Supplementary Material: Learning Compositional Visual Concepts with Mutual Consistency

Yunye Gong\(^1\), Srikrishna Karanam\(^3\), Ziyin Wu\(^3\), Kuan-Chuan Peng\(^3\), Jan Ernst\(^4\), 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 .............................. 1
   1.1. Adversarial loss .......................... 1
   1.2. Extended cycle-consistency loss .......... 1
   1.3. Commutative loss .......................... 2

2. Additional implementation details .............. 2

3. Additional results .............................. 2

4. Discussion ...................................... 4

5. Generalizing ConceptGAN ........................ 4
   5.1. Assumption: Concepts have distinct states . 4
   5.2. Assumption: Concepts are mutually compatible 6
   5.3. Generalization ............................. 6

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:

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

(1)

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

\[
G_1^* = \arg\min_{G_1} \max_{D_{10}} \mathcal{L}_{\text{adv}}(G_1, D_{10}, \Sigma_{00}, \Sigma_{10})
\]

(2)

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

\[
\mathcal{L}_{\text{adv}}(G_2, D_{01}, \Sigma_{00}, \Sigma_{01}) = \mathbb{E}_{\sigma_{01} \sim P_{\sigma_{01}}} \log D_{01}(\sigma_{01}) + \mathbb{E}_{\sigma_{00} \sim P_{\sigma_{00}}} \log(1 - D_{01}(G_2(\sigma_{00})))
\]

(3)

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

\[
\mathcal{L}_{\text{adv}}(F_1, D_{00}, \Sigma_{10}, \Sigma_{00}) = \mathbb{E}_{\sigma_{00} \sim P_{\sigma_{00}}} \log D_{00}(\sigma_{00}) + \mathbb{E}_{\sigma_{01} \sim P_{\sigma_{01}}} \log(1 - D_{00}(F_1(\sigma_{10})))
\]

(4)

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

\[
\mathcal{L}_{\text{adv}}(F_2, D_{00}, \Sigma_{01}, \Sigma_{00}) = \mathbb{E}_{\sigma_{00} \sim P_{\sigma_{00}}} \log D_{00}(\sigma_{00}) + \mathbb{E}_{\sigma_{01} \sim P_{\sigma_{01}}} \log(1 - D_{00}(F_2(\sigma_{01})))
\]

(5)

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

\[
\mathcal{L}_{\text{ADV}} = \mathcal{L}_{\text{adv}}(G_1, D_{10}, \Sigma_{00}, \Sigma_{10}) + \mathcal{L}_{\text{adv}}(G_2, D_{01}, \Sigma_{00}, \Sigma_{01}) + \mathcal{L}_{\text{adv}}(F_1, D_{00}, \Sigma_{10}, \Sigma_{00}) + \mathcal{L}_{\text{adv}}(F_2, D_{00}, \Sigma_{01}, \Sigma_{00})
\]

(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:

\[
\mathcal{L}_{\text{cyc4}}(G, F, \Sigma_{00}) = \mathbb{E}_{\sigma_{00} \sim P_{\sigma_{00}}} \|[F_2 \circ F_1 \circ G_2 \circ G_1](\sigma_{00}) - \sigma_{00}\|_1 + \mathbb{E}_{\sigma_{00} \sim P_{\sigma_{00}}} \|[F_1 \circ F_2 \circ G_1 \circ G_2](\sigma_{00}) - \sigma_{00}\|_1.
\]

(7)
Similarly, \( L_{\text{comm}}(G, F, \Sigma_{01}) \) is defined as:
\[
L_{\text{comm}}(G, F, \Sigma_{01}) = \mathbb{E}_{\sigma_{01} \sim P_{01}}[\| (F \circ G_1 \circ F_2)(\sigma_{01}) - (G_1 \circ G_2)(\sigma_{01}) \|_1] \\
+ \mathbb{E}_{\sigma_{01} \sim P_{01}}[\| (G_2 \circ F_1 \circ G_1)(\sigma_{01}) - (G_2 \circ G_1)(\sigma_{01}) \|_1].
\] (8)

Finally, \( L_{\text{comm}}(G, F, \Sigma_{10}) \) is defined as:
\[
L_{\text{comm}}(G, F, \Sigma_{10}) = \mathbb{E}_{\sigma_{10} \sim P_{10}}[\| (G_2 \circ F_1 \circ G_2)(\sigma_{10}) - (G_2 \circ G_1)(\sigma_{10}) \|_1] \\
+ \mathbb{E}_{\sigma_{10} \sim P_{10}}[\| (F_2 \circ G_1 \circ F_1)(\sigma_{10}) - (G_2 \circ G_1)(\sigma_{10}) \|_1].
\] (9)

Let \( L_{\text{CYC}} \) denotes the sum of these three terms:
\[
L_{\text{CYC}} = L_{\text{CYC}1} + L_{\text{CYC}2} + L_{\text{CYC}4}
\] (10)

The overall cycle consistency loss \( L_{\text{CYC}} \) is defined as:
\[
L_{\text{CYC}} = L_{\text{CYC}2} + L_{\text{CYC}4}
\] (11)

where \( L_{\text{CYC}2} \) 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 \( \sigma_0 \) in subdomain \( \Sigma_0 \), we introduce a constraint \((G_2 \circ G_1)(\sigma_0) \approx (G_1 \circ G_2)(\sigma_0)\) implemented by the penalty function:
\[
L_{\text{comm}}(G_1, G_2, \Sigma_0) = \mathbb{E}_{\sigma_0 \sim P_0}[\| (G_2 \circ G_1)(\sigma_0) - (G_1 \circ G_2)(\sigma_0) \|_1].
\] (12)

Similarly, \( L_{\text{comm}}(G_1, F, \Sigma_{01}) \) is defined as:
\[
L_{\text{comm}}(G_1, F_2, \Sigma_{01}) = \mathbb{E}_{\sigma_{01} \sim P_{01}}[\| (F_2 \circ G_1)(\sigma_{01}) - (G_1 \circ F_2)(\sigma_{01}) \|_1].
\] (13)

and \( L_{\text{comm}}(F_1, G_2, \Sigma_{10}) \) as:
\[
L_{\text{comm}}(F_1, G_2, \Sigma_{10}) = \mathbb{E}_{\sigma_{10} \sim P_{10}}[\| (G_2 \circ F_1)(\sigma_{10}) - (F_1 \circ G_2)(\sigma_{10}) \|_1].
\] (14)

The overall commutative loss \( L_{\text{COMM}} \) is the sum of the three terms.
\[
L_{\text{COMM}} = L_{\text{comm}}(G_1, G_2, \Sigma_{00}) + L_{\text{comm}}(G_1, F_2, \Sigma_{01}) + L_{\text{comm}}(F_1, G_2, \Sigma_{10})
\] (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 \( \Sigma_0 \), 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 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 $\sigma_{00}, G_2(\sigma_{00}), G_1(\sigma_{00}), (G_2 \circ G_1)(\sigma_{00}), (G_1 \circ G_2)(\sigma_{00})$ 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 $\sigma_{00}, G_2(\sigma_{00}), G_1(\sigma_{00}), (G_2 \circ G_1)(\sigma_{00}), (G_1 \circ G_2)(\sigma_{00})$ 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.

<table>
<thead>
<tr>
<th>Attributes</th>
<th>Smiling, Bangs, &amp; Eyeglasses</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ranking Method</td>
<td>$l_2$</td>
</tr>
<tr>
<td>Augmentation</td>
<td>No</td>
</tr>
<tr>
<td>CaffeFace</td>
<td>10.4</td>
</tr>
<tr>
<td>VGGFace</td>
<td>46.7</td>
</tr>
</tbody>
</table>

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 $n \geq 1$ 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
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.

<table>
<thead>
<tr>
<th>Classifier</th>
<th>Val CycleGAN</th>
<th>Full model</th>
<th>Without $L_{COM}$</th>
<th>Without $L_{CYC}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>C1: “with” vs. “no” eyeglasses</td>
<td>98</td>
<td>93</td>
<td>98</td>
<td>88</td>
</tr>
<tr>
<td>C2: “with” vs. “no” bangs</td>
<td>93</td>
<td>61</td>
<td>67</td>
<td>68</td>
</tr>
<tr>
<td>Both C1 and C2</td>
<td>N/A</td>
<td>56</td>
<td>66</td>
<td>60</td>
</tr>
</tbody>
</table>

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 $\{c_1, \ldots, c_n\} \subset \Theta$ the non-concept variables $\Theta \setminus C$ may be continuous random variables and the distribution $P(x|\Theta) = P(x|\{C, \Theta \setminus C\})$ 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 × 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 framework 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” (Σ_{00}), “color/shoe”(Σ_{10}), “edge/handbag”(Σ_{01}) and “edge/shoe”(Σ_{11}) respectively. Yellow boxes highlights synthetic results in subdomain Σ_{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 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.

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.
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 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 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 \((6, 7, 3, 2) \). 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.

Figure 13: Sketching the modes of inference for \( n = 4 \) concepts.
References


