A Missing details

A.1 Proof for Corollary 1

Proof. For every cluster i, we have:

$$\begin{aligned} L_i &= \int_x w_i P_r(x) \log D_i(x) + P_{f_i}(x) \log(1 - D_i(x)) dx \\ &= \int_x P_{r_i}(x) \log D_i(x) + P_{f_i}(x) \log(1 - D_i(x)) dx \ . \end{aligned}$$

Let $\frac{\partial L_i}{\partial D_i} = 0$, we have:

$$D_i^*(x) = \frac{P_{r_i}(x)}{P_{r_i}(x) + P_{f_i}(x)}.$$

Then we substitute this optimal into the summation in Eq. 11:

$$\begin{aligned} L =& \int_x P_{r_i}(x) \log \frac{P_{r_i}(x)}{P_{r_i}(x) + P_{f_i}(x)} \\ &+ P_{f_i}(x) \log \frac{P_{f_i}(x)}{P_{r_i}(x) + P_{f_i}(x)} dx \\ =& \int_x P_{r_i}(x) \log \frac{P_{r_i}(x)}{(P_{r_i}(x) + P_{f_i}(x))/2} \\ &+ P_{f_i}(x) \log \frac{P_{f_i}(x)}{(P_{r_i}(x) + P_{f_i}(x))/2} dx - (w_i + 1) \log 2 \\ =& KL(P_{r_i} \parallel \frac{P_{r_i} + P_{f_i}}{2}) + KL(P_{f_i} \parallel \frac{P_{r_i} + P_{f_i}}{2}) \\ &- (w_i + 1) \log 2 \\ =& - (w_i + 1) \log 2 + 2JSD(P_{r_i} \parallel P_{f_i}) \ . \end{aligned}$$

Since we already have the optimal solution for all clusters as given in Eq. 13, we can say that for each cluster, we minimize the JS divergence between real data and fake data. When summed up, the optimal remains since each term is independent with others. □

A.2 Derivation for ϕ_i^*

This result can be easily obtained by solving the optimization problem:

$$\begin{aligned} \max_{\phi} \quad & \mathop{\mathbb{E}}_{x \sim P_{r_i}} \log \phi_i \ , \\ \text{s.t.} \quad & \sum_{i=1}^{K} \phi_i = 1 \ . \end{aligned}$$

We can derive that

$$\phi_i = \frac{\sum_x P(z|x, \theta^{old})}{\sum_z \sum_x P(z|x, \theta^{old})} = \frac{N_i}{N} \ .$$

A.3 Results on all 40 CelebA attributes

Here in Tab. 1 we show the experiment results of GAN-EM on all 40 CelebA attributes with both unsupervised and semi-supervised settings. We bold top three results for unsupervised clustering and semi-supervised classification respectively.

B Implementation details

B.1 MNIST

MNIST dataset has 60,000 training samples, among which we assign 10,000 samples as validation data. The images are 28×28 pixels of which the values are normalized to [0, 1]. The input of the generator is 72 dimension (62 dimension random noise and 10 dimension conditional class label)The generator has one fully connected layer with 6272 hidden units ($128 \times 7 \times 7$) followed by 2 transpose convolutional layers. The first transpose convolutional layer has 64 filters of which the size is 4×4. The convolutional operation is stride 2 with 1 zero padding. The second convoluational layer uses the same structure with the first one except that the number of filters is 1. Then the output of the generator is in size 28×28 which is the same as the real image size. The discriminator also has 3 convolutional layers with 64, 128, 256 filters respectively. All of these 3 layers uses kernel size 4, stride 2 and zero padding 1. Then 2 fully connected layers are used with 1024 and 11 hidden units respectively. E-net shares the same structure with the discriminator except that the number of units in last layer is 10. Tab. 2 describes the network structure in detail.

We apply RMSprop optimizer to all these 3 networks with learning rate 0.0002 (decay rate: 0.98). The random noise of generator is in uniform distribution. In each M-step, there are 5 epoches with a minibatch size of 70 for both the generated batch and the real samples batch. We use a same update frequency for generator and discriminator. For E-step, we generate samples using well trained generator with batch size of 200, then we apply 1000 iterations to update E-net.

For semi-supervised classification, we add the supervision to both E-net. At the end of E-step, we train the E-net by labeled data until the prediction accuracy given by E-net is 100%. Then we feed the real data to E-net to obtain $P(c|x_{real})$ which is the same as unsupervised clustering.

B.2 SVHN

Similar to MNIST, SVHN is also a digits recognition datasets with about 53,000 training images (32 pixels). Since the images of SVHN are all from the real world and many of them are blurry, the recognition problem is much more difficult than MNIST which is preprocessed to grey-scale images. We normalize the images to [-1, 1]. We use almost the same network structure with MNIST with only a slight difference. The details of GAN are in Tab. 3.

For semi supervision, we use the same strategies with MNIST which is discussed above.

B.3 CelebA

CelebA dataset is a large-scale human face dataset with more than 200k images. Each image is annotated by 40 binary attributes. For example, for gender attribute, we first select all 12k male images, then we select the same amount of female images. The selection of female images is based on such principle: all other attributes except for gender should maintain as much purity as possible. Since we are unable to guarantee all the other attributes are 100% pure, we regard those impure attributes as noise. The selected images are cropped to 64×64

Table 1: Error rates of GAN-EM on all 40 CelebA attributes

Attributes	Unsupervised	Semi-supervised (100 labels)		Attributes	Unsupervised	Semi-supervised (100 labels)
Goatee	0.4103	0.2218		Bald	0.4855	0.4001
Narrow_Eyes	0.3308	0.1757		Arched_Eyebrows	0.4447	0.2993
Bangs	0.3592	0.1506		Wavy_Hair	0.3376	0.2657
Gray_Hair	0.4538	0.1849		Mouth_Slightly_Open	0.4272	0.1895
Big_Lips	0.4804	0.1767		Young	0.4557	0.3216
Heavy_Makeup	0.2366	0.2218		No_Beard	0.4869	0.4315
Attractive	0.4749	0.3736		Pointy_Nose	0.3426	0.1847
Bags_Under_Eyes	0.4468	0.3545		Bushy_Eyebrows	0.4447	0.3592
High_Cheekbones	0.4455	0.2637		Double_Chin	0.4546	0.3294
Oval_Face	0.4858	0.2583		gender	0.3683	0.1446
Rosy_Cheeks	0.3518	0.2891		hat	0.2941	0.1598
Sideburns	0.4726	0.2507		glass	0.2983	0.1662
Mustache	0.4539	0.2947		Male	0.4763	0.2746
Brown_Hair	0.4463	0.4362		Receding_Hairline	0.4847	0.4138
Pale_Skin	0.4736	0.4457		Wearing_Necklace	0.4758	0.2755
Chubby	0.4478	0.3242		Wearing_Necktie	0.4805	0.4157
Big_Nose	0.4695	0.4337		Wearing_Lipstick	0.4141	0.1601
Blurry	0.3045	0.2188		Straight_Hair	0.4687	0.3102
Black_Hair	0.4555	0.3949		Wearing_Earrings	0.4393	0.3686
Blond_Hair	0.3145	0.3057		5_o_Clock_Shadow	0.4435	0.2822

Table 2: GAN structure for MNIST.

Generator	Discriminator	E-net
Input 72 dim vector	Input 28×28	Input 28×28
FC ($128 \times 7 \times 7$)	4×4 Conv, f:64 s:2 d:1	4×4 Conv, f:64 s:2 d:1
BN lRelu(0.2)	BN lRelu(0.2)	BN lRelu(0.2)
4×4 Deconv, f:64 s:2 d:1	4×4 Conv, f:128 s:2 d:1	4×4 Conv, f:128 s:2 d:1
BN lRelu(0.2)	BN lRelu(0.2)	BN lRelu(0.2)
4×4 Deconv, f:1 s:2 d:1	4×4 Conv, f:256 s:2 d:1	4×4 Conv, f:256 s:2 d:1
	BN lRelu(0.2)	BN lRelu(0.2)
	FC(1024), BN lRelu(0.2)	FC(1024), BN lRelu(0.2)
	FC(11)	FC(10)

BN: batch normalization, FC: fully connected layer. f: filter number. s: stride size. d: padding size

Table 3: GAN structure for SVHN.

Generator	Discriminator	E-net
Input 72 dim vector	Input 28×28	Input 28×28
FC ($128 \times 4 \times 4$)	4×4 Conv, f:64 s:2 d:1	4×4 Conv, f:64 s:2 d:1
BN lRelu(0.2)	BN lRelu(0.2)	BN lRelu(0.2)
4×4 Deconv, f:64 s:2 d:1	4×4 Conv, f:128 s:2 d:1	4×4 Conv, f:128 s:2 d:1
BN lRelu(0.2)	BN lRelu(0.2)	BN lRelu(0.2)
4×4 Deconv, f:32 s:2 d:1	4×4 Conv, f:256 s:2 d:1	4×4 Conv, f:256 s:2 d:1
BN lRelu(0.2)	BN lRelu(0.2)	BN lRelu(0.2)
4×4 Deconv, f:3 s:2 d:1	FC(1024), BN lRelu(0.2)	FC(1024), BN lRelu(0.2)
	FC(11)	FC(10)

BN: batch normalization, FC: fully connected layer. f: filter number. s: stride size. d: padding size

pixels and are normalized to [-1, 1]. We apply the same strategy for the other attributes. The network details are in Tab. 4.

Table 4: GAN structure for CelebA.

Generator	Discriminator	E-net
Input 300 dim vector	Input 64×64	Input 64×64
4×4 Deconv, f:1024 s:1 d:0	4×4 Conv, f:128 s:2 d:1	4×4 Conv, f:128 s:2 d:1
BN lRelu(0.2)	BN lRelu(0.2)	BN lRelu(0.2)
4×4 Deconv, f:512 s:1 d:0	4×4 Conv, f:256 s:2 d:1	4×4 Conv, f:256 s:2 d:1
BN lRelu(0.2)	BN lRelu(0.2)	BN lRelu(0.2)
4×4 Deconv, f:256 s:1 d:0	4×4 Conv, f:512 s:2 d:1	4×4 Conv, f:512 s:2 d:1
BN lRelu(0.2)	BN lRelu(0.2)	BN lRelu(0.2)
4×4 Deconv, f:128 s:1 d:0	4×4 Conv, f:1024 s:2 d:1	4×4 Conv, f:1024 s:2 d:1
BN lRelu(0.2)	BN lRelu(0.2)	BN lRelu(0.2)
4×4 Deconv, f:3 s:1 d:0	4×4 Conv, f:3 s:1 d:0	4×4 Conv, f:3 s:1 d:0
	BN lRelu(0.2)	BN lRelu(0.2)
	FC(3)	FC(2)

BN: batch normalization, FC: fully connected layer. f: filter number. s: stride size. d: padding size