172–189.
[16] H.-Y. Lee, H.-Y. Tseng, Q. Mao, J.-B. Huang, Y.-D. Lu, M. Singh, and M.-H. Yang, "Drit++: Diverse image-to-image translation via disentangled representations," *Int. J. Comput. Vis.*, pp. 1–16, 2020.
[17] Z. Zhang, Y. Song, and H. Qi, "Age progression/regression by conditional adversarial autoencoder," in *Proc. CVPR*, 2017, pp. 5810–5818.
[18] T. Karras, S. Laine, and T. Aila, "A style-based generator architecture for generative adversarial networks," in *Proc. CVPR*, 2019, pp. 4401–4410.
[19] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proc. ICCV*, 2017, pp. 2223–2232.
[20] Z. Yi, H. Zhang, P. Tan, and M. Gong, "Dualgan: Unsupervised dual learning for image-to-image translation," in *Proc. ICCV*, 2017, pp. 2849–2857.
[21] M.-Y. Liu, T. Breuel, and J. Kautz, "Unsupervised image-to-image translation networks," in *Proc. NeurIPS*, 2017, pp. 700–708.
[22] 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," in *Proc. NeurIPS*, 2017, pp. 6626–6637.
[23] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in *Proc. CVPR*, 2018, pp. 586–595.
[24] Q. Mao, H.-Y. Lee, H.-Y. Tseng, S. Ma, and M.-H. Yang, "Mode seeking generative adversarial networks for diverse image synthesis," in *Proc. CVPR*, 2019, pp. 1429–1437.
[25] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein generative adversarial networks," in *Proc. ICML*, 2017, pp. 214–223.
[26] M. Mirza and S. Osindero, "Conditional generative adversarial nets," *arXiv preprint arXiv:1411.1784*, 2014.
[27] T. Karras, T. Aila, S. Laine, and J. Lehtinen, "Progressive growing of gans for improved quality, stability, and variation," in *Proc. ICLR*, 2018.
[28] T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, and T. Aila, "Analyzing and improving the image quality of stylegan," in *Proc. CVPR*, 2020, pp. 8110–8119.
[29] G. Lore, A. Alex, O. Aude, and I. Phillip, "Ganalyze: Toward visual definitions of cognitive image properties," in *Proc. ICCV*, 2019, pp. 5744–5753.
[30] J. Ali, C. Lucy, and I. Phillip, "On the "steerability" of generative adversarial networks," in *Proc. ICLR*, 2019.
[31] D. Bau, J.-Y. Zhu, J. Wulff, W. Peebles, H. Strobelt, B. Zhou, and A. Torralba, "Inverting layers of a large generator," in *Proc. ICLR Workshop*, 2019, p. 4.
[32] A. Gabbay and Y. Hoshen, "Style generator inversion for image enhancement and animation," *arXiv preprint arXiv:1906.11880*, 2019.
[33] R. Abdal, Y. Qin, and P. Wonka, "Image2stylegan: How to embed images into the stylegan latent space?" in *Proc. ICCV*, 2019, pp. 4432–4441.
[34] Abdal, Rameen and Qin, Yipeng and Wonka, Peter, "Image2stylegan++: How to edit the embedded images?" in *Proc. CVPR*, 2020, pp. 8296–8305.
[35] Y. Shen, J. Gu, X. Tang, and B. Zhou, "Interpreting the latent space of gans for semantic face editing," in *Proc. CVPR*, 2020, pp. 9243–9252.
[36] Y. Viazovetskyi, V. Ivashkin, and E. Kashin, "Stylegan2 distillation for feed-forward image manipulation," *arXiv preprint arXiv:2003.03581*, 2020.
[37] E. Richardson, Y. Alaluf, O. Patashnik, Y. Nitzan, Y. Azar, S. Shapiro, and D. Cohen-Or, "Encoding in style: a stylegan encoder for image-to-image translation," in *Proc. CVPR*, 2021, pp. 2287–2296.
[38] O. Langner, R. Dotsch, G. Bijlstra, D. H. Wigboldus, S. T. Hawk, and A. Van Knippenberg, "Presentation and validation of the radboud faces database," *Cognition and Emotion*, vol. 24, no. 8, pp. 1377–1388, 2010.
[39] G. Perarnau, v. d. W. Joost, B. Raducanu, and M. A. Jose, "Invertible conditional gans for image editing," in *Proc. NeurIPS Workshop*, 2016.
[40] S. Zhou, T. Xiao, Y. Yang, D. Feng, Q. He, and W. He, "Genegan: Learning object transfiguration and attribute subspace from unpaired data," in *Proc. BMVC*, 2017, pp. 111.1–111.13.
[41] T. Xiao, J. Hong, and J. Ma, "Dna-gan: Learning disentangled representations from multi-attribute images," *arXiv preprint arXiv:1711.05415*, 2017.
[42] G. Zhang, M. Kan, S. Shan, and X. Chen, "Generative adversarial network with spatial attention for face attribute editing," in *Proc. ECCV*, 2018, pp. 417–432.
[43] 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 *Proc. CVPR*, 2018, pp. 8789–8797.
[44] L. Song, Z. Lu, R. He, Z. Sun, and T. Tan, "Geometry guided adversarial facial expression synthesis," in *Proc. ACM MM*, 2018, pp. 627–635.
[45] T. Li, R. Qian, C. Dong, S. Liu, Q. Yan, W. Zhu, and L. Lin, "Beautygan: Instance-level facial makeup transfer with deep generative adversarial network," in *Proc. ACM MM*, 2018, pp. 645–653.
[46] A. Pumarola, A. Agudo, A. M. Martinez, A. Sanfeliu, and F. Moreno-Noguer, "Ganimation: Anatomically-aware facial animation from a single image," in *Proc. ECCV*, 2018, pp. 818–833.
[47] T. Xiao, J. Hong, and J. Ma, "Elegant: Exchanging latent encodings with gan for transferring multiple face attributes," in *Proc. ECCV*, 2018, pp. 168–184.
[48] Z. He, W. Zuo, M. Kan, S. Shan, and X. Chen, "Attgan: Facial attribute editing by only changing what you want," *IEEE Trans. Image Process.*, vol. 28, no. 11, pp. 5464–5478, 2019.
[49] Z. He, M. Kan, J. Zhang, and S. Shan, "Pa-gan: Progressive attention generative adversarial network for facial attribute editing," *arXiv preprint arXiv:2007.05892*, 2020.

[50] Y. Nitzan, A. Bermano, Y. Li, and D. Cohen-Or, "Face identity disentanglement via latent space mapping," *arXiv preprint arXiv:2005.07728*, 2020.

[51] O. Tov, Y. Alaluf, Y. Nitzan, O. Patashnik, and D. Cohen-Or, "Designing an encoder for stylegan image manipulation," *ACM Trans. on Graphics*, vol. 40, no. 4, pp. 1–14, 2021.

[52] X. Zhu, C. Xu, and D. Tao, "Learning disentangled representations with latent variation predictability," in *Proc. ECCV*, 2020, pp. 684–700.

[53] ——, "Where and what? examining interpretable disentangled representations," in *Proc. CVPR*, 2021, pp. 5861–5870.

[54] Z. He, M. Kan, and S. Shan, "Eigengan: Layer-wise eigen-learning for gans," in *Proc. ICCV*, 2021, pp. 14 408–14 417.

[55] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proc. CVPR*, 2017, pp. 1125–1134.

[56] A. Shrivastava, T. Pfister, O. Tuzel, J. Susskind, W. Wang, and R. Webb, "Learning from simulated and unsupervised images through adversarial training," in *Proc. CVPR*, 2017, pp. 2107–2116.

[57] K. Bousmalis, N. Silberman, D. Dohan, D. Erhan, and D. Krishnan, "Unsupervised pixel-level domain adaptation with generative adversarial networks," in *Proc. CVPR*, 2017, pp. 3722–3731.

[58] T. Kim, M. Cha, H. Kim, J. K. Lee, and J. Kim, "Learning to discover cross-domain relations with generative adversarial networks," in *Proc. ICML*, 2017, pp. 1857–1865.

[59] Y. Choi, Y. Uh, J. Yoo, and J.-W. Ha, "Stargan v2: Diverse image synthesis for multiple domains," in *Proc. CVPR*, 2020, pp. 8188–8197.

[60] J. Lin, Y. Pang, Y. Xia, Z. Chen, and J. Luo, "Tuigan: Learning versatile image-to-image translation with two unpaired images," in *Proc. ECCV*, 2020, pp. 18–35.

[61] H. Chen, Y. Wang, H. Shu, C. Wen, C. Xu, B. Shi, C. Xu, and C. Xu, "Distilling portable generative adversarial networks for image translation," in *Proc. AAAI*, 2020, pp. 3585–3592.

[62] D. Bhattacharjee, S. Kim, G. Vizier, and M. Salzmann, "Dunit: Detection-based unsupervised image-to-image translation," in *Proc. CVPR*, 2020, pp. 4787–4796.

[63] Y. Liu, E. Sangineto, Y. Chen, L. Bao, H. Zhang, N. Sebe, B. Lepri, W. Wang, and M. D. Nadai, "Smoothing the disentangled latent style space for unsupervised image-to-image translation," in *Proc. CVPR*, 2021, pp. 10 785–10 794.

[64] F. Pizzati, P. Cerri, and R. d. Charette, "Comogan: continuous model-guided image-to-image translation," in *Proc. CVPR*, 2021, pp. 14 288–14 298.

[65] X. Li, S. Zhang, J. Hu, L. Cao, X. Hong, X. Mao, F. Huang, Y. Wu, and R. Ji, "Image-to-image translation via hierarchical style disentanglement," in *Proc. CVPR*, 2021, pp. 8639–8648.

[66] H. Emami, M. M. Aliabadi, M. Dong, and R. B. Chinnam, "Spagan: Spatial attention gan for image-to-image translation," *IEEE Trans. Multimedia*, vol. 23, pp. 391–401, 2020.

[67] S. Kwong, J. Huang, and J. Liao, "Unsupervised image-to-image translation via pre-trained stylegan2 network," *IEEE Trans. Multimedia*, 2021.

[68] S. Gu, Y. Li, L. V. Gool, and R. Timofte, "Self-guided network for fast image denoising," in *Proc. ICCV*, 2019, pp. 2511–2520.

[69] J. Yu, Z. Lin, J. Yang, X. Shen, X. Lu, and T. S. Huang, "Free-form image inpainting with gated convolution," in *Proc. ICCV*, 2019, pp. 4471–4480.

[70] X. Wang, K. Yu, S. Wu, J. Gu, Y. Liu, C. Dong, Y. Qiao, and C. Change Loy, "Esrgan: Enhanced super-resolution generative adversarial networks," in *Proc. ECCV Workshop*, 2018, pp. 0–0.

[71] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Proc. MICCAI*, 2015, pp. 234–241.

[72] A. Odena, C. Olah, and J. Shlens, "Conditional image synthesis with auxiliary classifier gans," in *Proc. ICML*, 2017, pp. 2642–2651.

[73] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. ICML*, 2015, pp. 448–456.

[74] A. L. Maas, A. Y. Hannun, and A. Y. Ng, "Rectifier nonlinearities improve neural network acoustic models," in *Proc. ICML*, 2013, p. 3.

[75] D. Ulyanov, A. Vedaldi, and V. Lempitsky, "Instance normalization: The missing ingredient for fast stylization," *arXiv preprint arXiv:1607.08022*, 2016.

[76] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proc. ICML*, 2010, pp. 807–814.

[77] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Proc. ICCV*, 2015, pp. 1026–1034.

[78] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. ICLR*, 2014.

[79] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. CVPR*, 2016, pp. 770–778.

[80] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proc. AISTATS*, 2010, pp. 249–256.

[81] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. ICLR*, 2014.

[82] W. Liu, Y. Wen, Z. Yu, M. Li, B. Raj, and L. Song, "Sphereface: Deep hypersphere embedding for face recognition," in *Proc. CVPR*, 2017, pp. 212–220.

[83] H. Wang, Y. Wang, Z. Zhou, X. Ji, D. Gong, J. Zhou, Z. Li, and W. Liu, "Cosface: Large margin cosine loss for deep face recognition," in *Proc. CVPR*, 2018, pp. 5265–5274.

[84] Z. Liu, P. Luo, X. Wang, and X. Tang, "Deep learning face attributes in the wild," in *Proc. ICCV*, 2015, pp. 3730–3738.

[85] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proc. CVPR*, 2016, pp. 2818–2826.

[86] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *Proc. CVPR*, 2009, pp. 248–255.

[87] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. NeurIPS*, 2012, pp. 1097–1105.

[88] J. P. Robinson, M. Shao, Y. Wu, H. Liu, T. Gillis, and Y. Fu, "Visual kinship recognition of families in the wild," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 11, pp. 2624–2637, 2018.

[89] R. Fang, A. C. Gallagher, T. Chen, and A. Loui, "Kinship classification by modeling facial feature heredity," in *Proc. ICIP*, 2013, pp. 2983–2987.

[90] L. Xie, K. Lin, S. Wang, F. Wang, and J. Zhou, "Differentially private generative adversarial network," *arXiv preprint arXiv:1802.06739*, 2018.

[91] Z. Lin, V. Sekar, and G. Fanti, "On the privacy properties of gan-generated samples," in *Proc. AISTATS*, 2021, pp. 1522–1530.

[92] D. Chen, T. Orekondy, and M. Fritz, "Gs-wgan: A gradient-sanitized approach for learning differentially private generators," in *Proc. NeurIPS*, 2020, pp. 12 673–12 684.

[93] T. Xiao, Y.-H. Tsai, K. Sohn, M. Chandraker, and M.-H. Yang, "Adversarial learning of privacy-preserving and task-oriented representations," in *Proc. AAAI*, 2020, pp. 12 434–12 441.

[94] S.-Y. Wang, O. Wang, R. Zhang, A. Owens, and A. A. Efros, "Cnn-generated images are surprisingly easy to spot... for now," in *Proc. CVPR*, 2020, pp. 8695–8704.

Yuzhi Zhao (S'19) received the B.Eng. Degree in electronic information from Huazhong University of Science and Technology, Wuhan, China, in 2018. He is currently pursuing the Ph.D. degree with the Department of Electronic Engineering, City University of Hong Kong. His research interests include image processing and deep learning, particularly generative models, image & video enhancement, and low-level & physic-based computer vision.

Lai-Man Po (M'92–SM'09) received the B.S. and Ph.D. degrees in electronic engineering from the City University of Hong Kong, Hong Kong, in 1988 and 1991, respectively. He has been with the Department of Electronic Engineering, City University of Hong Kong, since 1991, where he is currently an Associate Professor of Department of Electrical Engineering. He has authored over 150 technical journal and conference papers. His research interests include image and video coding with an emphasis deep learning based computer vision algorithms.

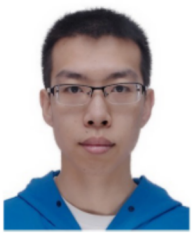
Dr. Po is a member of the Technical Committee on Multimedia Systems and Applications and the IEEE Circuits and Systems Society. He was the Chairman of the IEEE Signal Processing Hong Kong Chapter in 2012 and 2013. He was an Associate Editor of HKIE Transactions in 2011 to 2013. He also served on the Organizing Committee, of the IEEE International Conference on Acoustics, Speech and Signal Processing in 2003, and the IEEE International Conference on Image Processing in 2010.

Xuehui Wang (S'21) received the B.S. Degree from Shandong University, China, and the Master degree in the School of Computer Science from Sun Yat-sen Uninversity, China, in 2018 and 2021, respectively. He is currently pursuing the PhD degree at Artificial Intelligence Institute, Shanghai Jiao Tong University, China. His research interests include computer vision (super resolution, instance segmentation), deep learning.

Qiong Yan received her Ph.D. degree in computer science and engineering from Chinese University of Hong Kong in 2013 and the Bachelor's degree in computer science and technology from University of Science and Technology of China in 2009. She is now a research director in SenseTime, leading a group on computational imaging related research and production. Her research focuses on low-level vision tasks, such as image/video restoration and enhancement, image editing and generation.

Wei Shen is a tenure-track Associate Professor at the Artificial Intelligence Institute, Shanghai Jiao Tong University, since October 2020. Before that, he was an Assistant Research Professor at the Department of Computer Science, Johns Hopkins University. His research interests lie in the fields of computer vision, machine learning, deep learning, and medical image analysis. He serves as an Associate Editor for Neurocomputing and an Area Chair for CVPR 2022.

Yujia Zhang received the B.E. degree in electrical engineering and automation in Huazhong University of Science and Technology in 2015, and the M.S. degree in electrical engineering in South China University of Technology, China, in 2018. He is currently pursuing the Ph. D. degree in City University of Hong Kong. His current research interests include computer vision, video understanding.



Wei Liu received his B.S. and PhD degrees from Harbin Institute of Technology, Harbin, China, in 2016 and 2020, respectively. He was a visiting student in the Ohio State University for two years. He used to be an intern at SenseTime and currently works as an algorithm engineer at ByteDance. His research interests include image generation, domain adaptation, semantic segmentation and low-level computer vision. Dr. Liu serves as a Peer Reviewer for IEEE Transactions on Image Processing, ISPRS Journal of Photogrammetry and Remote Sensing, IEEE Transactions on Geoscience and Remote Sensing, etc.

Chun-Kit Wong is pursuing the B.Eng. Degree of Information Engineering in City University of Hong Kong. His research interests include deep learning and computer vision.

Chiu-Sing Pang is pursuing the B.Eng. Degree of Engineering in Information Engineering in City University of Hong Kong. His research interests include deep learning and computer vision.

Weifeng Ou received his B.Eng. degree in Telecommunication Engineering from Guangdong University of Technology in 2013, his M.Eng. degree in Signal & Information Processing from South China University of Technology in 2016, and his Ph.D. degree in the Department of Electrical Engineering from City University of Hong Kong in 2021. He was with Huawei as an R & D engineer from 2016 to 2018. He is currently working in Sensetime. His research interests include biometrics and deep learning.

Wing-Yin Yu received the B.Eng. degree in Information Engineering from City University of Hong Kong, in 2019. He is currently pursuing the Ph.D. degree at Department of Electrical Engineering at City University of Hong Kong. His research interests are deep learning and computer vision.

Buhua Liu received the B.Eng. degree in School of Electronic Information and Communications from Huazhong University of Science and Technology, Wuhan, China, in 2019. He is currently pursuing the Ph.D. degree with the Department of Computer Science, Hong Kong Baptist University. His research interests lie in the fields of AI security, privacy and computer vision, particularly in adversarial learning, federated learning.