# MatchGAN: A Self-Supervised Semi-Supervised Conditional Generative Adversarial Network

Jiaze Sun[1], Binod Bhattarai[1], and Tae-Kyun Kim[1,2]

[1] Imperial College London, Exhibition Road, London SW7 2AZ, UK
[2] Korea Advanced Institute of Science and Technology, 291 Daehak-ro, Yuseong-gu,
Daejeon 34141, Republic of Korea
{j.sun19,b.bhattarai,tk.kim}@imperial.ac.uk
https://labicvl.github.io/

**Abstract.** We present a novel self-supervised learning approach for conditional generative adversarial networks (GANs) under a semi-supervised setting. Unlike prior self-supervised approaches which often involve geometric augmentations on the image space such as predicting rotation angles, our pretext task leverages the label space. We perform augmentation by randomly sampling sensible labels from the label space of the few labelled examples available and assigning them as target labels to the abundant unlabelled examples from the same distribution as that of the labelled ones. The images are then translated and grouped into positive and negative pairs by their target labels, acting as training examples for our pretext task which involves optimising an auxiliary match loss on the discriminator's side. We tested our method on two challenging benchmarks, CelebA and RaFD, and evaluated the results using standard metrics including Fréchet Inception Distance, Inception Score, and Attribute Classification Rate. Extensive empirical evaluation demonstrates the effectiveness of our proposed method over competitive baselines and existing arts. In particular, our method surpasses the baseline with only 20% of the labelled examples used to train the baseline.

**Keywords:** Conditional generative adversarial network, self-supervised learning, semi-supervised learning, face analysis.

## 1 Introduction

Face attribute and expression editing [1,2,3,4] has attracted tremendous attention thanks to the ongoing advancements in GANs [5], in particular conditional GANs (cGANs) [6,7,1,8,9] which provide greater flexibility and control by incorporating labels in the generation process. However, deploying such cGANs in practice can be challenging as they rely heavily on large numbers of annotated examples. For instance, commonly used labelled datasets for training conditional GANs [1,10,11,7] such as CelebA and ImageNet contain examples in the order of $10^5$ to $10^6$, which might be expensive to obtain in many applications.

To reduce the need of such huge labelled datasets in training cGANs, a promising approach is to utilise self-supervised methods which are successfully

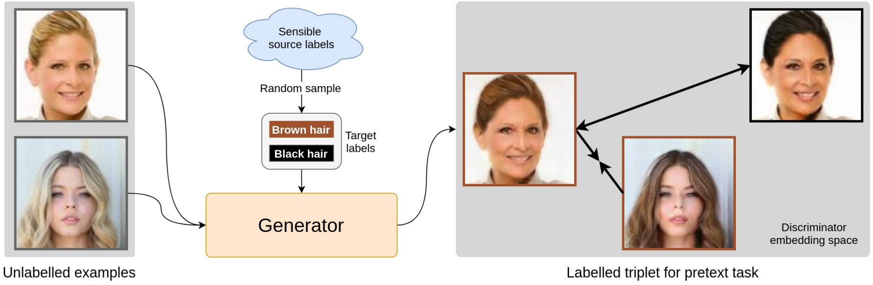

**Fig. 1.** The procedure of generating triplet examples for our pretext task.

employed in a wide range of computer vision tasks including image classification [12,13], semantic segmentation [14], robotics [15], and many more. Recently, self-supervised learning is also gaining traction with GAN training [16,17,18], but prior work in this area [16,17] has mostly focused on the input image space when designing the pretext task. For instance, [17] proposed rotating images and minimising an auxiliary rotation loss similar to that of RotNet [12], which [16] also adopted but in a semi-supervised setting. In general, existing methods mostly incorporate geometric augmentations on the input *image space* as part of the pretext task. A main limitation of such approaches is their inability to generate new examples under each class label, for example rotating an image does not change its class label. In addition, our downstream task, attribute/expression editing, is more fine-grained in nature in comparison to tasks such as image classification on ImageNet. Therefore, we present a self-supervised method for training cGANs by making use of the *label space* from the target domain and, if available, source domain as well. Specifically, under a semi-supervised setting wherein only few labelled examples are available, we utilise the large number of unlabelled examples to automatically generate additional labelled examples for our pretext task. Hence, our approach is orthogonal to existing methods.

Our idea draws inspirations from [19], a self-supervised approach for reinforcement learning which trains a policy by randomly sampling imagined goals using a variational auto-encoder (VAE) [20]. In a similar fashion, we can task the generator in a cGAN with synthesising images conditioned on randomly sampled target labels as a means to automatically provide additional supervision to the network. Motivated by our end goal of attribute editing and classification, we require that given unlabelled examples from the same distribution as the labelled ones, irrespective of their true source attribute labels, the generator should map the source images to similar regions of the synthetic image manifold if assigned the same target label and different regions otherwise. We treat every augmented target label vector as a unique state that needs to be reached on the translated domain regardless of the source labels of input images. In other words, whilst standard cGANs such as StarGAN [1] and STGAN [10] consider each component of the attribute vectors individually, our pretext task considers these vectors holistically.

Specifically, we propose to create a large pool of labelled examples by uniformly sampling labels from the source domain and assigning them to unlabelled data as their target labels. These unlabelled real images are then translated by the generator to create triplets of synthetic images as additional training examples for the generator (as illustrated in Figure 1). In addition, we also create such triplets using real images and their source labels from the small labelled pool to train the discriminator. Whilst triplets of real examples can help distill knowledge from the discriminator to the generator, both synthetic and real triplets are needed to maximise the benefits of our pretext task. The pretext task itself is trained using an auxiliary match loss optimised alongside existing losses of the baseline network [21]. This objective alleviates the overfitting problem for the discriminator in a semi-supervised setting as these triplets serve as additional supervision for the network. Unlike the standard triplet loss [22,23] which uses the Euclidean distance for comparison and fully shared weights between embeddings, we employ a learned convolutional head in the discriminator which takes *concatenated pairs* of embeddings for comparison in a manner akin to learning a custom metric. However, instead of learning a metric, we employ a cross-entropy loss to directly classify pairs with matched labels and those with mismatched ones. Compared to linear loss functions such as the hinge loss, the cross-entropy loss allows for more precise probability estimations and ultimately better performance. Unlike [17] which is purely geometric in nature, we view our approach as an image operation guided by augmented label codes performed on source images and is more in line with our end goal of attribute/expression editing.

We evaluated our method on two challenging benchmarks, CelebA and RaFD, which are popular benchmarks for facial attribute and expression translations. We take StarGAN [1] as our baseline cGAN, but our method is generic in nature. We compared the results both quantitatively and qualitatively. We used standard metrics Fréchet Inception Distance (FID) and Inception Score (IS) for quantitative comparisons.

## 2 Related Work

**Image-to-Image (I2I) Translation with cGANs.** cGANs [7] incorporate labels as additional inputs, allowing the network to handle multiple modalities and providing greater flexibility and control over generated examples. I2I translation and facial attribute editing frameworks such as Pix2pix [8] and IcGAN [24] have greatly benefited from employing cGANs, with IcGAN allowing for multi-attribute manipulation without needing to be retrained for different source-target combinations. StarGAN [1] and AttGAN [11] both improve upon IcGAN by using an end-to-end framework, an encoder-decoder architecture for the generator, and a cycle-consistency loss. STGAN [10] further improves upon these frameworks by using the difference between source and target labels as conditional input to the generator. All these methods rely heavily on source attribute labels which can be difficult to obtain in practical applications.

**Self-Supervised Learning.** Self-supervised methods have been successfully employed to fill the gap between unsupervised and supervised frameworks, particularly for image classification tasks. Well-known self-supervised approaches include predicting relative positioning of image patches [25], generating image content from surroundings [26], colouring greyscale images [27], counting visual primitives [28], and predicting rotation angles [12]. These pretext tasks all involve certain artificially designed geometric transformations on the input images. However, it might be challenging to choose the transformation most optimal for a specific task. Recently, a few approaches have been proposed which rely purely on the model's interaction with data, particularly in reinforcement learning. Grasp2Vec [15] learns object-centric visual embeddings purely through autonomous interaction between a robot and the environment. [19] uses a VAE to randomly sample imagined goals for the agent to perform. Both these frameworks serve as inspirations for MatchGAN.

**Self- and Semi-Supervised Learning in GANs.** Semi-supervised learning methods become relevant in situations where there are limited number of labelled examples and a large number of unlabelled ones. One of the popular approaches is to annotate unlabelled data with pseudo-labels [29]. Self-supervised approaches have also been explored in semi-supervised learning settings. For example, [13] employed the rotation loss [12] and outperformed fully-supervised methods with a fraction of examples labelled. As for GANs, [17] proposed to minimise the rotation loss [12] on the discriminator, mitigating the discriminator-forgetting problem and allowing more stable representations to be learned. In a semi-supervised setting, [16] proposed training an auxiliary classifier with the few labelled data which is then used to annotate the unlabelled data with pseudo-labels, and [30] differs from [16] by adding these pseudo-labels progressively and through consensus. These method, however, are reliant on the performance of the auxiliary classifier and add significant complexity to the training process.

## 3 Method

Our task is to perform I2I translation in a semi-supervised setting where the *majority* of training examples are unlabelled except for a *small* number. As training a large network in such a scenario could lead to overfitting, we aim to mitigate the problem by providing weak supervision using the large number of unlabelled examples available. In short, we propose to utilise the translated images and their associated target labels as extra training examples for a pretext task. The goal of the pretext task is to minimise an auxiliary match loss classifying positive and negative pairs of images in a manner akin to metric learning. Compared to optimising a cross-entropy loss across all possible target labels, this approach is more efficient and has been successfully adopted in one-shot learning [31] and face recognition [23]. We use StarGAN [1] as the baseline for our experiments, and as a result we will give a brief overview of its architecture and loss functions before introducing our method. However, we emphasise that our method is generic in nature and can be applied to any other cGAN.

### 3.1 Background on StarGAN

**Overview.** Here we provide a brief background on cGANs taking reference from StarGAN [1] but in a semi-supervised setting. Let $X$ be the set of source images and $Y$ the labels, where $X$ is partitioned into labelled and unlabelled subsets, $X^L$ and $X^U$, respectively. StarGAN aimed to tackle the problem of multi-domain I2I translation without having to train a new GAN for each domain pair. It accomplished this by encoding target domain information as binary or one-hot labels and feeding them along with source images to the generator. During training, the generator $G$ is required to translate a source image $x \sim X$ conditioned on a target domain label $y \sim Y$. The discriminator $D$ receives an image and produces an embedding $D_{emb}(x)$, which is then used to produce two outputs $D_{adv}(x)$ and $D_{cls}(x)$. The former, $D_{adv}(x)$, is used to optimise the Wasserstein GAN with gradient penalty [32] defined by

$$\mathcal{L}_{adv} = \mathop{\mathbb{E}}_{x\sim X; y\sim Y}[D_{adv}(G(x,y))] - \mathop{\mathbb{E}}_{x\sim X}[D_{adv}(x)] + \lambda_{gp} \mathop{\mathbb{E}}_{\hat{x}\in\hat{X}}[\|\nabla_{\hat{x}} D(\hat{x})\|_2 - 1]^2, \quad (1)$$

where $\hat{X}$ consists of points uniformly sampled from straight lines between $X$ and the synthetic distribution $G(X,Y)$. The latter output $D_{cls}(x)$ consists of probabilities over attributes/expressions used for optimising a classification loss to help guide $G$ towards generating images that more closely resemble the target domain. The classification loss for $D$ and $G$ are given by

$$\mathcal{L}^D_{cls} = \mathop{\mathbb{E}}_{x\sim X^L_y; y\sim Y}[-y \cdot \log(D_{cls}(x))], \quad (2)$$

$$\mathcal{L}^G_{cls} = \mathop{\mathbb{E}}_{x\sim (X^L\cup X^U); y\sim Y}[-y \cdot \log(D_{cls}(G(x,y)))] \quad (3)$$

respectively, where the subset $X^L_y \subset X^L$ consists of examples with label $y$. In addition, a cycle-consistency loss [33],

$$\mathcal{L}_{cyc} = \mathop{\mathbb{E}}_{x\sim X^L_y; y,y'\sim Y}[\|x - G(G(x,y'),y)\|_1], \quad (4)$$

is incorporated to ensure that $G$ preserves content unrelated to the domain translation task. The overall objective for StarGAN is given by

$$\mathcal{L}_D = \mathcal{L}_{adv} + \lambda_{cls}\mathcal{L}^D_{cls}, \qquad \mathcal{L}_G = -\mathcal{L}_{adv} + \lambda_{cls}\mathcal{L}^G_{cls} + \lambda_{cyc}\mathcal{L}_{cyc}. \quad (5)$$

**Achitectural Details.** StarGAN is fully convolutional. Its generator consists of 3 downsampling convolutional layers, 6 bottleneck residual blocks, and 3 upsampling convolutional layers. Each downsampling or upsampling layer halves or doubles the spatial dimensions of the input. Instance normalisation and ReLU is used for all layers except the output layer. The discriminator consists of 6 downsampling convolutional layers with leaky ReLUs with a slope of 0.01 for negative values. The discriminator output $D_{emb}(x)$ has 2048 channels and is then fed through two separate convolutional heads to produce $D_{adv}(x)$ and $D_{cls}(x)$, with $D_{cls}(x)$ having passed through an additional Softmax layer if ground truths are one-hot or Sigmoid layer otherwise. With an input image size of 128 × 128, StarGAN has 53.22M learned parameters in total, comprising 8.43M from the generator and 44.79M from the discriminator.

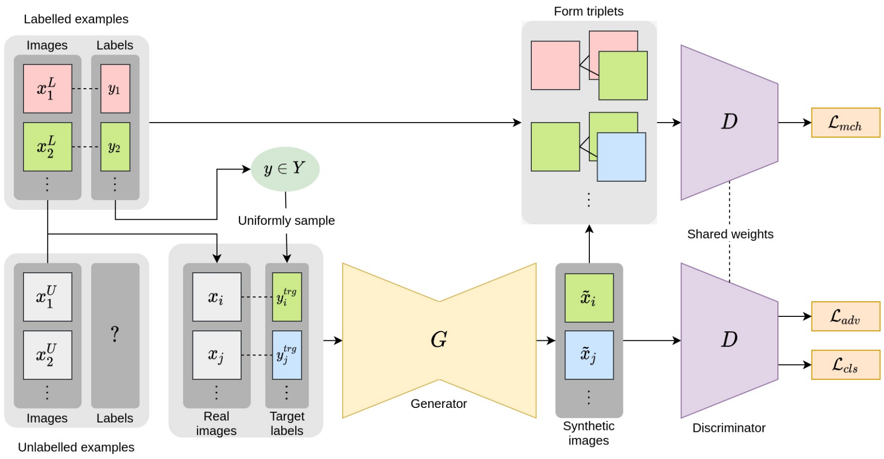

**Fig. 2.** Detailed pipeline of the proposed framework. Our contribution lies in augmenting the label space of the small-scale labelled pool followed by assigning them to the large-scale unlabelled pool as their target labels. We then generate triplets containing both matched and mismatched pairs based on the target labels in the synthetic domain and source labels in the real domain. Finally, we minimise a match loss as an auxiliary loss to the existing framework.

### 3.2 Triplet Matching Objective as Pretext Task

**Pretext from Synthetic Data.** Whilst existing self-supervised approaches mostly rely on geometric transformation of input images, we take the inspiration of utilising target domain information from reinforcement learning literature. [15] learned an embedding of object-centric images by comparing the difference prior to and after an object is grasped. However, this requires source information which can be scarce in semi-supervised learning settings. [19] utilised a variational autoencoder [20] to randomly generate a large amount of goals to train the agent in a self-supervised manner. Similar to [19], our self-supervised method involves translating unlabelled images to random target domains and using the resulting synthetic images to optimise a match loss (see Figure 2).

Recently [17] proposed to minimize the rotation loss [12] on the discriminator to mitigate its forgetting problem due to the continuously changing generator distribution. Compared to this work which involves only four rotations, the number of possible goals in our setting grows exponentially with respect to the number of attributes (CelebA is multi-labelled) and this would be challenging to implement with a softmax in the same way. Imposing a triplet-like constraint also forces the generator to maintain consistency on translated attributes, ultimately allowing attributes to be retained better on synthetic images. Hence, we propose an auxiliary match loss based on label information as a pretext task for both $G$ and $D$. A triplet consists of an anchor example $x_a$, a positive example $x_p$ which shares the same label information as $x_a$, and a negative example $x_n$ which

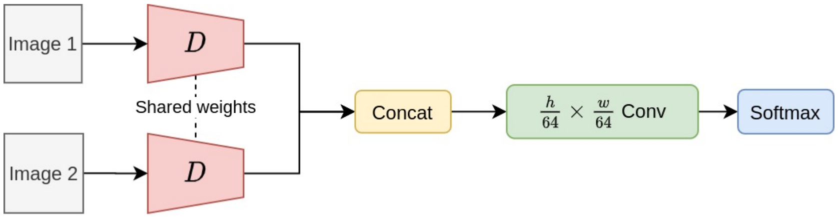

**Fig. 3.** Architecture of the match loss head, where $h$ and $w$ are the height and width of the input images.

has a different label. Unlike the standard triplet loss [22,23], we concatenate the discriminator embeddings of the positive pair $(D_{emb}(x_a), D_{emb}(x_p))$ and negative pair $(D_{emb}(x_a), D_{emb}(x_n))$ respectively along the channel axis and feed them through a single convolutional layer, producing probability distributions $D_{mch}(x_a, x_p)$ and $D_{mch}(x_a, x_n)$ respectively over whether each pair has matching labels. Specifically, we propose the following triplet matching objective

$$\mathcal{L}_{mch}^{D} = \underset{\substack{x_a, x_p \sim X_y^L; x_n \sim X_{y'}^L \\ y \neq y' \sim Y}}{\mathbb{E}} -[\log(D_{mch}(x_a, x_p)) + \log(1 - D_{mch}(x_a, x_n))], \quad (6)$$

$$\begin{aligned} \mathcal{L}_{mch}^{G} = \underset{x_1, x_2, x_3 \sim (X^L \cup X^U); y \neq y' \sim Y}{\mathbb{E}} & - [\log(D_{mch}(G(x_1, y), G(x_2, y))) \\ & + \log(1 - D_{mch}(G(x_1, y), G(x_3, y')))]. \quad (7) \end{aligned}$$

Rather than using the standard triplet loss which sticks with the Euclidean distance as a single measurement, concatenation allows the network to continuously adapt itself to the pattern in the data and thus acquire more optimal ways of carrying out such comparisons, in a manner similar to learning a custom metric[3]. In addition, the cross-entropy loss allows the network to make more precise probability estimations compared to linear loss functions and ultimately learn more refined representations. Our overall loss function is given by

$$\mathcal{L}_D = \mathcal{L}_{adv} + \lambda_{cls}\mathcal{L}_{cls}^{D} + \lambda_{mch}\mathcal{L}_{mch}^{D}, \quad (8)$$

$$\mathcal{L}_G = -\mathcal{L}_{adv} + \lambda_{cls}\mathcal{L}_{cls}^{G} + \lambda_{cyc}\mathcal{L}_{cyc} + \lambda_{mch}\mathcal{L}_{mch}^{G}. \quad (9)$$

As some of the components in Equation 8 and 9 require source labels, they cannot be directly implemented on unlabelled examples in a semi-supervised setting. As a result, we train the network with labelled and unlabelled examples in an alternating fashion as detailed in Algorithm 1.

**Architectural Details.** MatchGAN is built directly on top of StarGAN but includes an additional head for $\mathcal{L}_{mch}$ after the discriminator output $D_{emb}(x)$ (see Figure 3). Specifically, a triplet of images $(x_a, x_p, x_n)$ are passed through $D$

[3] However, concatenation does not enforce symmetry so it is not strictly a metric.

**Algorithm 1** MatchGAN.

```
Input: Labelled set $X^L$ with set of all possible class labels $Y = \{y_1, \dots, y_K\}$ separated into disjoint sets $X^L = X^L_1 \sqcup \dots \sqcup X^L_K$ by label, and unlabelled set $X^U$.
Initialise: Generator $G$, Discriminator $D$, weights $\theta_G$ and $\theta_D$, learning rates $\eta_G$ and $\eta_D$, # of iterations $N$, batch size $B$, # of $D$ updates per $G$ update $n_G$.
for $i = 1, \dots, N$ do
    if $i$ is odd then
        Form a batch of $B$ real images and labels $(R^{(i)}, Y^{(i)}_{src})$ chosen uniformly from $k$ classes $\mathcal{K}_{src} \subset \{1, \dots, K\}$, where $R^{(i)} = \bigsqcup_{j \in \mathcal{K}_{src}} R^{(i)}_j$ and each $R^{(i)}_j \subset X^L_j$.
        $\mathcal{L}^D_{cls} \leftarrow \frac{1}{B} \sum_{(r,y) \in (R^{(i)}, Y^{(i)}_{src})} -y \cdot \log(D_{cls}(r))$.
        Get $T^{(i)}_R = \{(x_a, x_p, x_n) : x_a, x_p \in R^{(i)}_{k_1} \text{ and } x_n \in R^{(i)}_{k_2}, k_1 \neq k_2 \in \mathcal{K}_{src}\}$, a set of triplets sampled from the mini-batch $R^{(i)}$.
        $\mathcal{L}^D_{mch} \leftarrow \frac{1}{|T^{(i)}_R|} \sum_{(x_a, x_p, x_n) \in T^{(i)}_R} -[\log(D_{mch}(x_a, x_p)) + \log(1 - D_{mch}(x_a, x_n))]$.
    else
        Sample mini-batch of $B$ unlabelled real images $R^{(i)} \subset X^U$.
    end if
    Form a batch of $B$ target labels $Y^{(i)}_{trg}$ chosen uniformly from $k$ classes $\mathcal{K}_{trg} \subset \{1, \dots, K\}$.
    Generate fake images $F^{(i)} = \{G(r, y) : (r, y) \in (R^{(i)}, Y^{(i)}_{trg})\}$.
    $\mathcal{L}^D_{adv} \leftarrow \frac{1}{B} \sum_{(r,f) \in (R^{(i)}, F^{(i)})} [D_{adv}(f) - D_{adv}(r) + \lambda_{gp}(\|\nabla_{\hat{x}} D_{adv}(\hat{x})\|_2 - 1)^2]$, where $\hat{x} = \alpha r + (1 - \alpha) f$ and $\alpha \sim U(0, 1)$ is random.
    $\theta_D \leftarrow Adam\left(\nabla_{\theta_D}(\mathcal{L}^D_{adv} + \mathrm{odd}(i)(\lambda_{cls}\mathcal{L}^D_{cls} + \lambda_{mch}\mathcal{L}^D_{mch})), \eta_D\right)$ using Adam [34], where $odd(i) = 1$ if $i$ is odd or 0 otherwise.
    if $i$ is a multiple of $n_G$ then
        if $i$ is odd then
            $\mathcal{L}_{cyc} \leftarrow \frac{1}{B} \sum_{(r,y,y') \in (R^{(i)}, Y^{(i)}_{src}, Y^{(i)}_{trg})} \|r - G(G(r, y'), y)\|_1$.
        end if
        $\mathcal{L}^G_{adv} \leftarrow \frac{1}{B} \sum_{f \in F^{(i)}} -D_{adv}(f)$.
        $\mathcal{L}^G_{cls} \leftarrow \frac{1}{B} \sum_{(f,y) \in (F^{(i)}, Y^{(i)}_{trg})} -y \cdot \log(D_{cls}(G(f, y)))$.
        Get $T^{(i)}_F = \{(x_a, x_p, x_n) : x_a, x_p \in F^{(i)}_{k_1} \text{ and } x_n \in F^{(i)}_{k_2}, k_1 \neq k_2 \in \mathcal{K}_{trg}\}$, a set of triplets sampled from the mini-batch $F^{(i)}$, where $F^{(i)} = \bigsqcup_{j \in \mathcal{K}_{trg}} F^{(i)}_j$ and each $F^{(i)}_j$ corresponds to target label $y_j$.
        $\mathcal{L}^G_{mch} \leftarrow \frac{1}{|T^{(i)}_F|} \sum_{(x_a, x_p, x_n) \in T^{(i)}_F} -[\log(D_{mch}(x_a, x_p)) + \log(1 - D_{mch}(x_a, x_n))]$.
        $\theta_G \leftarrow Adam\left(\nabla_{\theta_G}(\mathcal{L}_{adv} + \lambda_{cls}\mathcal{L}^G_{cls} + \lambda_{mch}\mathcal{L}^G_{mch} + \mathrm{odd}(i)\lambda_{cyc}\mathcal{L}_{cyc}), \eta_G\right)$.
    end if
end for
Output: Optimal $G$.
```

to produce embeddings $(D_{emb}(x_a), D_{emb}(x_p), D_{emb}(x_n))$. The positive and negative pairs $(D_{emb}(x_a), D_{emb}(x_p))$ and $(D_{emb}(x_a), D_{emb}(x_n))$ are concatenated respectively along the channel dimension to produce 4096-channel embeddings.

These embeddings are then convolved and passed through a Softmax layer to produce probabilities of whether each image pair is matched. For input images of size 128 × 128, this head adds approximately 32.77K to the total number of learned parameters which is negligible compared to the 53.22M parameters in the StarGAN baseline, and thus has very little impact on training efficiency.

## 4 Experiments

### 4.1 Implementation details

We used StarGAN [1] as a baseline for our experiments[4]. StarGAN unifies multi-domain image-to-image translation with a single generative network and is well suited to our label-based self-supervised approach. However, we would like to re-emphasise that our method is a general idea and can be extended to other cGANs. To avoid potential issues during training, we used the same hyperparameters as the original StarGAN. Specifically, we trained the network for 200K discriminator iterations with 1 generator update after every 5 discriminator updates. We used the Adam optimiser [34] with $\beta_1 = 0.5$ and $\beta_2 = 0.999$, and the initial learning rates for both generator and discriminator were set to $10^{-4}$ for the first 100K iterations and decayed linearly to 0 for the next 100K. We trained the model using mini-batches of 16 examples (sampled from 4 random classes if from the labelled pool) and mapped to 4 random target classes. Training took approximately 10 hours to complete on an NVIDIA RTX 2080Ti GPU.

### 4.2 Datasets

We evaluated our method on two challenging face attributes and expression manipulation datasets, The CelebFaces Attributes Dataset (CelebA) [35] and The Radboud Faces Database (RaFD) [36]. Both datasets were split into training and test sets, and we report results on the test set.

**CelebA.** CelebA contains 202,599 images of celebrities of size $178 \times 218$ with 40 attribute annotations. We selected 5 attributes including 3 hair colours (black, blond, and brown), gender, and age. The images were cropped to $178 \times 178$ then resized to 128 × 128. The experiments followed the official partition of 162,770 examples for training and 19,962 for testing. We created a semi-supervised scenario with limited labelled training examples by uniformly sub-sampling a percentage of training examples as labelled and setting the rest as unlabelled. The sub-sampling process was done to ensure that the examples were spread evenly between classes whenever possible to avoid potential class imbalance issues.

**RaFD.** RaFD is a much smaller dataset with 8,040 images of size $681 \times 1024$ of 67 identities of different genders, races, and ages displaying 8 emotional expressions. The images were cropped to 600 × 600 (centred on face) before being resized to $128 \times 128$. A total of 7 randomly selected identities, comprising 840 images, were chosen as the test set and the rest (60 identities comprising 7200 images) as

[4] Code and pretrained model at `https://github.com/justin941208/MatchGAN`.

the training set. Similar to CelebA, a semi-supervised setting was created by splitting the training set into labelled and unlabelled pools.

### 4.3 Baseline

Our baseline was established by setting $\lambda_{mch}$ to 0 whilst leaving all other procedures unchanged. As for MatchGAN, the value of $\lambda_{mch} = 0.5$ was used for all experiments. To verify that our method is scalable to both small and large number of annotated data, we tested our approach with various percentages of training examples labelled. Specifically, we performed experiments setting 1%, 5%, 10%, and 20% of CelebA training data as labelled examples, and similarly for 10%, 20%, and 50% of RaFD training data as RaFD is a significantly smaller dataset. Finally, we also evaluated our method on the full datasets to verify the effectiveness of our method on benchmarks designed for supervised learning. We also tested the rotation loss [17] for comparison.

### 4.4 Evaluation metrics

We employed the Fréchet Inception Distance (FID) [37] and Inception Score (IS) [38] for quantitative evaluations. FID measures the similarity between the distribution of real examples and that of the generated ones by comparing their respective embeddings from a pretrained Inception-v3 network [39]. IS also measures image quality but relies on the probability outputs of the same Inception-v3 network, taking into account the meaningfulness of individual images and the diversity of all images. If the generated images are of high quality, then FID should be small whereas IS should be large. We computed the FID by translating test images to each of the target attribute domains (5 for CelebA, 8 for RaFD) and comparing the distributions before and after translations. The IS was computed as an average obtained from a 10-fold split of the test set.

In addition to FID and IS, we also used GAN-train and GAN-test [40] to measure the attribute classification rate of translated images. In short, given a set of real images $X$ with a train-test split $X = X_{train} \sqcup X_{test}$, GAN-train is the accuracy obtained from a classifier trained on synthetic images $G(X_{train})$ and tested on real images $X_{test}$, whereas for GAN-test the classifier is trained on real images $X_{train}$ and tested on synthetic images $G(X_{test})$.

### 4.5 Ablation studies

MatchGAN involves extracting triplets from labelled real examples and all synthetic examples - labelled and unlabelled. To show that the proposed method does not simply rely on the few labelled examples and that both unlabelled and synthetic examples are necessary to achieve good performance, the network was trained in several other scenarios in which various amounts of real and synthetic data used for updating the match loss $\mathcal{L}_{mch}$ were removed (shown in Table 1). A few observations can be made from this table. First, including a large number of unlabelled data is essential for improving performance, which is clear from comparing A and B with the rest. Second, incorporating the match loss $\mathcal{L}_{mch}$ provides substantial improvement in performance, as observed from A vs B, and C vs D–G. This improvement was achieved despite the match loss not utilising all the available data, as seen from setups D–F. Third, match loss indeed benefits from training with synthetic examples which is evident from C vs D, and E vs F–G. Fourth, unlabelled synthetic examples can be used to achieve further performance improvement, as seen from F vs G. In addition, H was trained using the standard triplet loss [23] which G also outperforms. Therefore, G will be used as the default setup for MatchGAN in all following experiments.

**Table 1.** The results of the ablation studies – FID scores obtained using various amounts of training data. Setups A and C are baseline StarGAN [1], whereas the other setups update the match loss $\mathcal{L}_{mch}$ using different portions of the training data.

| Setup | | A | B | C | D | E | F | G | H[23] |
|---|---|---|---|---|---|---|---|---|---|
| Total number of training examples | | 2.5K | 2.5K | 162K | 162K | 162K | 162K | 162K | 162K |
| Number of examples for $\mathcal{L}_{mch}$ | Real (labelled) | 0 | 2.5K | 0 | 0 | 2.5K | 2.5K | 2.5K | 2.5K |
| | Synthetic (labelled) | 0 | 2.5K | 0 | 2.5K | 0 | 2.5K | 2.5K | 2.5K |
| | Synthetic (unlabelled) | 0 | 0 | 0 | 160K | 0 | 0 | 160K | 160K |
| FID↓ | | 24.20 | 17.26 | 16.11 | 13.78 | 13.66 | 10.88 | **9.43** | 14.86 |

### 4.6 Quantitative evaluations

We evaluated the performance of our proposed method, the baseline, and rotation loss [17] using FID and IS and the results are shown in Table 2. In terms of FID, MatchGAN consistently outperformed the baseline in both CelebA and RaFD. For CelebA in particular, with just 20% of training examples labelled, our method was able to achieve better performance than the baseline with 100% of the training examples labelled. Our method also has a distinct lead over the baseline when there are very few labelled examples. In addition, our method was also on par with or even outperformed rotation loss in both datasets, again with a distinct advantage over rotation loss when labelled examples are limited.

In terms of IS, we still managed to outperform both the baseline and rotation loss in the majority of the setups. In other setups our method was either on par with the baseline or slightly underperforming within a margin of 0.02. We would like to emphasise that IS is less consistent than FID as it does not compare the synthetic distribution with an "ideal" one. In addition, IS is computed using the 1000-dimensional output of Inception-v3 pretrained on ImageNet which is arguably less suitable for human face datasets such as CelebA and RaFD. However, we included IS here as it is still one of the most widely used metrics for evaluating the performance of GANs.

In terms of GAN-train and GAN-test classification rates, our method outperformed the baseline in both CelebA and RaFD (shown in Table 3) under the

**Table 2.** Baseline vs Rotation vs MatchGAN in terms of FID and IS scores.

| Dataset | Metric | Setup | Percentage of training data labelled | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | | 1% | 5% | 10% | 20% | 50% | 100% |
| CelebA | FID↓ | Baseline [1] | 17.04 | 10.54 | 9.47 | 7.07 | / | 6.65 |
| | | Rotation [17] | 17.08 | 10.00 | **8.04** | 6.82 | / | 5.91 |
| | | MatchGAN | **12.31** | **9.34** | 8.81 | **6.34** | / | **5.58** |
| | IS↑ | Baseline [1] | 2.86 | 2.95 | **3.00** | 3.01 | / | 3.01 |
| | | Rotation [17] | 2.82 | **2.99** | 2.96 | 3.01 | / | 3.06 |
| | | MatchGAN | **2.95** | 2.95 | 2.99 | **3.03** | / | **3.07** |
| RaFD | FID↓ | Baseline [1] | / | / | 32.015 | 11.75 | 7.24 | 5.14 |
| | | Rotation [17] | / | / | 28.88 | 10.96 | **6.57** | **5.00** |
| | | MatchGAN | / | / | **22.75** | **9.94** | 6.65 | 5.06 |
| | IS↑ | Baseline [1] | / | / | **1.66** | 1.60 | 1.58 | 1.56 |
| | | Rotation [17] | / | / | 1.62 | 1.58 | 1.58 | **1.60** |
| | | MatchGAN | / | / | 1.64 | **1.61** | **1.59** | 1.58 |

100% setup which has the best FID overall. MatchGAN again obtained a higher GAN-train accuracy than the baseline, indicating that the synthetic examples generated by MatchGAN can be more effectively used to augment small data for training classifiers. We report the results under the 100% setup as it has the lowest FID and that FID is considered one of the most robust metrics for evaluating the performance of GANs. We expect GAN-train and GAN-test in other setups to be proportional to their respective FIDs as well.

**Table 3.** Baseline vs MatchGAN in terms of GAN-train and GAN-test classification rate under the 100% setup. GAN-train for CelebA and GAN-test were obtained by averaging individual attribute accuracies, whereas top-1 accuracy was used when computing GAN-train for RaFD.

| Dataset | Setup | GAN-train | GAN-test |
|---|---|---|---|
| CelebA | Baseline [1] | 87.29% | 81.11% |
| | MatchGAN | **87.43%** | **82.26%** |
| RaFD | Baseline [1] | 95.00% | 75.00% |
| | MatchGAN | **97.78%** | **75.95%** |

### 4.7 Qualitative evaluations

Figure 4 and 5 compare the visual quality of the images generated by Baseline and MatchGAN on CelebA and RaFD respectively. MatchGAN can be observed to produce images that are less noisy, less blurry, and more coherent. For instance, in the 1% setup in Figure 4, the baseline can often be observed to produce artefacts, blurry patches, or incomplete translations (e.g. the brown patch in the hair in the fourth row) which are not present in the images generated by MatchGAN. Similarly on RaFD, MatchGAN generates more coherent

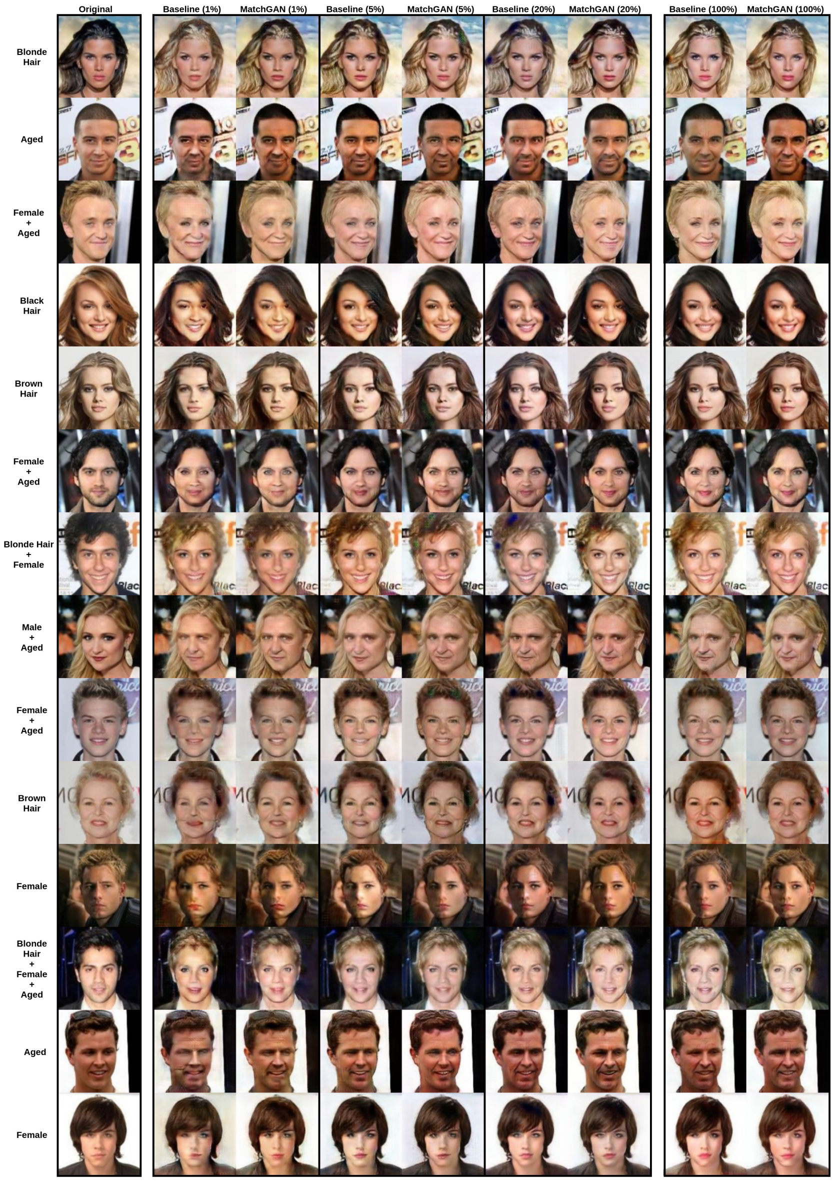

**Fig. 4.** Synthetic examples of MatchGAN vs Baseline on CelebA (zoom in for a better view). Each row corresponds to a single- or multi-attribute manipulation, with target attributes listed on the left side.

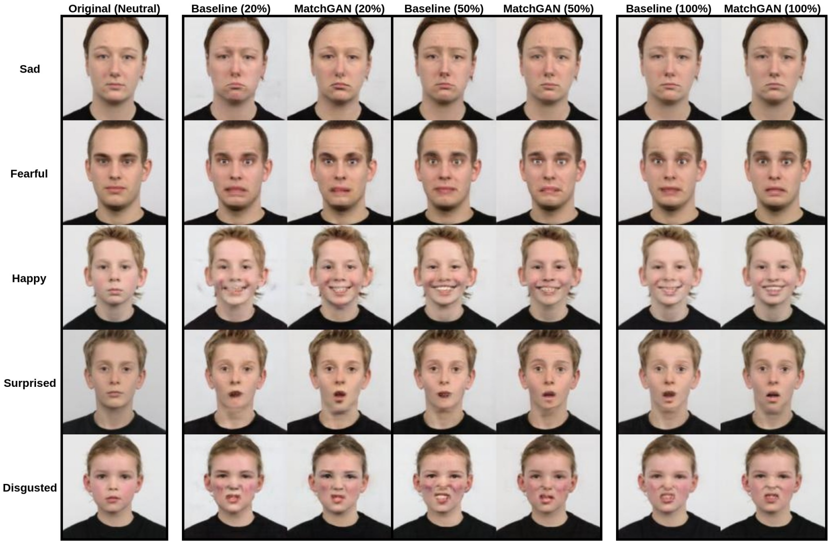

**Fig. 5.** Synthetic examples of MatchGAN vs Baseline on RaFD (zoom in for a better view). Each row corresponds to a single expression manipulation, with target expression listed on the left side.

expressions compared to the baseline (e.g. the "surprised mouth" in the fourth row in Figure 5) and produces fewer artefacts. The image quality of our method also improves substantially with more labelled examples. In Figure 4, the overall quality of the images generated by MatchGAN in the 20% setup is on par with or even outmatches that of the Baseline under the 100% setup in terms of clarity, colour tone, and coherence of target attributes, corroborating our quantitative results shown in Table 2.

## 5 Conclusion

In this paper we proposed MatchGAN, a novel self-supervised learning approach for training conditional GANs under a semi-supervised setting with very few labelled examples. MatchGAN utilises synthetic examples and their target labels as additional annotated examples and minimises a triplet matching objective as a pretext task. With 20% of the training data labelled, it is able to outperform the baseline trained with 100% of examples labelled and shows a distinct advantage over other self-supervised approaches such as [17] under both fully-supervised and semi-supervised settings.

**Acknowledgements.** This work is supported by the Huawei Consumer Business Group, Croucher Foundation, and EPSRC Programme Grant 'FACER2VM' (EP/N007743/1).

## References

1. Choi, Y., Choi, M., Kim, M., Ha, J.W., Kim, S., Choo, J.: Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. In: CVPR. (2018)
2. Chen, X., Duan, Y., Houthooft, R., Schulman, J., Sutskever, I., Abbeel, P.: Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In: NIPS. (2016)
3. Pumarola, A., Agudo, A., Martinez, A.M., Sanfeliu, A., Moreno-Noguer, F.: Ganimation: Anatomically-aware facial animation from a single image. In: ECCV. (2018)
4. Karras, T., Laine, S., Aila, T.: A style-based generator architecture for generative adversarial networks. In: CVPR. (2019)
5. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: NeurIPS. (2014)
6. Odena, A., Olah, C., Shlens, J.: Conditional image synthesis with auxiliary classifier gans. In: ICML. (2017)
7. Mirza, M., Osindero, S.: Conditional generative adversarial nets. arXiv preprint arXiv:1411.1784 (2014)
8. Isola, P., Zhu, J.Y., Zhou, T., Efros, A.A.: Image-to-image translation with conditional adversarial networks. In: CVPR. (2017)
9. Bhattarai, B., Kim, T.K.: Inducing optimal attribute representations for conditional gans. In: ECCV. (2020)
10. Liu, M., Ding, Y., Xia, M., Liu, X., Ding, E., Zuo, W., Wen, S.: Stgan: A unified selective transfer network for arbitrary image attribute editing. In: CVPR. (2019)
11. He, Z., Zuo, W., Kan, M., Shan, S., Chen, X.: Attgan: Facial attribute editing by only changing what you want. IEEE TIP (2019)
12. Gidaris, S., Singh, P., Komodakis, N.: Unsupervised representation learning by predicting image rotations. In: ICLR. (2018)
13. Zhai, X., Oliver, A., Kolesnikov, A., Beyer, L.: S4l: Self-supervised semi-supervised learning. In: ICCV. (2019)
14. Zhan, X., Liu, Z., Luo, P., Tang, X., Loy, C.C.: Mix-and-match tuning for self-supervised semantic segmentation. In: AAAI. (2018)
15. Jang, E., Devin, C., Vanhoucke, V., Levine, S.: Grasp2vec: Learning object representations from self-supervised grasping. In: CRL. (2018)
16. Lučić, M., Tschannen, M., Ritter, M., Zhai, X., Bachem, O., Gelly, S.: High-fidelity image generation with fewer labels. In: ICML. (2019)
17. Chen, T., Zhai, X., Ritter, M., Lucic, M., Houlsby, N.: Self-supervised gans via auxiliary rotation loss. In: CVPR. (2019)
18. Tran, N.T., Tran, V.H., Nguyen, B.N., Yang, L., et al.: Self-supervised gan: Analysis and improvement with multi-class minimax game. In: NeurIPS. (2019)
19. Nair, A.V., Pong, V., Dalal, M., Bahl, S., Lin, S., Levine, S.: Visual reinforcement learning with imagined goals. In: NIPS. (2018)
20. Kingma, D.P., Welling, M.: Auto-encoding variational bayes. In: ICLR. (2014)
21. Caruana, R.: Multitask learning. Machine Learning (1997)
22. Weinberger, K.Q., Blitzer, J., Saul, L.K.: Distance metric learning for large margin nearest neighbor classification. In: NIPS. MIT Press (2006)
23. Schroff, F., Kalenichenko, D., Philbin, J.: Facenet: A unified embedding for face recognition and clustering. In: CVPR. (2015)

24. Perarnau, G., Van De Weijer, J., Raducanu, B., Álvarez, J.M.: Invertible conditional gans for image editing. In: NIPSW. (2016)
25. Doersch, C., Gupta, A., Efros, A.A.: Unsupervised visual representation learning by context prediction. In: CVPR. (2015)
26. Pathak, D., Krahenbuhl, P., Donahue, J., Darrell, T., Efros, A.A.: Context encoders: Feature learning by inpainting. In: CVPR. (2016)
27. Zhang, R., Isola, P., Efros, A.A.: Colorful image colorization. In: ECCV. (2016)
28. Noroozi, M., Pirsiavash, H., Favaro, P.: Representation learning by learning to count. In: ICCV. (2017)
29. Lee, D.H.: Pseudo-label : The simple and efficient semi-supervised learning method for deep neural networks. In: ICML. (2013)
30. Wang, Y., Khan, S., Garcia, A.G., Weijer, J.v.d., Khan, F.S.: Semi-supervised learning for few-shot image-to-image translation. CVPR (2020)
31. Koch, G., Zemel, R., Salakhutdinov, R.: Siamese neural networks for one-shot image recognition. In: ICML 2015 Deep Learning Workshop. (2015)
32. Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., Courville, A.: Improved training of wasserstein gans. In: NIPS. (2017)
33. Zhu, J.Y., Park, T., Isola, P., Efros, A.A.: Unpaired image-to-image translation using cycle-consistent adversarial networks. In: ICCV. (2017)
34. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. In: ICLR. (2015)
35. Liu, Z., Luo, P., Wang, X., Tang, X.: Deep learning face attributes in the wild. In: ICCV. (2015)
36. Langner, O., Dotsch, R., Bijlstra, G., Wigboldus, D., Hawk, S., van Knippenberg, A.: Presentation and validation of the radboud faces database. Cognition and Emotion **24** (2010)
37. Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., Hochreiter, S.: Gans trained by a two time-scale update rule converge to a local nash equilibrium. In: NeurIPS. (2017)
38. Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., Chen, X.: Improved techniques for training gans. In: NIPS. (2016)
39. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: Rethinking the inception architecture for computer vision. In: CVPR. (2016)
40. Shmelkov, K., Schmid, C., Alahari, K.: How good is my gan? In: ECCV. (2018)