# Supplementary Material: Learning Compositional Visual Concepts with Mutual Consistency

Yunye Gong[1], Srikrishna Karanam[3], Ziyan Wu[3], Kuan-Chuan Peng[3], Jan Ernst[3], and Peter C. Doerschuk[1,2]
[1]School of Electrical and Computer Engineering, Cornell University, Ithaca NY
[2]Nancy E. and Peter C. Meinig School of Biomedical Engineering, Cornell University, Ithaca NY
[3]Siemens Corporate Technology, Princeton NJ
{yg326,pd83}@cornell.edu,{first.last}@siemens.com

## Contents



## 1. Objective functions

In this section, we provide complete mathematical expressions for each of the three terms in our loss function, following the notation defined in Section 3 of the main paper and the assumption that no training data is available in subdomain $\Sigma_{11}$.

### 1.1. Adversarial loss

For generator $G_1$ and discriminator $D_{10}$, for example, the adversarial loss is expressed as:

$$\begin{aligned}\mathcal{L}_{adv}(G_1, D_{10}, \Sigma_{00}, \Sigma_{10}) = \mathbb{E}_{\sigma_{10}\sim P_{10}}[\log D_{10}(\sigma_{10})] \\ +\mathbb{E}_{\sigma_{00}\sim P_{00}}[\log(1 - D_{10}(G_1(\sigma_{00})))]\end{aligned} \tag{1}$$

where the generator $G_1$ and discriminator $D_{10}$ are learned to optimize a minimax objective such that

$$G_1^* = \arg\min_{G_1}\max_{D_{10}} \mathcal{L}_{adv}(G_1, D_{10}, \Sigma_{00}, \Sigma_{10}) \tag{2}$$

For generator $G_2$ and discriminator $D_{01}$, the adversarial loss is expressed as:

$$\begin{aligned}\mathcal{L}_{adv}(G_2, D_{01}, \Sigma_{00}, \Sigma_{01}) = \mathbb{E}_{\sigma_{01}\sim P_{01}}[\log D_{01}(\sigma_{01})] \\ +\mathbb{E}_{\sigma_{00}\sim P_{00}}[\log(1 - D_{01}(G_2(\sigma_{00})))]\end{aligned} \tag{3}$$

For generator $F_1$ and discriminator $D_{00}$, the adversarial loss is expressed as:

$$\begin{aligned}\mathcal{L}_{adv}(F_1, D_{00}, \Sigma_{10}, \Sigma_{00}) = \mathbb{E}_{\sigma_{00}\sim P_{00}}[\log D_{00}(\sigma_{00})] \\ +\mathbb{E}_{\sigma_{10}\sim P_{10}}[\log(1 - D_{00}(F_1(\sigma_{10})))]\end{aligned} \tag{4}$$

For generator $F_2$ and discriminator $D_{00}$, the adversarial loss is expressed as:

$$\begin{aligned}\mathcal{L}_{adv}(F_2, D_{00}, \Sigma_{01}, \Sigma_{00}) = \mathbb{E}_{\sigma_{00}\sim P_{00}}[\log D_{00}(\sigma_{00})] \\ +\mathbb{E}_{\sigma_{01}\sim P_{01}}[\log(1 - D_{00}(F_2(\sigma_{01})))]\end{aligned} \tag{5}$$

The overall adversarial loss $\mathcal{L}_{ADV}$ is the sum of these four terms.

$$\begin{aligned}\mathcal{L}_{ADV} =& \mathcal{L}_{adv}(G_1, D_{10}, \Sigma_{00}, \Sigma_{10}) \\ &+ \mathcal{L}_{adv}(G_2, D_{01}, \Sigma_{00}, \Sigma_{01}) \\ &+ \mathcal{L}_{adv}(F_1, D_{00}, \Sigma_{10}, \Sigma_{00}) \\ &+ \mathcal{L}_{adv}(F_2, D_{00}, \Sigma_{01}, \Sigma_{00})\end{aligned} \tag{6}$$

### 1.2. Extended cycle-consistency loss

Following our discussion in Section 3.2 of the main paper, for any data sample $\sigma_{00}$ in subdomain $\Sigma_{00}$, a distance-4 cycle consistency constraint is defined in the clockwise direction $(F_2 \circ F_1 \circ G_2 \circ G_1)(\sigma_{00}) \approx \sigma_{00}$ and in the counterclockwise direction $(F_1 \circ F_2 \circ G_1 \circ G_2)(\sigma_{00}) \approx \sigma_{00}$. Such constraints are implemented by the penalty function:

$$\begin{aligned}&\mathcal{L}_{cyc4}(G, F, \Sigma_{00}) \\ &\quad= \mathbb{E}_{\sigma_{00}\sim P_{00}}[\|(F_2 \circ F_1 \circ G_2 \circ G_1)(\sigma_{00}) - \sigma_{00}\|_1] \\ &\quad+ \mathbb{E}_{\sigma_{00}\sim P_{00}}[\|(F_1 \circ F_2 \circ G_1 \circ G_2)(\sigma_{00}) - \sigma_{00}\|_1].\end{aligned} \tag{7}$$

Similarly, $\mathcal{L}_{cyc4}(G, F, \Sigma_{01})$ is defined as:

$$\begin{aligned}
&\mathcal{L}_{cyc4}(G, F, \Sigma_{01}) \\
&\quad = \mathbb{E}_{\sigma_{01}\sim P_{01}}[\|(F_1 \circ G_2 \circ G_1 \circ F_2)(\sigma_{01}) - \sigma_{01}\|_1] \\
&\quad + \mathbb{E}_{\sigma_{01}\sim P_{01}}[\|(G_2 \circ F_1 \circ F_2 \circ G_1)(\sigma_{01}) - \sigma_{01}\|_1].
\end{aligned} \tag{8}$$

Finally, $\mathcal{L}_{cyc4}(G, F, \Sigma_{10})$ is defined as:

$$\begin{aligned}
&\mathcal{L}_{cyc4}(G, F, \Sigma_{10}) \\
&\quad = \mathbb{E}_{\sigma_{10}\sim P_{10}}[\|(G_1 \circ F_2 \circ F_1 \circ G_2)(\sigma_{10}) - \sigma_{10}\|_1] \\
&\quad + \mathbb{E}_{\sigma_{10}\sim P_{10}}[\|(F_2 \circ G_1 \circ G_2 \circ F_1)(\sigma_{10}) - \sigma_{10}\|_1].
\end{aligned} \tag{9}$$

Let $\mathcal{L}_{CYC4}$ denotes the sum of these three terms:

$$\begin{aligned}
\mathcal{L}_{CYC4} =&\mathcal{L}_{cyc4}(G, F, \Sigma_{00}) + \mathcal{L}_{cyc4}(G, F, \Sigma_{01}) \\
&+ \mathcal{L}_{cyc4}(G, F, \Sigma_{10})
\end{aligned} \tag{10}$$

The overall cycle consistency loss $\mathcal{L}_{CYC}$ is defined as:

$$\mathcal{L}_{CYC} = \mathcal{L}_{CYC2} + \mathcal{L}_{CYC4} \tag{11}$$

where $\mathcal{L}_{CYC2}$ is the sum of all pairwise distance-2 cycle consistency losses as described in Section 3.2 of the main paper.

### 1.3. Commutative loss

Following our discussion in Section 3.3 of the main paper, for any data sample $\sigma_{00}$ in subdomain $\Sigma_{00}$, we introduce a constraint $(G_2 \circ G_1)(\sigma_{00}) \approx (G_1 \circ G_2)(\sigma_{00})$ implemented by the penalty function:

$$\begin{aligned}
&\mathcal{L}_{comm}(G_1, G_2, \Sigma_{00}) \\
&\quad =\mathbb{E}_{\sigma_{00}\sim P_{00}}[\|(G_2 \circ G_1)(\sigma_{00}) - (G_1 \circ G_2)(\sigma_{00})\|_1]
\end{aligned} \tag{12}$$

Similarly, $\mathcal{L}_{comm}(G_1, F_2, \Sigma_{01})$ is defined as:

$$\begin{aligned}
&\mathcal{L}_{comm}(G_1, F_2, \Sigma_{01}) \\
&\quad =\mathbb{E}_{\sigma_{01}\sim P_{01}}[\|(F_2 \circ G_1)(\sigma_{01}) - (G_1 \circ F_2)(\sigma_{01})\|_1]
\end{aligned} \tag{13}$$

and $\mathcal{L}_{comm}(F_1, G_2, \Sigma_{10})$ as:

$$\begin{aligned}
&\mathcal{L}_{comm}(F_1, G_2, \Sigma_{10}) \\
&\quad =\mathbb{E}_{\sigma_{10}\sim P_{10}}[\|(G_2 \circ F_1)(\sigma_{10}) - (F_1 \circ G_2)(\sigma_{10})\|_1]
\end{aligned} \tag{14}$$

The overall commutative loss $\mathcal{L}_{COMM}$ is the sum of the three terms.

$$\begin{aligned}
\mathcal{L}_{COMM} =&\mathcal{L}_{comm}(G_1, G_2, \Sigma_{00}) + \mathcal{L}_{comm}(G_1, F_2, \Sigma_{01}) \\
&+ \mathcal{L}_{comm}(F_1, G_2, \Sigma_{10})
\end{aligned} \tag{15}$$

The overall loss function is as defined in Equation 5 in Section 3.4 of the main paper.

## 2. Additional implementation details

In this section we provide additional implementation details to reproduce our results. For all three discriminators, we use the architecture adapted from Kim et al. [4] which contains 5 convolution layers with $4 \times 4$ filters where the first four are each followed by a leaky ReLU. Compared to the PatchGAN used in Zhu et al. [7], the discriminator network takes 64x64x3 input images and output a scalar for each image. For all the generators, we use the architecture adapted from Zhu et al [7], which contains 2 convolution layers with stride 2, 6 residual blocks and 2 fractionally-strided convolution layers with stride $\frac{1}{2}$. We use batch normalization for both the discriminator network and the generator network.

At the training stage, we apply the algorithm from Arjovsky et al. [1] for an alternative adversarial training. We use Adam optimizer [5] with an initial learning rate of 0.0002 at the first 150 epochs, followed by a linearly decaying learning rate for the next 150 epochs as the rate goes to zero. We set $\mu = \lambda = 10$ and we also include an identity loss component [7] with weight 10. In particular for the experiment involving two concepts with greater difference (i.e., "handbag vs. shoe" and "color vs. edge") , we include additional distance-3 adversarial components. For each training sample in $\Sigma_{00}$, the synthetic image generated by sequentially applying $(G_1, G_2, F_1)$ and $(G_2, G_1, F_2)$ are discriminated from real data in the corresponding output subdomains $\Sigma_{01}$ and $\Sigma_{10}$ respectively. To compare results of the proposed method to baseline CycleGANs [7], we consider two CycleGAN models each between two adjacent subdomains in our proposed framework, which are trained separately using the same network architecture of discriminators and generators as described above.

## 3. Additional results

In Figures 1 through 3, we provide additional image synthesis outputs of the proposed ConceptGAN. In Figure 4, we provide additional illustrations of improvement in face verification with augmented data generated by ConceptGAN. In Figures 5 and 6, we show additional qualitative results demonstrating the transferability of concepts learned using ConceptGAN to independent test datasets LFW [6] and MS-Celeb-1M [2].

In Table 3 in the main paper where we show face verification results for 3 concepts, we adopted a specific set of paths in our graph using which the augmented data, 8 images in total, one corresponding to each of the 8 vertices in our cyclic graph, was generated. To show that this is not a critical constraint, we repeat this experiment with multiple randomly chosen set of paths to generate the augmented data, again, 8 images in total as above. The average results over these multiple trials are shown

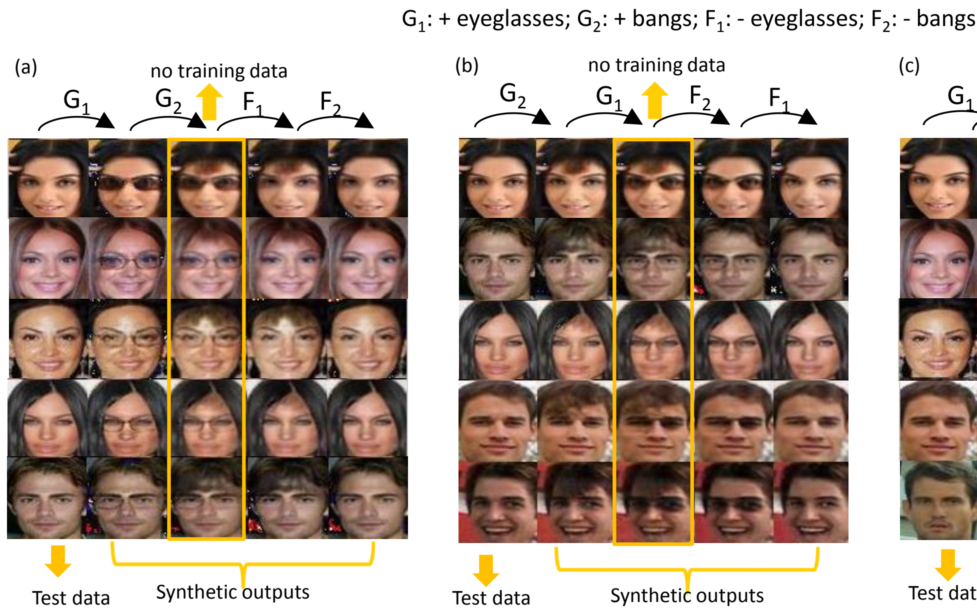


Figure 1: Image translation and synthesis conditional on concepts “bangs” and ”eyeglasses”. Each panel in column (a) demonstrates the clockwise cycle consistency where $\sigma_{00}$, $G_1(\sigma_{00})$, $(G_2 \circ G_1)(\sigma_{00})$, $(F_1 \circ G_2 \circ G_1)(\sigma_{00})$, $(F_2 \circ F_1 \circ G_2 \circ G_1)(\sigma_{00})$ are shown in sequence, from left to right. Each panel in column (b) demonstrates the counter-clockwise cycle consistency where $\sigma_{00}$, $G_2(\sigma_{00})$, $(G_1 \circ G_2)(\sigma_{00})$, $(F_2 \circ G_1 \circ G_2)(\sigma_{00})$, $(F_1 \circ F_2 \circ G_1 \circ G_2)(\sigma_{00})$ are shown in sequence, from left to right. Each panel in column (c) demonstrates the commutative property of the concept composition where $\sigma_{00}$, $G_1(\sigma_{00})$, $G_2(\sigma_{00})$, $(G_2 \circ G_1)(\sigma_{00})$, $(G_1 \circ G_2)(\sigma_{00})$ are shown in sequence, from left to right. Synthesis results obtained in the subdomains where no training data is available are highlighted in yellow boxes.

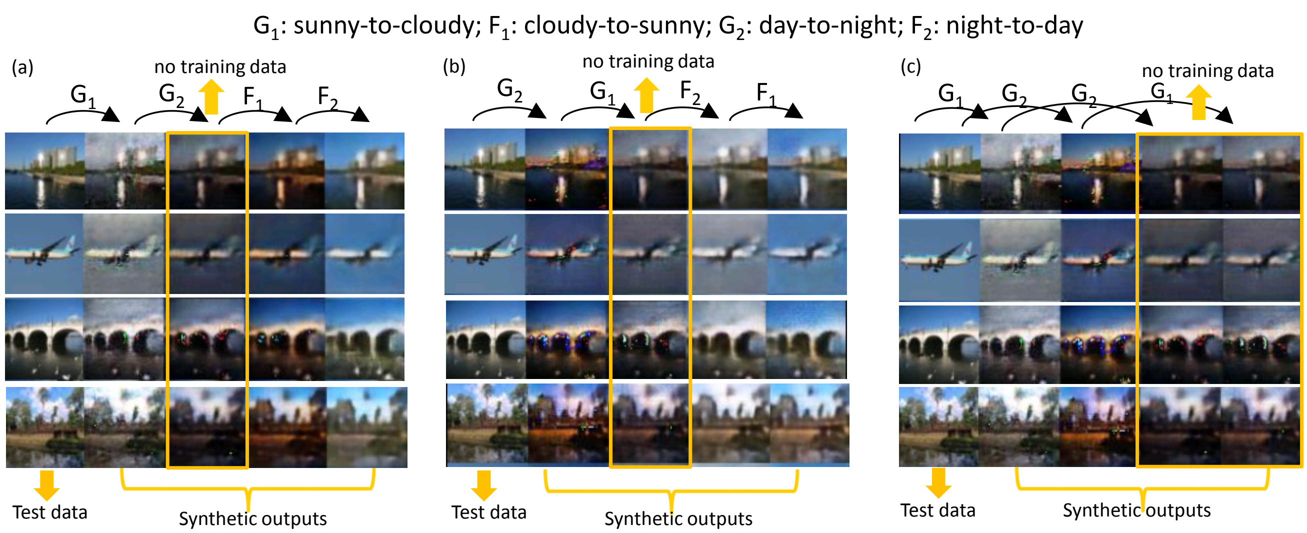


Figure 2: Image translation and synthesis conditional on scene attributes ”day/night” and ”sunny/cloudy”. Each panel in column (a) demonstrates the clockwise cycle consistency where $\sigma_{00}$, $G_1(\sigma_{00})$, $(G_2 \circ G_1)(\sigma_{00})$, $(F_1 \circ G_2 \circ G_1)(\sigma_{00})$, $(F_2 \circ F_1 \circ G_2 \circ G_1)(\sigma_{00})$ are shown in sequence, from left to right. Each panel in column (b) demonstrates the counter-clockwise cycle consistency where $\sigma_{00}$, $G_2(\sigma_{00})$, $(G_1 \circ G_2)(\sigma_{00})$, $(F_2 \circ G_1 \circ G_2)(\sigma_{00})$, $(F_1 \circ F_2 \circ G_1 \circ G_2)(\sigma_{00})$ are shown in sequence, from left to right. Each panel in column (c) demonstrates the commutative property of the concept composition where $\sigma_{00}$, $G_1(\sigma_{00})$, $G_2(\sigma_{00})$, $(G_2 \circ G_1)(\sigma_{00})$, $(G_1 \circ G_2)(\sigma_{00})$ are shown in sequence, from left to right. Synthesis results obtained in the subdomains where no training data is available are highlighted in yellow boxes.

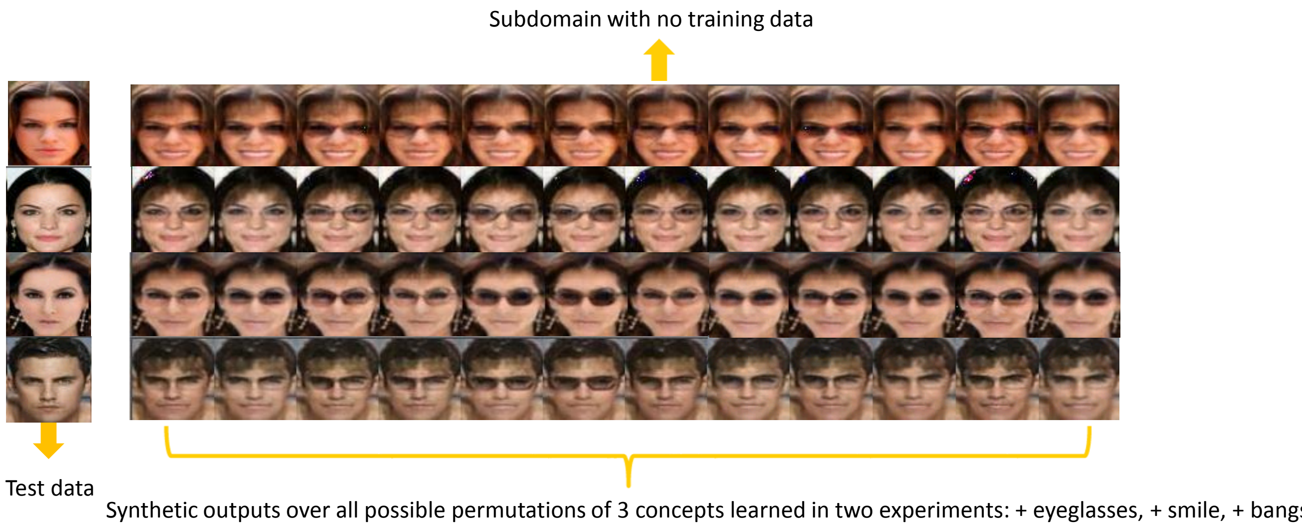


Figure 3: Image synthesis in a zero-shot subdomain by composing three concepts (smile, eyeglasses, bangs) learned in two separate experiments. Concept mappings with respect to "eyeglasses" is learned in each of two experiments therefore $2 \times (3!) = 12$ different compositions of mappings available to translate images labeled as (no smile, no eyeglasses, no bangs) to the target subdomain.

| Attributes | Smiling, Bangs, & Eyeglasses | | |
|---|---|---|---|
| Ranking Method | $l_2$ | RNP | SRID |
| Augmentation | No | Yes | Yes |
| CaffeFace | 10.4 | 10.7 | 13.3 |
| VGGFace | 46.7 | 54.7 | 59.3 |

Table 1: Rank-1 face verification results (in %) for three concepts: no augmentation (where we use $l_2$ distance to rank) vs. augmentation with ConceptGAN (where we use the multi-shot ranking algorithms, RNP and SRID to rank).

in Table 1, where we see: (a) improved face verification performance with augmented data, and (b) no substantial difference when compared to those of Table 3 in the main paper. In Section 5.3 in the main paper, we reported ablation results for the attribute classification experiment corresponding to the "Bangs" and "Eyeglasses" attributes. In Table 2, we report complete results corresponding to this experiment. Finally, in Figure 7, we show additional qualitative results for synthesizing $128 \times 128$ images for the "Bangs" and "Eyeglasses" attributes. We note that in this experiment, instead of $64 \times 64 \times 3$ images, our architecture takes $128 \times 128 \times 3$ images as input. The only difference to the architecture described in Section 2 is the filter size in the last layer of the discriminator, which is changed to obtain a scalar value as output.

# 4. Discussion

In this section, we provide further insight and discussion on ConceptGAN. In particular, while we do not make any assumptions on the type of concepts to be learned, except for encouraging a commutative composition, the symmetric design of our model suggests that training may be challenging in cases where the two concepts are greatly imbalanced. As an example, consider the experiment involving the concepts "handbag vs. shoe" and "color vs. edge", which are of markedly different types. As shown in Figure 8 panel (a), it is harder to achieve a semantically meaningful composition in subdomain$\Sigma_{11}$ by composing pairs of concepts in one particular order than the other, i.e., $G_2 \circ G_1$ gives better performance when compared to $G_1 \circ G_2$. In such cases, the results in subdomain $\Sigma_{11}$ may reflect translation with respect to only one concept instead of composition of the two concepts. To achieve plausible synthesis as reported in Figures 1 and 3 of the main paper and shown in panel (b) of Figure 8, we address the issue by further constraining the system with additional distance-3 adversarial constraints.

# 5. Generalizing ConceptGAN

This section provides an extended discussion on a principled generalization of our framework to $n >= 1$ concepts under the assumptions that concepts have distinct states and that they are not mutually inhibiting. We do not necessarily need to capture the *universe* of concepts in the data, i.e. all concepts in existence, as long as the domain mapping for each known concept can be learned from data.

## 5.1. Assumption: Concepts have distinct states

More precisely we assume that each sample $x$ has a likelihood $P(x|\Theta)$, where $\Theta$ is the universe of latent and

| Classifier | Val | CycleGAN | Full model | Without $\mathcal{L}_{COMM}$ | Without $\mathcal{L}_{CYC4}$ |
|---|---|---|---|---|---|
| C1: “with” vs. “no” eyeglasses | 98 | 93 | 98 | 88 | 31 |
| C2: “with” vs. “no” bangs | 93 | 61 | 67 | 68 | 62 |
| Both C1 and C2 | N/A | 56 | 66 | 60 | 18 |

Table 2: Ablation results for classifying face images synthesized via ConceptGAN (ours) vs. CycleGAN [7]. Classifier 1 is trained and validated with images with and without eyeglasses. Classifier 2 is trained and validated with images with and without bangs. The test set consists of “with eyeglasses, with bangs” images only and the different orders of composing learned mappings contribute equally. Joint classification accuracy is reported as the percentage of the images correctly classified in two tests at the same time.

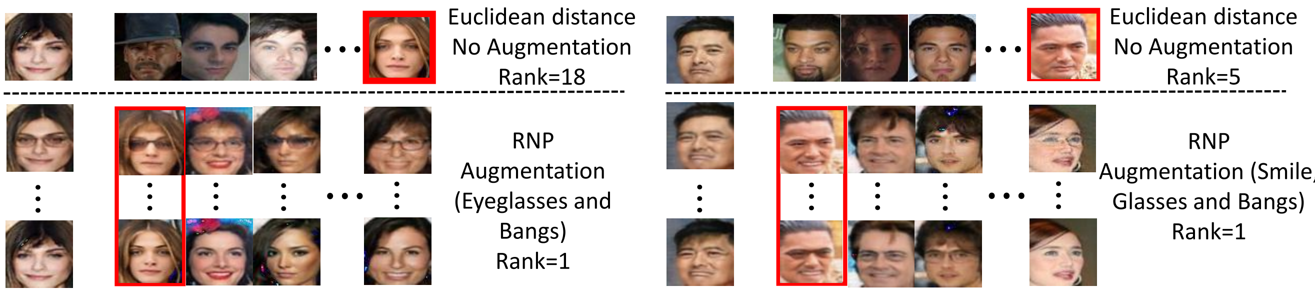


Figure 4: Qualitative illustrations of improvement in face verification performance- we show the improvement in the retrieved rank with augmented data using the (“eyeglasses”, “bangs”) and (“eyeglasses”,“bangs”,“smiling”) attribute sets.

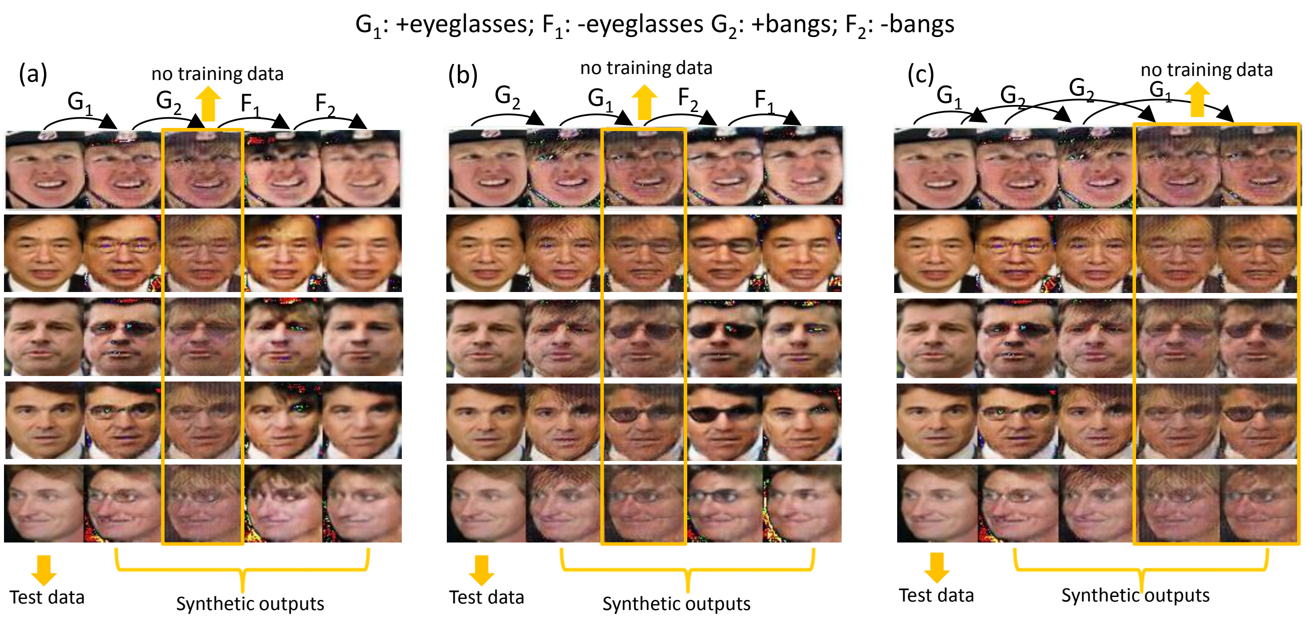


Figure 5: Transfer of learned concepts to LFW: Image translation and conditional synthesis on face attributes “eyeglasses” and “bangs” via direct application of models trained by CelebA data [6] on independent test dataset LFW [3].

observable variables that influence $x$, for instance illumination, geometry and object class. Then each concept $c_i \in \mathbb{C}$ of interest has to be attributable to a random variable $c_i \subseteq \Theta$ and $c_i$ has to be discrete. Naturally the case $c = \Theta$ is not particularly interesting as there is only one concept in the universe that generates the data. It is important to note that in the general case of $\{c_1, \dots, c_n\} \subset \Theta$ the non-concept variables $\Theta \setminus C$ may be continuous random variables and the distribution $P(x|\Theta) = P(x|\{\mathbb{C}, \Theta \setminus \mathbb{C}\})$ itself may be continuous. Without loss of generality the following sections assume that the number of states for each concept is two, i.e. binary concepts. Settings with more states may be mapped by assigning binary sub-concepts corresponding to a binary representation of the states.

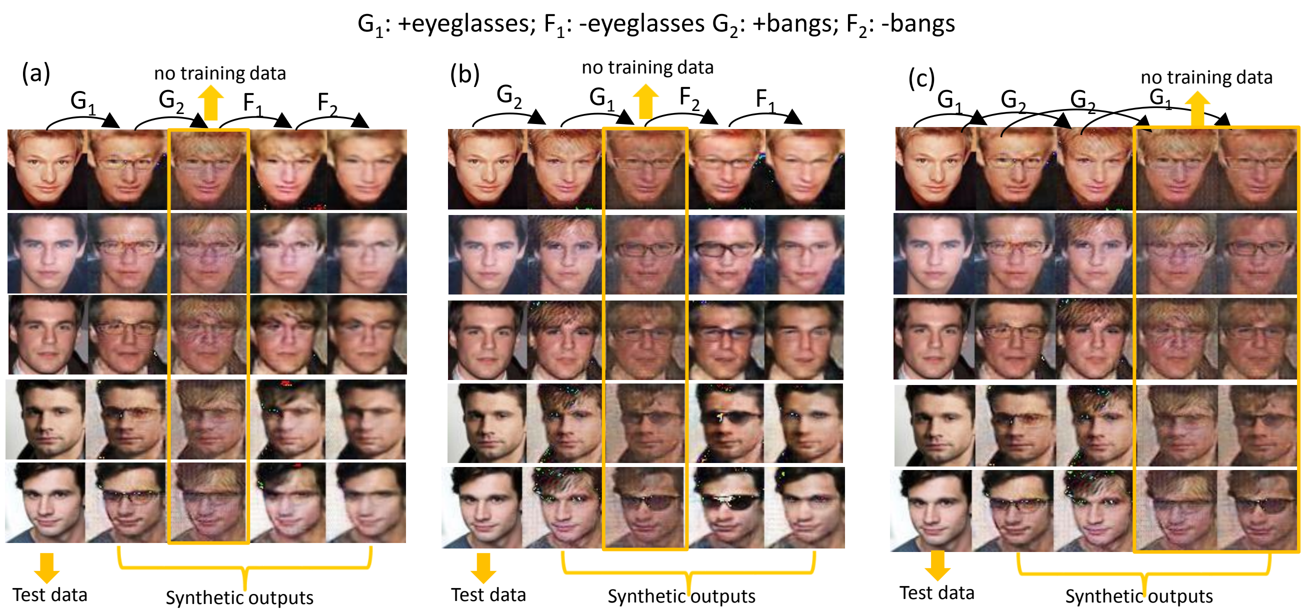


Figure 6: Transfer of learned concepts to MS-Celeb-1M: Image translation and conditional synthesis on face attributes “eyeglasses” and “bangs” via direct application of models trained by CelebA data [6] on independent test dataset MS-Celeb-1M [2].

Figure 7: Additional $128 \times 128$ synthesis results- in each panel, column 1 shows the base image in the $\Sigma_{00}$ domain and column 2 shows the synthesized image in the $\Sigma_{11}$ domain.

### 5.2. Assumption: Concepts are mutually compatible

The second simplifying assumption is that the activation of one concept does not preclude activation of any other concept, i.e. all combinations of concepts are physically meaningful. This is motivated from the perspective that it enables us to formulate a consistent optimization frame work based on constraint cycles without special cases. It is conceivable to impose less strict assumptions, and our graph-based solution exposed here may yield a suitable starting point to address the case where not all combinations of concepts are physically meaningful.

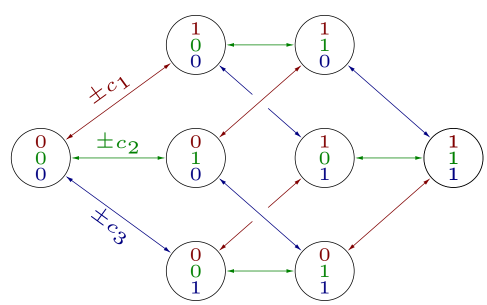


Figure 9: Generalizing ConceptGAN to $n$ concepts, illustrated with $n = 3$.

### 5.3. Generalization

The main insight in our generalization is that mutual constraints over two concepts are sufficient to provide prin-

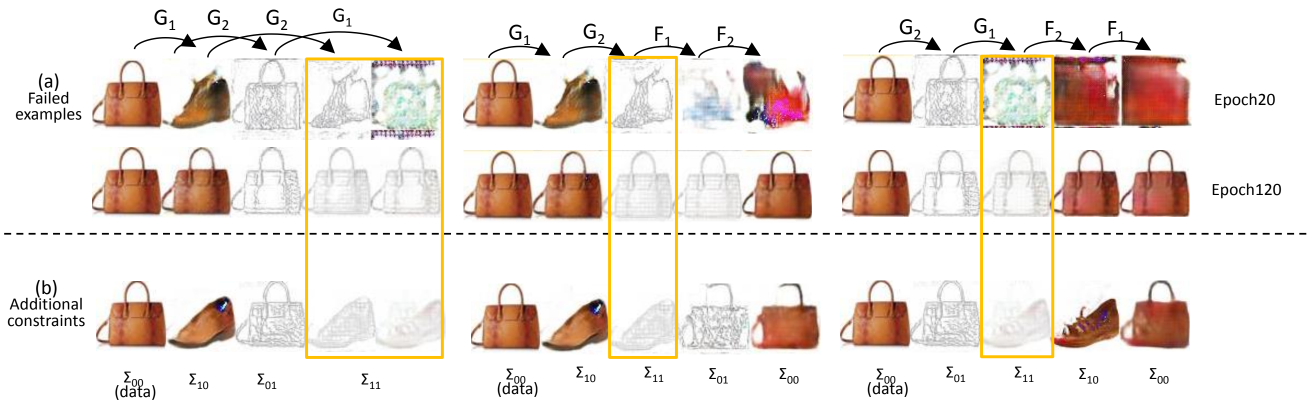


Figure 8: Examples illustrating the limitation of the symmetric setup of ConceptGAN in case of imbalanced concepts. Four subdomains are "color/handbag" ($\Sigma_{00}$), "color/shoe"($\Sigma_{10}$), "edge/handbag"($\Sigma_{01}$) and "edge/shoe"($\Sigma_{11}$) respectively. Yellow boxes highlights synthetic results in subdomain $\Sigma_{11}$ where no training data is available. Panel (a) shows results without additional distance-3 adversarial constraints. The first row provides results at an early stage of the training, where the differences between highlighted outputs suggest that in this example, concept "color vs. edge" ($G_2/F_2$) is easier to get transferred (i.e., performs well on different input subdomains) compared to the concept "handbag vs. shoe"($G_1/F_1$). The second row provides examples of failed composition as training proceeds, where only the concept that is easier to learn is reflected. Panel (b) shows results with the above-mentioned additional constraints.

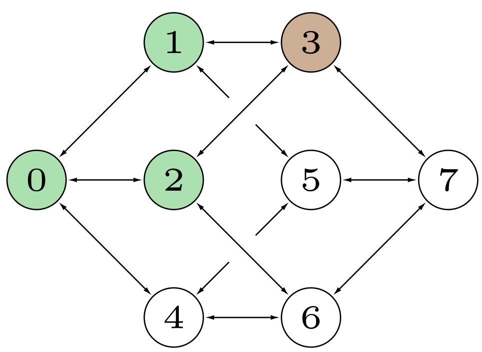


Figure 10: Green color indicates observed node, i.e., we have data available from the underlying distribution corresponding to the node.

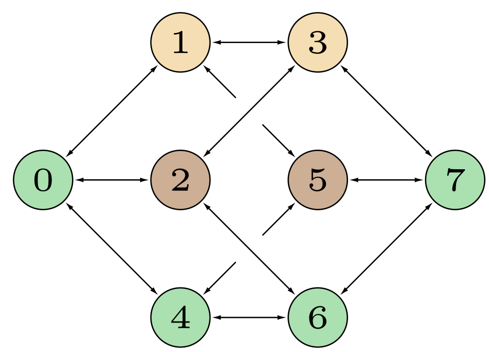


Figure 12: Concepts $c_1, c_2, c_3$ defined by observing nodes 0,4,6,7, allowing primary inference of nodes 2,5, and secondary inference of nodes 1,3.

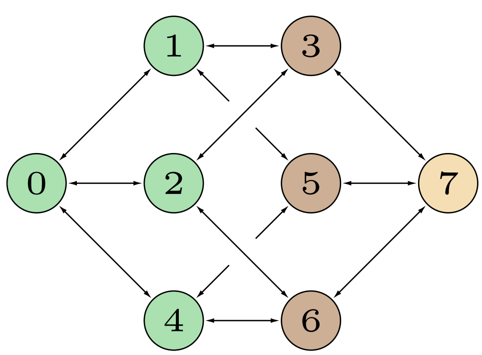


Figure 11: Concepts $c_1, c_2, c_3$ defined by observing nodes 0,1,2,4, allowing primary inference of nodes 3,5,6, and secondary inference of node 7.

cipled approximations to generating samplers from the distribution of all concepts, even when not all concept combinations can be observed in the data. Figure 9 illustrates the case $n = 3$ where the universe of concepts are the binary random variables $\mathbb{C} = \{c_1, c_2, c_3\}$ as a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where the edges $\mathcal{E}$ are changes in one binary concept and the nodes $\mathcal{V}$ are the $2^n$ states of the $n$ latent variables. All other influence factors $\Theta \setminus \mathbb{C}$ are not shown. Each node corresponds to one possible state over the concepts. This is where our assumption about mutual compatibility becomes relevant, as it implies that the graph is connected as shown, with each node having exactly $n$ incoming and $n$ outgoing edges corresponding to activating or deactivating a concept. Each concept transfer is represented multiple times in the graph, for instance $\pm c_3$ can be observed four times. Under our assumptions, in general each concept transfer will occur $2^{n-1}$ times in the

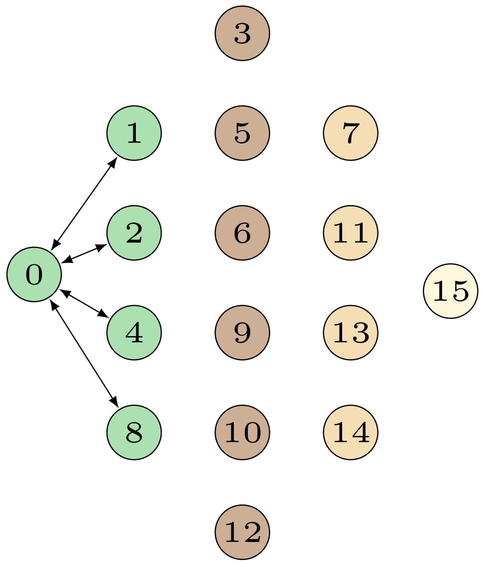


Figure 13: Sketching the modes of inference for $n = 4$ concepts.

graph. There are two options in how to model the concept transfer in this case: state-dependent or state-independent. In the first case we assume that the concept transfer $\pm c_i$ is a function of its originating state, in the first case we assume that the concept transfer is independent of the originating state. The latter leads to more constraints per concept transfer and a smaller solution space with the first case vice versa. For instance, in the example of figure 9, the concept transfer $+c_3$ may be considered state-dependent, where we have the four separate, but related concepts $+c_3|[0, 0, 0]$, $+c_3|[0, 1, 0]$, $+c_3|[1, 0, 0]$ and $+c_3|[1, 1, 0]$ to learn. If we are concerned about the lack of constraints available, we may choose to rather treat $+c_3$ state independent, leaving only $+c_3|[\cdot, \cdot, 0]$ to be learned, where we can aggregate all constraints over the combinations of $c_1$ and $c_2$. Our method presented in this paper and the further discussion assumes that the concept transfer is state-independent and aggregates the constraints over all possible combinations.

Given sets of data, each node may be observed or not as illustrated in figure 10, where the color green indicates an observed node. "Observed" precisely means that we have data available that is drawn from the underlying distribution corresponding to a node. In our example three nodes and thus three concept combinations are observable: $v_0 = [c_1 = 0, c_2 = 0, c_3 = 0]$, $v_1 = [c_1 = 1, c_2 = 0, c_3 = 0]$ and $v_2 = [c_1 = 0, c_2 = 1, c_3 = 0]$. We can hope to infer two concept transfers from this, with the corresponding edges between the observable nodes denoted by $\pm c_1$ and $\pm c_2$ in figure 9. Applying our previous discussion on cycle consistent concept learning then allows us to infer a generator for data from node $v_3$, indicated in brown, as there are two concepts involved, $\pm c_1$ and $\pm c_2$. Indeed, the sub-graph composed of the nodes $v_{\{0,1,2,3\}}$ and their corresponding edges is exactly our proposed solution with two concepts. Let's now add another data set drawn from $v_4$, as illustrated in figure 11, meaning that we observe the additional concept combination corresponding to $v_4$. This allows us to observe $\pm c_3$ between $v_0$ and $v_4$. The resulting graph shows that we can now infer also $v_5$ and $v_6$ by adding the two concept constraints corresponding the the cycles $(0, 2, 6, 4)$ and $(0, 1, 5, 4)$. Together with their reflections and shifts we need to add a total of $4 * 2$ additional constraints per added inferred node for complete cycle consistency. Note that so far we have only applied our method without changing the nature of optimizing over two concept constraints. We now take the next step in generalization by considering node $v_7$, which in this particular example corresponds to all three concepts being activated. Assuming that we indeed can infer nodes $v_{3,5,6}$, we can consider constraints that treat them as "observed", such as over the cycles $(3, 7, 5, 1)$, $(5, 7, 6, 4)$, and $(6, 7, 3, 2)$. This allows us in principle to estimate samples for $v_7$, and yields a generic template for approximate, iterative algorithms based on a structure graph representation of concepts and available data. To illustrate the generic nature of this approach, figure 12 illustrates a situation where we have data from the nodes $\{0, 4, 6, 7\}$. We then can primarily infer nodes $\{2, 5\}$ and secondarily infer nodes $\{1, 3\}$. Without loss of generality for arbitrary $n > 3$ we consider the special case where all observed concepts have one common ancestor as in figure 11 (compared to figure 12, where this is not the case). Then a baseline algorithm would first infer $m_2$ primary, then $m_3$ secondary nodes and so on, with $m_k = \binom{n}{k}$, and $\sum_k m_k = 2^n$. Figure 13 sketches this for $n = 4$. It is important to note here that one cannot escape the combinatorial complexity of generating samplers over all concept combinations. Nevertheless, our proposed generalization paves the way for iterative algorithms that yield approximate solutions in polynomial time. For instance, a simple optimization scheme may fully fix the inferred nodes at stage $k$ before starting to infer stage $k + 1$, where a more powerful scheme may instead weigh the uncertainty of each unobserved nodes' estimation in the joint inference process. The uncertainty may be proportional to the likelihood of the node resembling its true underlying distribution. Other factors in optimizing may include properties of the available data, where nodes with less data are less trusted. In the general case of $n >= 1$, one would reasonably expect that the uncertainty over the estimation of a sample generator increases with its nodes' graph distance to the available observed nodes. For instance, it may be proportional to the average distance to all observed nodes, or the minimum distance to the next observed node.

## References


[1] M. Arjovsky, S. Chintala, and L. Bottou. Wasserstein generative adversarial networks. In *ICML*, pages 214–223, 2017.

[2] Y. Guo, L. Zhang, Y. Hu, X. He, and J. Gao. Ms-celeb-1m: A dataset and benchmark for large scale face recognition. In *ECCV*, 2016.

[3] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Technical Report 07-49, University of Massachusetts, Amherst, October 2007.

[4] T. Kim, M. Cha, H. Kim, J. K. Lee, and J. Kim. Learning to discover cross-domain relations with generative adversarial networks. In *ICML*, pages 1857–1865, 2017.

[5] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.

[6] Z. Liu, P. Luo, X. Wang, and X. Tang. Deep learning face attributes in the wild. In *ICCV*, 2015.

[7] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *ICCV*, 2017.