HIERARCHICAL ADVERSARIALY LEARNED INFERENCE

Anonymous authors
Paper under double-blind review

ABSTRACT

We propose a novel hierarchical generative model with a simple Markovian structure and a corresponding inference model. Both the generative and inference model are trained using the adversarial learning paradigm. We demonstrate that the hierarchical structure supports the learning of progressively more abstract representations as well as providing semantically meaningful reconstructions with different levels of fidelity. Furthermore, we show that minimizing the Jensen-Shanon divergence between the generative and inference network is enough to minimize the reconstruction error. The resulting semantically meaningful hierarchical latent structure discovery is exemplified on the CelebA dataset. There, we show that the features learned by our model in an unsupervised way outperform the best handcrafted features. Furthermore, the extracted features remain competitive when compared to several recent deep supervised approaches on an attribute prediction task on CelebA. Finally, we leverage the model’s inference network to achieve state-of-the-art performance on a semi-supervised variant of the MNIST digit classification task.

1 INTRODUCTION

Deep generative models represent powerful approaches to modeling highly complex high-dimensional data. There has been a lot of recent research geared towards the advancement of deep generative modeling strategies, including Variational Autoencoders (VAE) (Kingma & Welling [2013]), autoregressive models (Oord et al. [2016a]), and hybrid models (Guirajani et al. [2016], Nguyen et al. [2016]). However, Generative Adversarial Networks (GANs) (Goodfellow et al. [2014]) have emerged as the learning paradigm of choice across a varied range of tasks, especially in computer vision (Zhu et al. [2017]), simulation and robotics (Finn et al. [2016], Shrivastava et al. [2016]). GANs cast the learning of a generative network in the form of a game between the generative and discriminator networks. While the discriminator is trained to distinguish between the true and generated examples, the generative model is trained to fool the discriminator. Using a discriminator network in GANs avoids the need for an explicit reconstruction-based loss function. This allows this model class to generate visually sharper images than VAEs while simultaneously enjoying faster sampling than autoregressive models.

Recent work, known as either ALI (Dumoulin et al. [2016]) or BiGAN (Donahue et al. [2016]), has shown that the adversarial learning paradigm can be extended to incorporate the learning of an inference network. While the inference network, or encoder, maps training examples $x$ to a latent space variable $z$, the decoder plays the role of the standard GAN generator mapping from space of the latent variables (that is typically sampled from some factorial distribution) into the data space. In ALI, the discriminator is trained to distinguish between the encoder and the decoder, while the encoder and decoder are trained to conspire together to fool the discriminator. Unlike some approaches that hybridize VAE-style inference with GAN-style generative learning (e.g., Larsen et al. [2015], Chen et al. [2016]), the encoder and decoder in ALI use a purely adversarial approach. One big advantage of adopting an adversarial-only formalism is demonstrated by the high-quality of the generated samples. Additionally, we are given a mechanism to infer the latent code associated with a true data example.

One interesting feature highlighted in the original ALI work (Dumoulin et al. [2016]) is that even though the encoder and decoder models are never explicitly trained to perform reconstruction, this can nevertheless be easily done by projecting data samples via the encoder into the latent space, copying
these values across to the latent variable layer of the decoder and projecting them back to the data space. Doing this yields reconstructions that often preserve some semantic features of the original input data, but are perceptually relatively different from the original samples. These observations naturally lead to the question of the source of the discrepancy between the data samples and their ALI reconstructions. Is the discrepancy due to a failure of the adversarial training paradigm, or is it due to the more standard challenge of compressing the information from the data into a rather restrictive latent feature vector? Ulyanov et al. (2017) show that an improvement in reconstructions is achievable when additional terms which explicitly minimize reconstruction error in the data space are added to the training objective. Chunyuan Li (2017) palliates to the non-identifiability issues pertaining to bidirectional adversarial training by augmenting the generator’s loss with an adversarial cycle consistency loss.

In this paper we explore issues surrounding the representation of complex, richly-structured data, such as natural images, in the context of a novel, hierarchical generative model, Hierarchical Adversarially Learned Inference (HALI), which represents a hierarchical extension of ALI. We show that within a purely adversarial training paradigm, and by exploiting the model’s hierarchical structure, we can modulate the perceptual fidelity of the reconstructions. We provide theoretical arguments for why HALI’s adversarial game should be sufficient to minimize the reconstruction cost and show empirical evidence supporting this perspective. Finally, we evaluate the usefulness of the learned representations on a semi-supervised task on MNIST and an attribution prediction task on the CelebA dataset.

2 RELATED WORK

Our work fits into the general trend of hybrid approaches to generative modeling that combine aspects of VAEs and GANs. For example, Adversarial Autoencoders (Makhzani et al., 2015) replace the Kullback-Leibler divergence that appears in the training objective for VAEs with an adversarial discriminator that learns to distinguish between samples from the approximate posterior and the prior. A second line of research has been directed towards replacing the reconstruction penalty from the VAE objective with GANs or other kinds of auxiliary losses. Examples of this include Larsen et al. (2015) that combines the GAN generator and the VAE decoder into one network and Lamb et al. (2016) that uses the loss of a pre-trained classifier as an additional reconstruction loss in the VAE objective. Another research direction has been focused on augmenting GANs with inference machinery. One particular approach is given by Dumoulin et al. (2016) and Donahue et al. (2016), where, like in our approach, there is a separate inference network that is jointly trained with the usual GAN discriminator and generator. Karaletos presents a theoretical framework to jointly train inference networks and generators defined on directed acyclic graphs by leverage multiple discriminators defined nodes and their parents. Another related work is that of Huang et al. (2016b) which takes advantage of the representational information coming from a pre-trained discriminator. Their model decomposes the data generating task into multiple subtasks, where each level outputs an intermediate representation conditioned on the representations from higher level. A stack of discriminators is employed to provide signals for these intermediate representations. The idea of stacking discriminator can be traced back to Denton et al. (2015) which used used a succession of convolutional networks within a Laplacian pyramid framework to progressively increase the resolution of the generated images.

3 HIERARCHICAL ADVERSARIALY LEARNED INFERENCE

The goal of generative modeling is to capture the data-generating process with a probabilistic model. Most real-world data is highly complex and thus, the exact modeling of the underlying probability density function is usually computationally intractable. Motivated by this fact, GANs (Goodfellow et al., 2014) model the data-generating distribution as a transformation of some fixed distribution over latent variables. In particular, the adversarial loss, through a discriminator network, forces the generator network to produce samples that are close to those of the data-generating distribution. While GANs are flexible and provide good approximations to the true data-generating mechanism, their original formulation does not permit inference on the latent variables. In order to mitigate this, Adversarially Learned Inference (ALI) (Dumoulin et al., 2016) extends the GAN framework to include an inference network that encodes the data into the latent space. The discriminator is then
We endeavor to improve on ALI in two aspects. First, as reconstructions from ALI only loosely match the input on a perceptual level, we want to achieve better perceptual matching in the reconstructions. Second, we wish to be able to compress the observables, \( x \), using a sequence of composed features maps, leading to a distilled hierarchy of stochastic latent representations, denoted by \( z_L \) to \( z_1 \). Note that, as a consequence of the data processing inequality (Cover & Thomas, 2012), latent representations higher up in the hierarchy cannot contain more information than those situated lower in the hierarchy. In information-theoretic terms, the conditional entropy of the observables given a latent variable is non-increasing as we ascend the hierarchy. This loss of information can be seen as responsible for the perceptual discrepancy observed in ALI’s reconstructions. Thus, the question we seek to answer becomes: How can we achieve high perceptual fidelity of the data reconstructions while also having a compressed latent space that is strongly coupled with the observables? In this paper, we propose to answer this using a novel model, Hierarchical Adversarially Learned Inference (HALI), that uses simple hierarchical Markovian inference network that is matched through adversarial training to a similarly constructed generator network.

Furthermore, we discuss the hierarchy of reconstructions induced by the HALI’s hierarchical inference network and show that the resulting reconstruction errors are implicitly minimized during adversarial training. Also, we leverage HALI’s hierarchical inference network to offer a novel approach to semi-supervised learning in generative adversarial models.

### 3.1 A Model for Hierarchical Features

Denote by \( \mathcal{P}(S) \) the set of all probability measures on some set \( S \). Let \( T_{z_L|x} \) be a Markov kernel associating to each element \( x \in X \) a probability measure \( \mathbb{P}_{z_L|x=x} \in \mathcal{P}(Z) \). Given two Markov kernels \( T_{W|V} \) and \( T_{V|U} \), a further Markov kernel can be defined by composing these two and then marginalizing over \( V \), i.e. \( T_{W|V} \circ T_{V|U} : U \to \mathcal{P}(W) \). Consider a set of random variables \( x, z_1, \ldots, z_L \). Using the composition operation, we can construct a hierarchy of Markov kernels or feature transitions as

\[
T_{z_L|x} = T_{z_L|z_{L-1}} \circ \cdots \circ T_{z_1|x}.
\]

A desirable property for these feature transitions is to have some form of inverses. Motivated by this, we define the adjoint feature transition as \( T_{z_L|x}^\dagger = T_{z_{L-1}|z_L} \circ \cdots \circ T_{z_1|x} \). From this, we see that

\[
T_{z_L|x}^\dagger = T_{z|x} \circ T_{z_1|z_2} \circ \cdots \circ T_{z_{L-1}|z_L}.
\]

This can be interpreted as the generative mechanism of the latent variables given the data being the "inverse" of the data-generating mechanism given the latent variables. Let \( q(x) \) denote the distribution of the data and \( p(z_L) \) be the prior on the latent variables. Typically the prior will be a simple distribution, e.g. a standard Gaussian \( p(z_L) = \mathcal{N}(0 \mid I) \).

The composition of Markov kernels in Eq. 1, mapping data samples \( x \) to samples of the latent variables \( z_L \) using \( z_1, \ldots, z_{L-1} \) constitutes the encoder. Similarly, the composition of kernels in Eq. 2, mapping prior samples of \( z_L \) to data samples \( x \) through \( z_{L-1}, \ldots, z_1 \) constitutes the decoder. Thus, the joint distribution of the encoder can be written as

\[
q(x, \ldots, z_L) = \prod_{l=2}^{L} q(z_l \mid z_{l-1}) q(z_1 \mid x) q(x),
\]

while the joint distribution of the decoder is given by

\[
p(x, \ldots, z_L) = p(x \mid z_1) \prod_{l=2}^{L} p(z_{l-1} \mid z_l) p(z_L).
\]
Algorithm 1 HALI training procedure.

\[ \theta_g, \theta_d \leftarrow \text{initialize network parameters} \]

repeat

for \( m \in \{1, \ldots, M\} \) do

\[ z_0^{(m)} \sim q(x) \quad \text{Sample from the dataset} \]

\[ z_L^{(m)} \sim p(z) \quad \text{Sample from the prior} \]

for \( l \in \{1, \ldots, L\} \) do

\[ z_l^{(m)} \sim q(z_l | z_{l-1}) \quad \text{Sample from each level in the encoder’s hierarchy} \]

end for

for \( l \in \{L + 1\} \) do

\[ z_l^{(m)} \sim p(z_{l-1} | z_l) \quad \text{Sample from each level in the decoder’s hierarchy} \]

end for

\[ \rho_q^{(m)} \leftarrow D(z_0^{(m)}, \ldots, z_l^{(m)}) \quad \text{Get discriminator predictions on encoder’s distribution} \]

\[ \rho_p^{(m)} \leftarrow D(z_0^{(m)}, \ldots, z_l^{(m)}) \quad \text{Get discriminator predictions on decoder’s distribution} \]

end for

\[ L_d \leftarrow -\frac{1}{M} \sum_{i=1}^{M} \log(\rho_q^{(i)}) - \frac{1}{M} \sum_{i=1}^{M} \log(1 - \rho_p^{(i)}) \quad \text{Compute discriminator loss} \]

\[ L_g \leftarrow -\frac{1}{M} \sum_{i=1}^{M} \log(1 - \rho_q^{(i)}) - \frac{1}{M} \sum_{i=1}^{M} \log(\rho_p^{(i)}) \quad \text{Compute generator loss} \]

\[ \theta_d \leftarrow \theta_d - \nabla_{\theta_d} L_d \quad \text{Gradient update on discriminator network} \]

\[ \theta_g \leftarrow \theta_g - \nabla_{\theta_g} L_g \quad \text{Gradient update on generator networks} \]

until convergence

The encoder and decoder distributions can be visualized graphically as

| \( x \) | \( T_{x|z_1} \) | \( z_1 \) | \( T_{x_1|z_2} \) | \( z_2 \) | \( \ldots \) | \( T_{x_L|z_{L-1}} \) | \( z_L \) |

Having constructed the joint distributions of the encoder and decoder, we can now match these distributions through adversarial training. It can be shown that, under an ideal (non-parametric) discriminator, this is equivalent to minimizing the Jensen-Shanon divergence between the joint [Eq. 3 and Eq. 4] and [Dumoulin et al. 2016]. Algorithm 1 details the training procedure.

3.2 A HIERARCHY OF RECONSTRUCTIONS

The Markovian character of both the encoder and decoder implies a hierarchy of reconstructions in the decoder. In particular, for a given observation \( x \sim p(x) \), the model yields \( L \) different reconstructions \( x_l \sim T_{x|z_l} \circ T_{z_l|z_{l-1}} \) for \( l \in \{1, \ldots, L\} \) with \( \hat{x}_l \) the reconstruction of the \( x \) at the \( l \)-th level of the hierarchy. Here, we can think of \( T_{x|z_l} \) as projecting \( x \) to the \( l \)-th intermediate representation and \( T_{z_l|z_{l-1}} \) as projecting it back to the input space. Then, the reconstruction error for a given input \( x \) at the \( l \)-th hierarchical level is given by

\[ L^l(x) = \mathbb{E}_{z_l \sim T_{z_l|x}} [-\log(p(x | z_l))]. \quad (5) \]

Contrary to models that try to merge autoencoders and adversarial models, e.g. [Rosca et al. 2017; Larsen et al. 2015], HALI does not require any additional terms in its loss function in order to minimize the above reconstruction error. Indeed, the reconstruction errors at the different levels of HALI are minimized down to the amount of information about \( x \) that a given level of the hierarchy is able to encode as training proceeds. Furthermore, under an optimal discriminator, training in HALI minimizes the Jensen-Shanon divergence between \( q(x, z_1, \ldots, z_L) \) and \( p(x, z_1, \ldots, z_L) \) as formalized in Proposition below. Furthermore, the interaction between the reconstruction error and training dynamics is captured in Proposition below.

Proposition 1. Assuming \( q(x, z_l) \) is bounded away for zero for all \( l \in \{1, \ldots, L\} \), we have that

\[ \mathbb{E}_{x \sim p(x)} [L^l(x)] - H(x | z_l) \leq K D_{JS}(p(x, z_1, \ldots, z_L) \| q(x, z_1, \ldots, z_L)), \quad (6) \]

where \( H(x | z_l) \) is computed under the encoder’s distribution and \( K \) is as defined in Lemma 2 in the appendix.
On the other hand, proposition 2 below relates the intermediate representations in the hierarchy to
the corresponding induced reconstruction error.

**Proposition 2.** For any given latent variable $z_l$,

$$
E_{x \sim q_x} [E_{z \sim T_{z|z_l} | x} [- \log p(x | z_l)] ] \geq H(x | z_l)
$$

(7)
i.e. the reconstruction error is an upper bound on $H(x | z_l)$.

In summary, Propositions 1 and 2 establish the dynamics between the hierarchical representation
learned by the inference network, the reconstruction errors and the adversarial matching of the joint
distributions Eq. 3 and Eq. 4. The proofs on the two propositions above are deferred to the appendix.

## 4 Empirical Analysis: Setup

We designed our experiments with the objective of addressing the following questions: Is HALI
successful in improving the fidelity perceptual reconstructions? Does HALI induces a semanti-
cally meaningful representation of the observed data? Are the learned representations useful for
downstream classification tasks? All of these questions are considered in turn in the following
sections.

We evaluated HALI on four datasets, CIFAR10 (Krizhevsky & Hinton, 2009), SVHN (Netzer et al.,
2011), ImageNet 128x128 (Russakovsky et al., 2015) and CelebA (Liu et al., 2015). We used
two conditional hierarchies in all experiments with the Markov kernels parametrized by conditional
isotropic Gaussians. For SVHN, CIFAR10 and CelebA the resolutions of two level latent variables are
$z_1 \in \mathbb{R}^{64 \times 16 \times 16}$ and $z_2 \in \mathbb{R}^{256}$. For ImageNet, the resolutions is $z_1 \in \mathbb{R}^{64 \times 32 \times 32}$ and $z_2 \in \mathbb{R}^{256}$.

For both the encoder and decoder, we use residual blocks (He et al., 2015) with skip connections
between the blocks in conjunction with batch normalization (Ioffe & Szegedy, 2015). We use
convolution with stride 2 for downsampling in the encoder and bilinear upsampling in the decoder. In
the discriminator, we use consecutive stride 1 and stride 2 convolutions and weight normalization
(Salimans & Kingma, 2016). To regularize the discriminator, we apply dropout every 3 layers with
a probability of retention of 0.2. We also add Gaussian noise with standard deviation of 0.2 at the
inputs of the discriminator and the encoder.

## 5 Empirical Analysis I: Reconstructions

One of the desired objectives of a generative model is to reconstruct the input images from the
latent representation. We show that HALI offers improved perceptual reconstructions relative to the
(non-hierarchical) ALI model.

### 5.1 Qualitative analysis

First, we present reconstructions obtained on ImageNet. Reconstructions from SVHN and CIFAR10
can be seen in Fig. 7 in the appendix. Fig. 1 highlights HALI’s ability to reconstruct the input
samples with high fidelity. We observe that reconstructions from the first level of the hierarchy exhibit
local differences in the natural images, while reconstructions from the second level of the hierarchy
displays global change. Higher conditional reconstructions are more often than not reconstructed as a
different member of the same class. Moreover, we show in Fig. 2 that this increase in reconstruction
fidelity does not impact the quality of the generative samples from HALI’s decoder.

### 5.2 Quantitative analysis

We further investigate the quality of the reconstructions with a quantitative assessment of the preser-
vation of perceptual features in the input sample. For this evaluation task, we use the CelebA
dataset where each image comes with a 40 dimensional binary attributes vector. A VGG-16 classifier
(Simonyan & Zisserman, 2014) was trained on the CelebA training set to classify the individual
attributes. This trained model is then used to classify the attributes of the reconstructions from the
Figure 1: ImageNet128 reconstructions from $z_1$ and $z_2$. Odd columns correspond to examples from the validation set while even columns are the model’s reconstructions.

Figure 2: Samples from 128 × 128 CelebA and ImageNet128 datasets.
Under review as a conference paper at ICLR 2018

<table>
<thead>
<tr>
<th></th>
<th>Mean</th>
<th>Std</th>
<th># Best</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data</td>
<td>77.13</td>
<td>12.48</td>
<td>40</td>
</tr>
<tr>
<td>VAE</td>
<td>81.28</td>
<td>10.50</td>
<td>5</td>
</tr>
<tr>
<td>ALI</td>
<td>84.60</td>
<td>5.73</td>
<td>3</td>
</tr>
<tr>
<td>HALI $z_1$</td>
<td>91.35</td>
<td>5.62</td>
<td>14</td>
</tr>
<tr>
<td>HALI $z_2$</td>
<td>86.28</td>
<td>5.64</td>
<td>3</td>
</tr>
</tbody>
</table>

Table 1: Summary of CelebA attributes classification from reconstructions for VAE, ALI and the two levels of HALI. The data row is the summary for the VGG classifier and the other scores have been normalized by it. Mean and standard deviation are expressed as percentages. # best represents the count of when a model has the best score on a single attribute. Note that it does not sum to 40 as there were ties.

We report a summary of the statistics of the classifier’s accuracies in Table 1. We do this for three different models, VAE, ALI and HALI. An inspection of the table reveals that the proportion of attributes where HALI’s reconstructions outperforms the other models is clearly dominant. Therefore, the encoder-decoder relationship of HALI better preserves the identifiable attributes compared to other models leveraging such relationships. Please refer to Table 5 in the appendix for the full table of attributes score.

5.3 Perceptual Reconstructions

In the same spirit as [Larsen et al., 2015], we construct a metric by computing the Euclidean distance between the input images and their various reconstructions in the discriminator’s feature space. More precisely, let $\cdot \mapsto \bar{D}(\cdot)$ be the embedding of the input to the pen-ultimate layer of the discriminator. We compute the discriminator embedded distance

$$d_c(u, v) = \|\bar{D}(u, \hat{u}_1, \hat{u}_2) - \bar{D}(v, \hat{v}_1, \hat{v}_2)\|_2,$$

(8)

where $\cdot \mapsto \|\cdot\|_2$ is the Euclidean norm. We then compute the average distances $d_c(x, \hat{x}_1)$ and $d_c(x, \hat{x}_2)$ over the ImageNet validation set. Fig. 3 shows that under $d_c$, the average reconstruction errors for both $\hat{x}_1$ and $\hat{x}_2$ decrease steadily as training advances. Furthermore, the reconstruction error under $d_c$ of the reconstructions from the first level of the hierarchy are uniformly bounded by above those of the second. We note that while the VAEGAN model of [Larsen et al., 2015] explicitly minimizes the perceptual reconstruction error by adding this term to their loss function, HALI implicitly minimizes it during adversarial training, as shown in subsection 3.2.

![Figure 3: Comparison of average reconstruction error over the validation set for each level of reconstructions using the Euclidean (a) and discriminator embedded (b) distances. Using both distances, reconstructions errors for $x \sim T_{z|x}$ are uniformly below those for $x \sim T_{\hat{z}|x}$. The reconstruction error using the Euclidean distance eventually stalls showing that the Euclidean metric poorly approximates the manifold of natural images.](image)

6 Empirical Analysis II: Learned Representations

We now move on to assessing the quality of our learned representation through inpainting, visualizing the hierarchy and innovation vectors.
6.1 INPAINTING

Inpainting is the task of reconstructing the missing or lost parts of an image. It is a challenging task since sufficient prior information is needed to meaningfully replace the missing parts of an image. While it is common to incorporate inpainting-specific training [10], [11], [12], in our case we simply use the standard HALI adversarial loss during training and reconstruct incomplete images during inference time.

We first predict the missing portions from the higher level reconstructions followed by iteratively using the lower level reconstructions that are pixel-wise closer to the original image. Fig. 4 shows the inpaintings on center-cropped SVHN, CelebA and MS-COCO [13] datasets without any blending post-processing or explicit supervision. The effectiveness of our model at this task is due the hierarchy – we can extract semantically consistent reconstructions from the higher levels of the hierarchy, then leverage pixel-wise reconstructions from the lower levels.

6.2 HIERARCHICAL LATENT REPRESENTATIONS

To qualitatively show that higher levels of the hierarchy encode increasingly abstract representation of the data, we individually vary the latent variables and observe the effect.

The process is as follow: we sample a latent code from the prior distribution $z_2$. We then multiply individual components of the vector by scalars ranging from $-3$ to $3$. For $z_1$, we fix $z_2$ and multiply each feature map independently by scalars ranging from $-3$ to $3$. In all cases these modified latent vectors are then decoded back to input data space. Fig. 6 (a) and (b) exhibit some of those decodings for $z_2$, while (c) and (d) do the same for the lower conditional $z_1$. The last column contain the decodings obtained from the originally sampled latent codes. We see that the representations learned in the $z_2$ conditional are responsible for high level variations like gender, while $z_1$ codes imply local/pixel-wise changes such as saturation or lip color.
6.3 Latent Semantic Innovation

With HALI, we can exploit the jointly learned hierarchical inference mechanism to modify actual data samples by manipulating their latent codes. We refer to these sorts of manipulations as latent semantic innovations.

Consider a given instance from a dataset $x \sim q(x)$. Encoding $x$ yields $\hat{z}_1$ and $\hat{z}_2$. We modify $\hat{z}_2$ by multiplying a specific entry by a scalar $\alpha$. We denote the resulting vector by $\hat{z}_2^\alpha$. We decode the latter and get $\tilde{z}_1^{\alpha} \sim T_{\tilde{z}_1|\hat{z}_2}$. We decode the unmodified encoding vector and get $\hat{z}_1 \sim T_{\hat{z}_1|\hat{z}_2}$. We then form the innovation tensor $\eta^{\alpha} = \hat{z}_1 - \hat{z}_1^{\alpha}$. Finally, we subtract the innovation vector from the initial encoding, thus getting $\hat{z}_1^{\alpha} = \hat{z}_1 - \eta^{\alpha}$, and sample $\tilde{x}^{\alpha} \sim T_{\tilde{x}|\hat{z}_1^{\alpha}}$. This method provides explicit control and allows us to carry out these variations on real samples in a completely unsupervised way. The results are shown in Fig. 5. These were done on the CelebA validation set and were not used for training.

7 Empirical Evaluations III: Learning Predictive Representations

We evaluate the usefulness of our learned representation for downstream tasks by quantifying the performance of HALI on attribute classification in CelebA and on a semi-supervised variant of the MNIST digit classification task.

7.1 Unsupervised Classification

Following the protocol established by Berg & Belhumeur (2013); Liu et al. (2015), we train 40 linear SVMs on HALI encoder representations (i.e. we utilize the inference network) on the CelebA validation set and subsequently measure performance on the test set. As in Berg & Belhumeur (2013); Huang et al. (2016a); Kalayeh et al. (2017), we report the balanced accuracy in order to evaluate the attribute prediction performance. We emphasize that, for this experiment, the HALI encoder and decoder were trained in on entirely unsupervised data. Attribute labels were only used to train the linear SVM classifiers.

A summary of the results are reported in Table 2. HALI’s unsupervised features surpass those of VAE and ALI, but more remarkably, they outperform the best handcrafted features by a wide margin (Zhang et al., 2014). Furthermore, our approach outperforms a number of supervised (Huang et al., 2016a) and deeply supervised (Liu et al., 2015) features. Table 6 in the appendix arrays the results per attribute.
Table 2: Summary of statistics for CelebA attributes mean per-class balanced accuracy given in percentage points. # best represents the count of when a model has the best score on a single attribute. Note that it does not sum to 40 as there were ties.

<table>
<thead>
<tr>
<th>Model</th>
<th>Mean</th>
<th>Std</th>
<th># Best</th>
</tr>
</thead>
<tbody>
<tr>
<td>Triplet-kNN (Schroff et al., 2015)</td>
<td>71.55</td>
<td>12.61</td>
<td>0</td>
</tr>
<tr>
<td>PANDA (Zhang et al., 2014)</td>
<td>76.95</td>
<td>13.33</td>
<td>0</td>
</tr>
<tr>
<td>Anet (Liu et al., 2015)</td>
<td>79.56</td>
<td>12.17</td>
<td>0</td>
</tr>
<tr>
<td>LMLE-kNN (Huang et al., 2016a)</td>
<td>83.83</td>
<td>12.33</td>
<td>22</td>
</tr>
<tr>
<td>VAE</td>
<td>73.30</td>
<td>9.65</td>
<td>0</td>
</tr>
<tr>
<td>ALI</td>
<td>73.88</td>
<td>10.16</td>
<td>0</td>
</tr>
<tr>
<td>HALI</td>
<td>83.75</td>
<td>8.96</td>
<td>15</td>
</tr>
</tbody>
</table>

Table 3: Comparison on semi-supervised learning with state-of-the-art methods on MNIST with 100 labels instance per class. Only methods without data augmentation are included.

<table>
<thead>
<tr>
<th>Method</th>
<th>MNIST (# errors)</th>
</tr>
</thead>
<tbody>
<tr>
<td>VAE (M1+M2) (Kingma et al., 2014)</td>
<td>233 ± 14</td>
</tr>
<tr>
<td>VAT (Miyato et al., 2017)</td>
<td>136</td>
</tr>
<tr>
<td>CatGAN (Springenberg, 2015)</td>
<td>191 ± 10</td>
</tr>
<tr>
<td>Adversarial Autoencoder (Makhzani et al., 2015)</td>
<td>190 ± 10</td>
</tr>
<tr>
<td>PixelGAN (Makhzani &amp; Frey, 2017)</td>
<td>108 ± 15</td>
</tr>
<tr>
<td>ADGM (Maaløe et al., 2016)</td>
<td>96 ± 2</td>
</tr>
<tr>
<td>Feature-Matching GAN (Salimans et al., 2016)</td>
<td>93 ± 6.5</td>
</tr>
<tr>
<td>Triple GAN (Li et al., 2017)</td>
<td>91 ± 58</td>
</tr>
<tr>
<td>GSSLTRABG (Dai et al., 2017)</td>
<td>79.5 ± 9.8</td>
</tr>
<tr>
<td>HALI (ours)</td>
<td>73</td>
</tr>
</tbody>
</table>

7.2 SEMI-SUPERVISED LEARNING WITHIN HALI

The HALI hierarchy can also be used in a more integrated semi-supervised setting, where the encoder also receives a training signal from the supervised objective. The currently most successful approach to semi-supervised in adversarially trained generative models are built on the approach introduced by Salimans et al. (2016). This formalism relies on exploiting the discriminator’s feature to differentiate between the individual classes present in the labeled data as well as the generated samples. Taking inspiration from Makhzani et al. (2015), Makhzani & Frey (2017), we adopt a different approach that leverages the Markovian hierarchical inference network made available by HALI,

\[ x \rightarrow z \rightarrow y, \] (9)

Where \( z = enc(x + \epsilon) \), with \( \epsilon \sim \mathcal{N}(0, I) \), and \( y \) is a categorical random variable. In practice, we characterize the conditional distribution of \( y \) given \( z \) by a softmax. The cost of the generator is then augmented by a supervised cost. Let us write \( D_{sup} \) as the set of pairs all labeled instance along with their label, the supervised cost reads

\[ L_{sup} = \frac{1}{|D_{sup}|} \sum_{(y, x) \in D_{sup}} \sum_{k} y_k \log(y_k). \] (10)

We showcased this approach on a semi-supervised variant of MNIST (LeCun et al., 1998) digit classification task with 100 labeled examples evenly distributed across classes. Table 3 shows that HALI achieves a new state-of-the-art result for this setting. Note that unlike Dai et al. (2017), HALI uses no additional regularization.

8 CONCLUSION AND FUTURE WORK

In this paper, we introduced HALI, a novel adversarially trained generative model. HALI learns a hierarchy of latent variables with a simple Markovian structure in both the generator and inference
network. We have shown both theoretically and empirically the advantages gained by extending the ALI framework to a hierarchy.

While there are many potential applications of HALI, one important future direction of research is to explore ways to render the training process more stable and straightforward. GANs are well-known to be challenging to train and the introduction of a hierarchy of latent variables only adds to this.

REFERENCES


Theofanis Karaletsos. Adversarial message passing for graphical models. *NIPS workshop on Advances in Approximate Bayesian Inference*.


### A Architecture Details

<table>
<thead>
<tr>
<th>Operation</th>
<th>Kernel</th>
<th>Strides</th>
<th>Feature maps</th>
<th>BN/WN?</th>
<th>Dropout</th>
<th>Nonlinearity</th>
</tr>
</thead>
<tbody>
<tr>
<td>$G_{z_1}(x) - 3 \times 128 \times 128$ input</td>
<td>Convolution</td>
<td>$3 \times 3$</td>
<td>$1 \times 1$</td>
<td>32</td>
<td>$\times$</td>
<td>0.0</td>
</tr>
<tr>
<td></td>
<td>Convolution</td>
<td>$3 \times 3$</td>
<td>$2 \times 2$</td>
<td>64</td>
<td>BN</td>
<td>0.0</td>
</tr>
<tr>
<td></td>
<td>Resnet Block</td>
<td>$3 \times 3$</td>
<td>$1 \times 1$</td>
<td>64</td>
<td>BN</td>
<td>0.0</td>
</tr>
<tr>
<td></td>
<td>Convolution</td>
<td>$3 \times 3$</td>
<td>$2 \times 2$</td>
<td>128</td>
<td>BN</td>
<td>0.0</td>
</tr>
<tr>
<td></td>
<td>Resnet Block</td>
<td>$3 \times 3$</td>
<td>$1 \times 1$</td>
<td>128</td>
<td>BN</td>
<td>0.0</td>
</tr>
<tr>
<td>Gaussian Layer</td>
<td>Convolution</td>
<td>$3 \times 3$</td>
<td>$1 \times 1$</td>
<td>128</td>
<td>$\times$</td>
<td>0.0</td>
</tr>
<tr>
<td>$G_{z_2}(z_1) - 64 \times 32 \times 32$ input</td>
<td>Convolution</td>
<td>$3 \times 3$</td>
<td>$2 \times 2$</td>
<td>256</td>
<td>BN</td>
<td>0.0</td>
</tr>
<tr>
<td></td>
<td>Convolution</td>
<td>$3 \times 3$</td>
<td>$2 \times 2$</td>
<td>512</td>
<td>BN</td>
<td>0.0</td>
</tr>
<tr>
<td></td>
<td>Resnet Block</td>
<td>$3 \times 3$</td>
<td>$1 \times 1$</td>
<td>512</td>
<td>BN</td>
<td>0.0</td>
</tr>
<tr>
<td></td>
<td>Convolution</td>
<td>$4 \times 4$</td>
<td>$valid$</td>
<td>512</td>
<td>BN</td>
<td>0.0</td>
</tr>
<tr>
<td></td>
<td>Convolution</td>
<td>$1 \times 1$</td>
<td>$1 \times 1$</td>
<td>512</td>
<td>$\times$</td>
<td>0.0</td>
</tr>
<tr>
<td>$G_{z_1}(z_2) - 128 \times 1 \times 1$ input</td>
<td>Convolution</td>
<td>$1 \times 1$</td>
<td>$1 \times 1$</td>
<td>$4 \times 4 \times 256$</td>
<td>BN</td>
<td>0.0</td>
</tr>
<tr>
<td>Bilinear Upsampling</td>
<td>Resnet Block</td>
<td>$3 \times 3$</td>
<td>$1 \times 1$</td>
<td>256</td>
<td>BN</td>
<td>0.0</td>
</tr>
<tr>
<td>Bilinear Upsampling</td>
<td>Convolution</td>
<td>$3 \times 3$</td>
<td>$1 \times 1$</td>
<td>128</td>
<td>BN</td>
<td>0.0</td>
</tr>
<tr>
<td>Bilinear Upsampling</td>
<td>Convolution</td>
<td>$3 \times 3$</td>
<td>$1 \times 1$</td>
<td>128</td>
<td>$\times$</td>
<td>0.0</td>
</tr>
<tr>
<td>Gaussian Layer</td>
<td>Convolution</td>
<td>$3 \times 3$</td>
<td>$1 \times 1$</td>
<td>64</td>
<td>BN</td>
<td>0.0</td>
</tr>
<tr>
<td>$G_x(z_1) - 64 \times 32 \times 32$ input</td>
<td>Convolution</td>
<td>$1 \times 1$</td>
<td>$1 \times 1$</td>
<td>64</td>
<td>BN</td>
<td>0.0</td>
</tr>
<tr>
<td>Bilinear Upsampling</td>
<td>Resnet Block</td>
<td>$3 \times 3$</td>
<td>$1 \times 1$</td>
<td>64</td>
<td>BN</td>
<td>0.0</td>
</tr>
<tr>
<td>Bilinear Upsampling</td>
<td>Convolution</td>
<td>$3 \times 3$</td>
<td>$1 \times 1$</td>
<td>64</td>
<td>BN</td>
<td>0.0</td>
</tr>
<tr>
<td>Bilinear Upsampling</td>
<td>Convolution</td>
<td>$3 \times 3$</td>
<td>$1 \times 1$</td>
<td>3</td>
<td>$\times$</td>
<td>0.0</td>
</tr>
<tr>
<td>$D(x) - 3 \times 128 \times 128$ input</td>
<td>Convolution</td>
<td>$3 \times 3$</td>
<td>$1 \times 1$</td>
<td>32</td>
<td>WN</td>
<td>0.2</td>
</tr>
<tr>
<td></td>
<td>Convolution</td>
<td>$3 \times 3$</td>
<td>$2 \times 2$</td>
<td>64</td>
<td>WN</td>
<td>0.5</td>
</tr>
<tr>
<td></td>
<td>Convolution</td>
<td>$3 \times 3$</td>
<td>$2 \times 2$</td>
<td>64</td>
<td>WN</td>
<td>0.5</td>
</tr>
<tr>
<td></td>
<td>Convolution</td>
<td>$3 \times 3$</td>
<td>$1 \times 1$</td>
<td>64</td>
<td>WN</td>
<td>0.5</td>
</tr>
<tr>
<td>$D(x, z_1) - 128 \times 32 \times 32$</td>
<td>Concatenate</td>
<td>$D(x)$ and $z_1$ along the channel axis</td>
<td>Convolution</td>
<td>$3 \times 3$</td>
<td>$1 \times 1$</td>
<td>128</td>
</tr>
<tr>
<td></td>
<td>Convolution</td>
<td>$3 \times 3$</td>
<td>$2 \times 2$</td>
<td>256</td>
<td>WN</td>
<td>0.5</td>
</tr>
<tr>
<td></td>
<td>Convolution</td>
<td>$3 \times 3$</td>
<td>$2 \times 2$</td>
<td>256</td>
<td>WN</td>
<td>0.5</td>
</tr>
<tr>
<td></td>
<td>Convolution</td>
<td>$3 \times 3$</td>
<td>$2 \times 2$</td>
<td>512</td>
<td>WN</td>
<td>0.5</td>
</tr>
<tr>
<td></td>
<td>Convolution</td>
<td>$4 \times 4$</td>
<td>$valid$</td>
<td>512</td>
<td>WN</td>
<td>0.2</td>
</tr>
<tr>
<td>$D(x, z_1, z_2) - 512 \times 1 \times 1$ input</td>
<td>Concatenate</td>
<td>$D(x, z_1)$ and $z_2$ along the channel axis</td>
<td>Convolution</td>
<td>$1 \times 1$</td>
<td>$1 \times 1$</td>
<td>1024</td>
</tr>
<tr>
<td></td>
<td>Convolution</td>
<td>$1 \times 1$</td>
<td>$1 \times 1$</td>
<td>1024</td>
<td>$\times$</td>
<td>0.5</td>
</tr>
<tr>
<td></td>
<td>Convolution</td>
<td>$1 \times 1$</td>
<td>$1 \times 1$</td>
<td>1</td>
<td>$\times$</td>
<td>0.5</td>
</tr>
</tbody>
</table>

Table 4: Architecture detail for HALI unsupervised on the Imagenet 128 and CelebA 128 Datasets.
B Proofs

Lemma 1. Let $f$ be a valid f-divergence generator. Let $p$ and $q$ be joint distributions over a random vector $x$. Let $x_A$ be any strict subset of $x$ and $x_{-A}$ its complement, then

$$D_f(p(x) || q(x)) \geq D_f(p(x_A) || q(x_A))$$

Proof. By definition, we have

$$D_f(p(x) || q(x)) = \mathbb{E}_{x \sim q(x)}[f\left(\frac{p(x)}{q(x)}\right)] = \mathbb{E}_{x_A \sim q(x_A)}\mathbb{E}_{x_{-A} \sim q(x_{-A}|x_A)}[f\left(\frac{p(x_A) p(x_{-A}|x_A)}{q(x_A) q(x_{-A}|x_A)}\right)]$$

Using that $f$ is convex, Jensen’s inequality yields

$$D_f(p(x) || q(x)) \geq \mathbb{E}_{x_A \sim q(x_A)}[f\left(\frac{p(x_A)}{q(x_A)}\right)]$$

Simplifying the inner expectation on the right hand side, we conclude that

$$D_f(p(x) || q(x)) \geq \mathbb{E}_{x_A \sim q(x_A)}[f\left(\frac{p(x_A)}{q(x_A)}\right)] \quad \square$$

Lemma 2 (Kullback-Leibler’s upper bound by Jensen-Shannon). Assume that $p$ and $q$ are two probability distribution absolutely continuous with respect to each other. Moreover, assume that $q$ is bounded away from zero. Then, there exist a positive scalar $K$ such that

$$D_{KL}(p || q) \leq K D_{JS}(p || q).$$

Proof. We start by bounding the Kullback-Leibler divergence by the $\chi^2$-distance. We have

$$D_{KL}(p || q) \leq \int \log\left(\frac{p(x)}{q(x)}\right) dx \leq \log(1 + D_{\chi^2}(p || q)) \leq D_{\chi^2}(p || q)$$

The first inequality follows by Jensen’s inequality. The third inequality follows by the Taylor expansion. Recall that both the $\chi^2$-distance and the Jensen-Shannon divergences are f-divergences with generators given by $f_{\chi^2}(t) = (t - 1)^2$ and $f_{JS}(t) = u \log \left( \frac{2t}{1+t} \right) + \log(\frac{2t}{1+t})$, respectively. We form the function $t \mapsto h(t) = \frac{f_{\chi^2}(t)}{f_{JS}(t)}$, $h$ is strictly increasing on $[0, \infty)$. Since we are assuming $q$ to be bounded away from zero, we know that there is a constant $c_1$ such that $q(x) > c_1$ for all $x$. Subsequently for all $x$, we have that $\frac{p(x)}{q(x)} \leq c_2 := \max_x \frac{p(x)}{q(x)}$. Thus, for all $x$ we have $h\left(\frac{p(x)}{q(x)}\right) \leq K := h(c_2)$ and hence $f_{\chi^2}\left(\frac{p(x)}{q(x)}\right) \leq K f_{JS}(\frac{p(x)}{q(x)})$. Integrating with respect to $q$, we conclude

$$D_{KL}(p || q) \leq D_{\chi^2}(p || q) \leq K D_{JS}(p || q) \quad \square$$

Proposition 3. Assuming $q(x, z_l)$ and $p(x, z_l)$ are positive for any $l \in \{1, \ldots, L\}$. We have

$$\mathbb{E}_{x \sim q(x)}[L^l(x)] - H(x | z_l) \leq K D_{JS}(p(x, z_1, \ldots, z_L) || q(x, z_1, \ldots, z_L))$$

Where $H(x | z_l)$ is computed under the encoder’s distribution $q(x, z_l)$

Proof. By elementary manipulations we have.

$$\mathbb{E}_{x \sim q(x)}[L^l(x)] = D_{KL}(p(x, z_l) || q(x, z_l)) + H(x | z_l) - D_{KL}(p(z_l) || q(z_l))$$

Where the conditional entropy $H(x_l | z_l)$ is computed $q(x, z_l)$. By the non-negativity of the KL-divergence we obtain

$$\mathbb{E}_{x \sim q(x)}[L^l(x)] \leq D_{KL}(p(x, z_l) || q(x, z_l)) + H(x | z_l)$$

By adding a non-negative term $D_{KL}(q(x, z_l) || p(x, z_l))$ on the right hand side, we have

$$\mathbb{E}_{x \sim q(x)}[L^l(x)] - H(x | z_l) \leq D_{f}(p(x, z_l) || q(x, z_l))$$
Using lemma 2 we have

\[ \mathbb{E}_{x \sim p_d(x)}[\mathcal{L}(x)] - H(x | z_l) \leq K D_{JS}(p(x, z_l) \parallel q(x, z_l)) \]

The Jensen-Shannon divergence being f-divergence, using Lemma 1 we conclude

\[ \mathbb{E}_{x \sim p_d(x)}[\mathcal{L}(x)] - H(x | z_l) \leq K D_{JS}(p(x, z_1, \ldots, z_L) \parallel q(x, z_1, \ldots, z_L)). \]

Proposition 4. For any given latent variable \( z_l \), the reconstruction likelihood \( \mathbb{E}_{x \sim q(x)}[\mathbb{E}_{z_l \sim T_{z_l}|x}[- \log p(x | z_l)]] \) is an upper bound on \( H(x | z_l) \).

**Proof.** By the non-negativity of the Kullback-Leibler divergence, we have that

\[ D_{KL}(q(x | z_l) \parallel p(x | z_l)) \geq 0 \implies H[q(x | z_l)] \leq \mathbb{E}_{x \sim p(x)}[\mathbb{E}_{z_l \sim T_{z_l}|x}[- \log p(x | z_l)]] \]

Integrating over the marginal and applying Fubini’s theorem yields

\[ H(x | z_l) \leq \mathbb{E}_{x \sim p(x)}[\mathbb{E}_{z_l \sim T_{z_l}|x}[- \log p(x | z_l)]], \]

where the conditional entropy \( H(x | z_l) \) is computed under the encoder distribution.

\[ \square \]
Table 5: CelebA attributes accuracies of reconstructions by different models. The data row displays the raw average of positive accuracies, predicting true 1, and negative accuracies, predicting a true 0 by our VGG classifier. Other rows show the same average accuracies where each individual accuracy is normalized by its corresponding data score. The numbers are all percentages.

<table>
<thead>
<tr>
<th></th>
<th>Black Hair</th>
<th>Blond Hair</th>
<th>Wavy Hair</th>
<th>Young</th>
<th>Makeup</th>
<th>Attractive</th>
<th>with Shadow</th>
<th>withNecktie</th>
<th>Blurry</th>
<th>DoubleChin</th>
<th>BrownHair</th>
<th>Mouth Open</th>
<th>Goatee</th>
<th>Razed</th>
<th>With Hat</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data</td>
<td>86</td>
<td>81</td>
<td>69</td>
<td>91</td>
<td>90</td>
<td>88</td>
<td>81</td>
<td>80</td>
<td>80</td>
<td>60</td>
<td>70</td>
<td>74</td>
<td>92</td>
<td>88</td>
<td>77</td>
</tr>
<tr>
<td>VAE</td>
<td>80</td>
<td>96</td>
<td>60</td>
<td>98</td>
<td>79</td>
<td>74</td>
<td>83</td>
<td>74</td>
<td>93</td>
<td>96</td>
<td>73</td>
<td>77</td>
<td>83</td>
<td>77</td>
<td>89</td>
</tr>
<tr>
<td>ALI (z1)</td>
<td>72</td>
<td>90</td>
<td>83</td>
<td>94</td>
<td>88</td>
<td>81</td>
<td>91</td>
<td>95</td>
<td>95</td>
<td>92</td>
<td>89</td>
<td>92</td>
<td>95</td>
<td>92</td>
<td>89</td>
</tr>
<tr>
<td>HALL (z2)</td>
<td>92</td>
<td>93</td>
<td>98</td>
<td>91</td>
<td>95</td>
<td>95</td>
<td>92</td>
<td>92</td>
<td>92</td>
<td>89</td>
<td>94</td>
<td>98</td>
<td>88</td>
<td>90</td>
<td>95</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Balding</th>
<th>Sideburns</th>
<th>Big Nose</th>
<th>Rosy Cheeks</th>
<th>Oval Face</th>
<th>Bangs</th>
<th>Male</th>
<th>Mustache</th>
<th>HighCheeks</th>
<th>No Beard</th>
<th>Eyeglasses</th>
<th>BigLips</th>
<th>NarrowEyes</th>
<th>Chubby</th>
<th>Smiling</th>
<th>WithEarrings</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data</td>
<td>72</td>
<td>60</td>
<td>69</td>
<td>71</td>
<td>56</td>
<td>89</td>
<td>97</td>
<td>80</td>
<td>85</td>
<td>96</td>
<td>94</td>
<td>69</td>
<td>54</td>
<td>94</td>
<td>54</td>
<td>58</td>
</tr>
<tr>
<td>VAE</td>
<td>89</td>
<td>94</td>
<td>69</td>
<td>67</td>
<td>83</td>
<td>78</td>
<td>93</td>
<td>82</td>
<td>91</td>
<td>98</td>
<td>73</td>
<td>65</td>
<td>92</td>
<td>86</td>
<td>82</td>
<td>78</td>
</tr>
<tr>
<td>ALI (z1)</td>
<td>86</td>
<td>87</td>
<td>89</td>
<td>82</td>
<td>89</td>
<td>76</td>
<td>90</td>
<td>77</td>
<td>86</td>
<td>92</td>
<td>81</td>
<td>87</td>
<td>93</td>
<td>88</td>
<td>83</td>
<td>85</td>
</tr>
<tr>
<td>HALL (z2)</td>
<td>96</td>
<td>91</td>
<td>90</td>
<td>71</td>
<td>91</td>
<td>94</td>
<td>97</td>
<td>84</td>
<td>96</td>
<td>98</td>
<td>90</td>
<td>94</td>
<td>93</td>
<td>94</td>
<td>83</td>
<td>88</td>
</tr>
</tbody>
</table>

Table 6: Mean per-class balanced accuracy in percentage points of each of the 40 face attributes on CelebA.